IMPLICIT FUNCTION MODELING OF NEURON MORPHOLOGY

Sandeep Tewari
Bruce H. McCormick

tewari@aic.lockheed.com
mccormick@cs.tamu.edu

Scientific Visualization Laboratory
Department of Computer Science
Texas A&M University, College Station, TX 77843, U.S.A.

ABSTRACT

Implicit functions are used to develop solid models of individual neurons as viewed at the limit of optical resolution. Stick models generated by the L-system grammars for several neuron types (pyramidal, stellate and motor cells) serve as the basis for these models. In addition, the growth and connectivity of space-filling neuronal structures are modeled. A ray tracing strategy emulates the filopodia growth mechanism of developmental neurobiology. The close-packing of these neuronal structures precludes their assembly from individual fully-grown neurons. Finally, the cylindrical environment for the development of the pyramidal cell module has been modeled as an example of a space-filling structure.

INTRODUCTION

Neurons, scaled by 10^5, show strong resemblance to trees. Implicit function modeling has been successfully used by Bloomenthal to model trees with sufficient detail.[1,2] This similarity of neurons to trees and the work done by Bloomenthal in modeling the maple tree has motivated the use of these "blobby techniques" to develop morphological models of neurons.

COMPUTATIONAL NEUROSCIENCE

L-systems are being used to model various neuron types (pyramidal, stellate, and motor cells) by McCormick & Mulchandani[3]. Starting from this basis, implicit function techniques allow us to model neurons in greater detail.

Neurons are densely packed into neural tissue. This dense-packing precludes assembly of the tissue from individual fully-grown neurons. This growth and connection of space-filling structures is a barrier to visualization and interpretation of neural tissue scanned at the limit of optical resolution. Recent developments in confocal microscopy and mass storage offer the potential for parallel tracking of fibers and neural processes in neural tissue. The interpretation of these massive volumetric data sets will rest on having creditable models of neuron morphology as viewed at the limit of optical resolution.

METHODS

Implicit function modeling of individual neurons

Stick models generated by a L-system grammar for each neuron type have been used as the basis for the implicit models. The stick model of a neuron type is specified as a list of three-dimensional points and list of connections between these points.

"Blobby techniques", similar to those used by Bloomenthal[1], have been used to model the neurons. Smooth blending is particularly significant when modeling junctions and spines. Implicitly-defined surfaces can be easily blended and are used here to advantage[2].

Arbor segments produced by the L-system grammar are generalized cylinders whose axes are space curves. We restricted each space curve to be a parametric spline. The generalized cylinders are defined implicitly. The neuron arbors are then a series of generalized cylinders blended together (Fig. 1). Each generalized cylinder, in turn, is approximated by a set of spheres. The centers of the spheres are referred to as the key points[4]. The axis of each generalized cylinder is then a series of keypoints.

Spines on the neuron surface are "mushroom shaped". Each spine is defined implicitly as a blend of a flattened sphere and a standard cylinder. Functions which depend on a set of independent key points were used. The normal at a point on an implicitly defined surface can be easily computed by taking the partial derivatives of the implicit function at that point[5]. Spines were stuck normal to the dendritic surface by blending the flattened sphere, standard cylinder

and the generalized cylinder based on a minimum distance criteria. To ensure smooth blending between these objects the function proposed by the Wyvills and McPheeters [4] has been used. Multiple distributions (e.g., poisson, the poisson sphere and the jittered distributions) of spines on dendritic segments are offered.

Considerable reduction in the implicit function evaluation time has been achieved by using implicitly defined bounding spheres on all generalized cylinders representing neuron segments and by keeping track of the generalized cylinders within range of a grid point. The above preprocessing step significantly reduces the number of implicit function evaluations because for each grid point only the implicit functions for the generalized cylinders within range are evaluated. The implicit function evaluations for the bounding spheres are much cheaper than those for the generalized cylinders. The marching cubes algorithm[8] has been used to render the neurons.

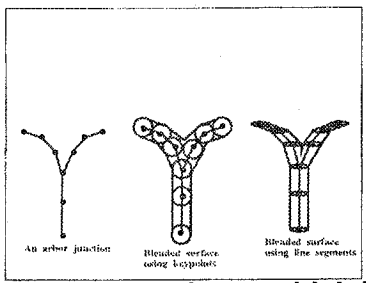

Figure 1: Arbors modeled by blended implicit cylinders.

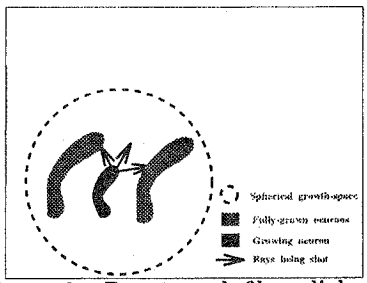

Figure 2: Ray traced filopodial growth.

Modeling space-filling neuronal structures

When neurons grow in parallel they are known to avoid or seek intersection by sending out filopodia. This natural process of collision detection is emulated by using ray tracing (Fig. 2). A small cluster of neurons was approximated by placing a group of implicitly defined cylinders in close juxtaposition; the neurons thus modeled did not have branches. Once this small cluster was modeled, a neuron was grown by stepping at discrete time steps. At each time step ray tracing was used to check for intersections, and then a suitable path for growth was selected. The idea of using ray tracing stems from the fact that the intersection of a ray (parametric form) with an implicit surface can be easily computed.

Computational efficiency is a primary concern when using ray tracing. The number of objects to be intersected with each ray was reduced by adaptively

subdividing the scene into cells. A decomposition rule which associates with each object only the cells through which the surface of the object passes has been used[6].

The growth of neurons is known to be guided by surface adhesion molecules and by netrin concentration gradients. These proteins help the neurons to seek or avoid intersections and guide the neurons through a maze of already existing neurons. The role of these proteins has been emulated by assigning scripts, i.e. properties of attraction and repulsion, to the cells obtained after performing an octree decomposing of the growth space. The growth space was restricted to a standard cylinder and a group of neurons were placed in close juxtaposition in this space. A neuron was grown through this maze and intersections induced or avoided by using ray tracing in conjunction with the cell scripts. By changing the cell scripts different trajectories of growth can be obtained.

Prior to modeling the pyramidal cell module, our methods were tested on a smaller scale. Repulsion properties were assigned to the cells obtained after decomposing the growth space and this simplified "cell script" was used to obtain different trajectories of growth for a neuron growing through this maze without intersections.

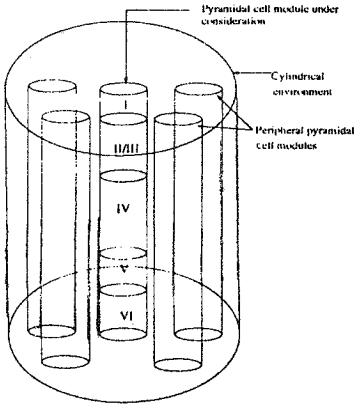

Figure 3: Pyramidal cell module model.

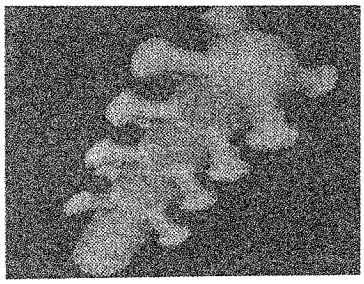

Figure 4: Implicit function model of spines on a neuronal segment.

Modeling the pyramidal cell module

There are 142 cells associated with each pyramidal cell module[7]. Typically, a pyramidal cell module is a cylinder 1600μm long and 31μm in diameter, which means that the pyramidal cell module is densely packed. This dense-packing precludes assembly of the module from individual fully grown neurons; rather, the module must be graphically grown in parallel from seed neurons. The

results of the previous steps were used to model the pyramidal cell module (Fig. 3). Cell scripts were assigned to the participating cells, which were arranged concentric in cylindrical wafers (Fig. 3). Multiple neurons were grown in parallel and the cell scripts modified so as to obtain a close approximation to the structure of a pyramidal cell module (Fig. 3).

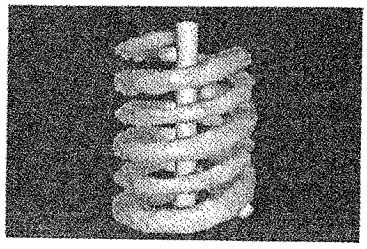

Figure 5: A neuron grown within a double helix.

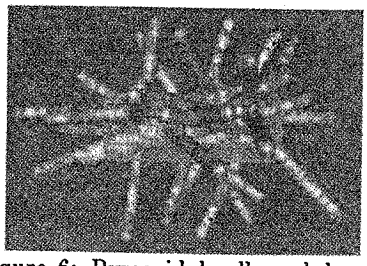

Figure 6: Pyramidal cell module wafer occupied by peripheral dendrites.

RESULTS

Three specific results of this research are summarized below.

1. *Solid modeling of individual neurons*, as viewed at the limit of optical resolution, has been conducted using implicit functions. Stick and disk models generated by the L-system grammars for several neuron types (pyramidal, stellate, and motor cells) are used as the basis for these geometric models. Figure 4 shows an implicit function model of spines on a neuronal segment.

2. *Growth and connection of space-filling neuronal structures* has been performed using a ray tracing strategy to emulate the role of filopodia in directing the growth of nuronal processes. A single neuron grown inside a double helix are shown in figure 5.

3. *The pyramidal cell module* has been partially modeled as an example of a space-filling structure. These pyramidal cell modules are the fundamental neuronal units of the cortex[7]. Figure 6 shows a model of a pyramidal cell module wafer showing the volume occupied by the peripheral dendrites.

Summary and Conclusions

A potential use of these morphological models is two-fold: (1) in simulating the growth of neural tissue at a neuronal level of detail, and (2) in disambiguating volumetric data sets of neural tissue. This research contributes to the visualization and modeling of neuronal circuitry by providing :

- Effective techniques for neuron solid modeling using implicit functions.
- Growth and connection of space-filling structures by using a ray tracing strategy to emulate the filopodia-guided growth of neuron processes.

Pyramidal cell modules can be morphologically modeled by application of these methods. Such morphological models lead naturally to future neuronal and neural network modeling of these fundamental cortical processing units.

References

1. Bloomenthal, J., "Modeling the mighty maple," in *Proc., ACM SIGGRAPH Conference*, vol.19, no. 3, pp 305-311, 1985.

2. Bloomenthal, J., "Techniques for implicit modeling," *ACM SIGGRAPH Course Notes*, no. 25, pp 12-1-12-11, 1993.

3. McCormick B.H. and Mulchandani, K., "L-system modeling of neurons," in *Proc. of the conference on Visualization in Biomedical Computing (VBC'94)*, SPIE 2359, pp 693-705, 1994.

4. Wyvill, G., McPheeters, C. and Wyvill, B., "Data structure for soft objects," *ACM SIGGRAPH course notes*, no. 25, pp 5-1-5-8, 1986.

5. Blinn, J.F., "The algebraic properties of homogeneous second order surfaces," *ACM SIGGRAPH course notes*, no. 25, pp 2-1-2-33, 1993.

6. Glassner, A.S., "Space subdivision for fast ray tracing," *IEEE Computer Graphics and Applications*, no. 4, 10, pp 15-22, 1984.

7. Peters, A., "The organization of the primary visual cortex in the macaque," A. Peters and K.S. Rockland (Eds), *Cerebral Cortex*, vol. 10, Plenum Press, 1994.

8. Lorenson, W.E., and Cline, H.E., "Marching cubes : a high resolution 3D surface reconstruction algorithm," *Computer Graphics*, vol. 21, no. 4, pp 163-169, 1987.