Neuron PRM: A Framework for Constructing Cortical Networks*

Jyh-Ming Lien Marco Morales Nancy M. Amato
Department of Computer Science
Texas A&M University
{neilien,marcom,amato}@cs.tamu.edu

Abstract

We present a framework to construct a 3D model of a cortical network using probabilistic roadmap methods (PRMs) developed in the robotics motion planning community. Our objective of building a network encoding the pathways of the cortical network is analogous to the PRM objective of constructing roadmaps containing feasible paths. We represent the network as a large-scale directed graph, and use L-systems and statistics data to 'grow' neurons that are morphologically indistinguishable from real neurons. Synapses are detected using geometric proximity tests.

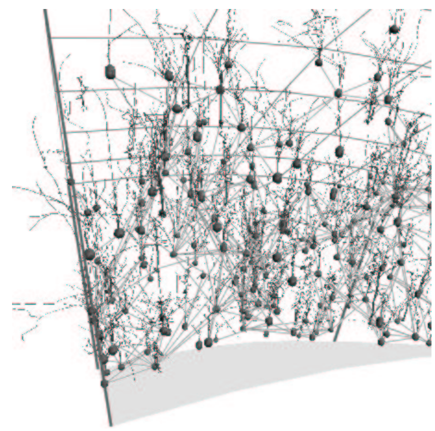


Figure 1: Synthetic neurons grown and connected in the cortex.

1 Introduction

The brain has extraordinary computational power which is determined in large part by the topology and geometry of its structures. A unique instrument developed at Texas A&M University, the Brain Tissue Scanner (BTS) [10], will enable an entire mouse brain to be imaged and reconstructed at the neuronal level of detail. In terms of 3D visualization, this project is the microscopic counterpart of the Visible Human Project at a vastly expanded scale. Moreover, by enabling studies of the topology of cortical networks, it could provide insight into one of the least understood biological processes — neural computation.

Using destructive sectioning and cross-sectional imaging, the BTS can scan an entire transgenic GFP/XFP-stained mouse brain in approximately one month. The data produced by the BTS will be used to reconstruct the three-dimensional structure–the neural forests and fibers–of the scanned tissue. However, only a small percentage (less than 10%) of neurons will be stained, so the neurons reconstructed from BTS data will be augmented with synthetic neurons that are grown based on the measured biological neurons. Next, their interconnections (synapses) will be generated. In the mouse cerebral cortex, there are about 16×10^6 neurons, with the average neuron receiving input from, and sending output to, 7800 neurons. The geometry and connectivity of the resulting cortical network will be studied theoretically and through simulation on massively parallel machines.

The work presented here is part of a larger project. Here, we start with geometric models of reconstructed neurons, and concentrate on the generation and connection of anatomically realistic synthetic neurons needed to complete the cortical network. A simple prototype system has been implemented. Neurons are generated from L-system-based neuron models, whose rules are computed from a statistical analysis of reconstructed actual neurons. In this paper, we describe techniques for identifying geometrically and anatomically realistic synapses.

2 Preliminaries

We are not aware of any work mapping the entire cortical network at the neuronal level of detail. Scientists, however, have made mappings of small areas of cortex.

Neuron models. Many different models of neurons have been proposed. Here we are interested in the morphological features of neurons and in their connectivity which will be needed to simulate their function.

The GENESIS (GEneral NEural SImulation System)

*This research supported in part by NSF CAREER Award CCR-9624315, NSF Grants IIS-9619850, ACI-9872126, EIA-9975018, EIA-0103742, EIA-9805823, ACI-0113971, CCR-0113974, EIA-9810937, EIA-0079874, and by the Texas Higher Education Coordinating Board grant ATP-000512-0261-2001. Morales is supported in part by a Fulbright/Garcia Robles (CONACYT) fellowship.

system [4] is based on the Hodgkin-Huxley model that describes quantitatively the dynamic behavior of the neuron membrane [6]. It can be used to model neurons at many levels of detail, from single compartments of a neuron to large networks of cells. Its main applications so far have been the modeling of single cells, although it has also been used for large scale studies such as constructing models of the hippocampal formation with about 1000 cells or studying the steady-state activity of a circuit of the granular layer of the rat cerebellum.

In 1994 McCormick and Mulchandani proposed the use of L-systems to represent neuron morphology [11]. These systems, which had been used to model plants, consist of a set of rules that describe the set of movements that a turtle-like device would have to do to draw the plant or neuron by displacements and changes in orientation. In 2000, Ascoli and Krichmar released L-Neuron which uses an L-system to generate and study anatomically correct neurons [2]. L-Neuron stochastically samples parameters from experimental distributions stored in a neuroanatomical database to be used in the generation of a virtual neuron.

Probabilistic Roadmaps. The Probabilistic Roadmap method (PRM) is a framework originally developed to compute collision-free paths for robots [8], but it has been shown to be useful in applications ranging from robotics to computational Biology (such as protein folding [14]). This framework has evolved and many methods have been proposed for particularly difficult situations [1, 3, 7]. PRMs involve the construction of a roadmap in which a set of configurations in C-space [9] are generated and connected. This roadmap can be queried for paths between given configurations. The application of this framework to construct a cortical network is described in the next section.

3 Overview of Our Approach

Our ultimate goal is to map and understand the connectivity and geometry of the cortical network. We begin our modeling by partitioning the cerebral cortex into a set of finite elements (FEs). Each FE has a set of geometrical models of reconstructed neurons which are used to generate additional anatomically realistic synthetic neurons. The cortical network is completed by making connections or synapses between the neurons. The construction process borrows from PRM (Probabilistic Roadmap) techniques by using simple, local techniques to understand a much larger unknown space viewed as a robotics problem.

A prototype system has been implemented. The general system structure is described in Figure 2. During *neuron generation*, neurons are "grown" in the *cortex* using information describing the spatial distribution of neurons and their geometry contained in the cortex FE model. The neurons generated are stored in a neuron data base, where a neuron is associated with the FE containing its soma. During *synapse generation*, neurons are connected using geometric proximity tests. Finally, a roadmap representing cortical connectivity is ready for analysis and simulation.

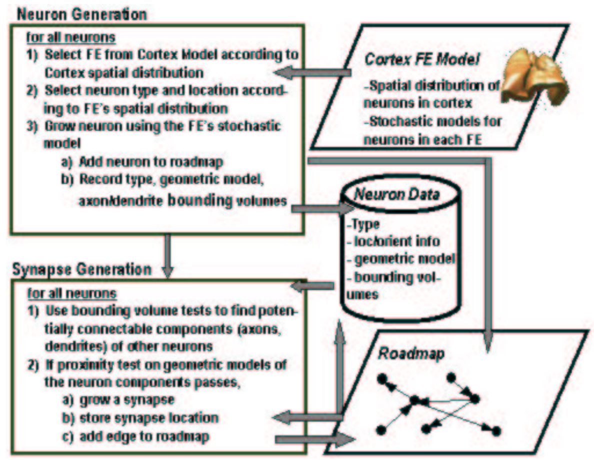


Figure 2: Framework of neuron PRM: Finite elements, Neuron generation/connection, roadmap graph, and database.

Figure 3: (a) A simple neuron. (b) Spines on a neuron.

In the following sections we show how we use the basic building blocks to construct a virtual cortical network. In Figure 2, the interactions and data streams between the elements described above are shown. We now describe some of the major data structures and models of the method.

A Neuron is represented as a tree of neural components, a soma, and a set of arbors. Each arbor has as children a set of segments who in turn have children that are either junctions (when they branch) or termination marks (when they are terminal segments). Each segment has a list of micro-segments which may or may not have a spine. Figure 3(a) shows a simplification of a neuron with a soma and two arbors. One of the arbors has no branches while the other has a junction with two branches. Its micro-segments have no spines. Figure 3(b) shows a neuron with spines.

Since neurons may be composed of tens of thousands of segments and may have thousands of synapses, the storage required to record the pre-synaptic and post-synaptic endings in the cortical network can be massive.

Finite Element Model of the Cortex. The cerebral cortex is partitioned into Finite Elements (FEs) which contain *local information* about types, amount, and spatial distributions of neurons in that FE (see upper-right corner in

Figure 2). The FEs are organized into an adjacency graph where each vertex is an FE and each edge represents an incident face between two FEs. As described in Section 5, this graph helps to speed up the process of finding connections between neurons.

Databases contain information for growing and distributing neurons (see center part of Figure 2). There are three types of databases: (i) types of neurons and probability for each type, (ii) Spatial distribution of each type of neuron, and (iii) Statistics data for neuron model.

The information in the first two databases is different from one FE to another. The third database is associated with each neuron type, and its data is obtained by a statistics calculator that is described in section 4.2.

A Roadmap Graph (lower-right part of Figure 2) is a directed graph representing an abstract cortical network. Following a common schema used in the PRM method, Neuron-PRM (N-PRM) samples graph nodes and connects nodes using local information. Each vertex in the graph defines a configuration of a neuron and each *directed edge* represents an abstract connection or synapse between neurons. Here, we define an abstract synapse as a set of real synapses from one neuron to another. Since synapses can be complex, these abstract synapses help maintain data more easily and also provide a hierarchical representation for searching the cortical network.

4 Generation of Neuron Nodes

Neuron PRM samples points and grows synthetic neurons during the neuron generation phase. Information gathered in the sampling stage is sent to the neuron generator to create morphologically correct neurons. Therefore, a vertex in the N-PRM roadmap (cortical network) contains a sampled point that represents the geometry and additional information used to determine the morphology of the neuron. We first describe how points are sampled and then how they are used to generate synthetic neurons.

4.1 Abstract neuron nodes

We generate points based on some given statistical distribution for neurons and neuron types. For example, statistically, very few neurons are found close to the upper layers of the cortex, and neuron types and numbers vary from layer to layer. We can use data from the literature or eventually from a system, such as the BTS described in Section 1, which can give information regarding the distributions and types of neurons from a particular brain.

The *abstract neuron generation* starts by selecting a FE randomly. The probability of selecting a FE depends on neuron population densities in the cortex; ultimately these will come from BTS data, but currently depend on FE size. Next, the position of the neuron is determined based on the distribution and statistics for its type in the FE's database. Finally, having all this abstract information, we are able to create synthetic neurons as described in Section 4.2. An outline of the algorithm for generating neurons is in the upper-left corner in Figure 2. To illustrate how neuron nodes are generated in the cortex, we constructed one half of the cortex of a mouse (see Figure 4)

4.2 Growing synthetic neurons

N-PRM calls a *neuron server* each time it has to grow a synthetic neuron. The server is provided with the position of the neuron to be created and the finite element that will contain its soma. The *neuron server* determines the type of neuron to be generated and creates a proper *neuron generator* to grow one (if one doesn't already exist.) The *neuron generator* creates the neuron and stores it so it can be used in later steps such as the connection of the neurons and in the future the simulation of the dynamic properties of the cortical network.

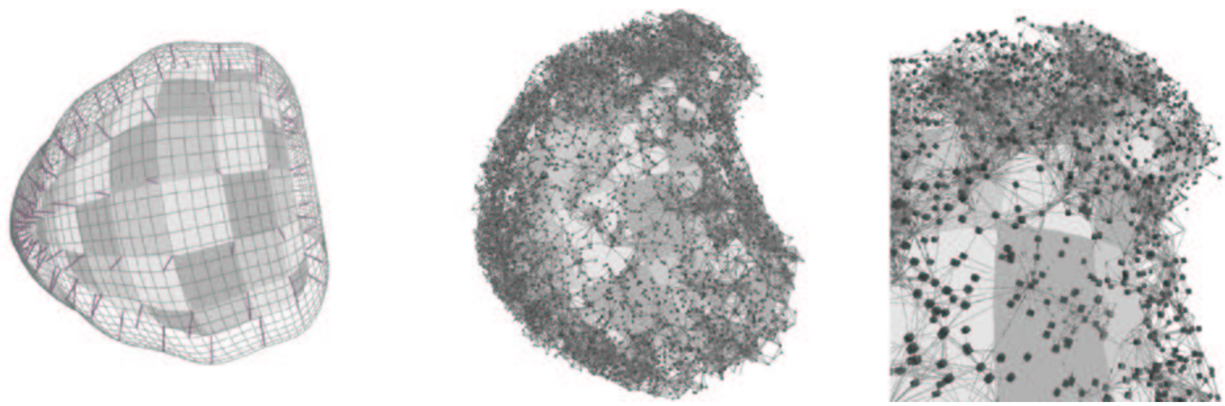
(a) (b)

Figure 5: Synthetic neurons grown using data on [5, 12, 13]: (a) based on files n400.swc to n423.swc (pyramidal cells), (b) based on files n220.swc to n275.swc (granule cells and interneuron).

Neuron Generators. The *neuron generator* can generate multiple types of neurons. It has a set of databases with statistics about each type of neuron. It picks the proper database and uses it to create each component of the neuron with an L-system grammar. Our system is like those used in [11] and [2].

The generator maps the string generated by the grammar into the tree that models the neuron. These neurons are used later in the connection stage. They also can be displayed or used by the statistics calculator, and in the future to simulate the dynamic properties of the network.

Statistics Calculator. The statistics calculator is responsible for measuring real neurons and generating the



(a) (b) (c)

Figure 4: (a) Empty cortical model of mouse. Each finite element colored with different grays. (b) Cortical model filled with abstract neuron nodes and connections. (c) A closer view of cortical network.

databases used by the *neuron generator*. Data for real neurons can be obtained from several public databases released by researchers in different electronic formats. Currently we use the SWC format also used by the database in [5]. This format describes a neuron in terms of linked nodes in a plain text file in which each line defines a compartment of the neuron in the following format:

`id type x y z radius parent_id`

The `id` is used as a way to link different compartments (through their parent `id`), the `type` differentiates compartments from the soma, and can be defined by the producer of the file.

The statistics calculator loads a sample of neurons in SWC format into our tree-like representation. Then the statistics calculator measures their diameters, lengths, orientations, amounts and rates for each interesting part of the neuron. Finally the maximum, minimum, range, average, and standard deviations are computed and stored in XML format to be used in the generation stage. In Figure 5 we can see two synthetic neurons that were generated based on two different sets of neurons of the hippocampus of the rat found on [5, 12, 13].

5 Connection of Neurons

After the FE model has been populated with neurons, that is, after the spatial distribution and stochastic geometric models of the neurons and fibers are ready, the cortical network is assembled by identifying potential synapses and connecting the neurons.

Statistically each neuron has thousands of synapses. An adult human being has more than one hundred billion neurons and even the mouse has more than sixteen million neurons. Currently, it is not feasible to compute and store such a huge number of synaptic connections. The usual PRM connection strategy of computing all pairwise distances and attempting to connect the k-closest nodes is not feasible for roadmaps with millions or billions of nodes, each of which consists of thousands of segments. To deal with such large numbers of complex nodes, we define simple distance metrics to reject neuron pairs and find potential synapses very quickly.

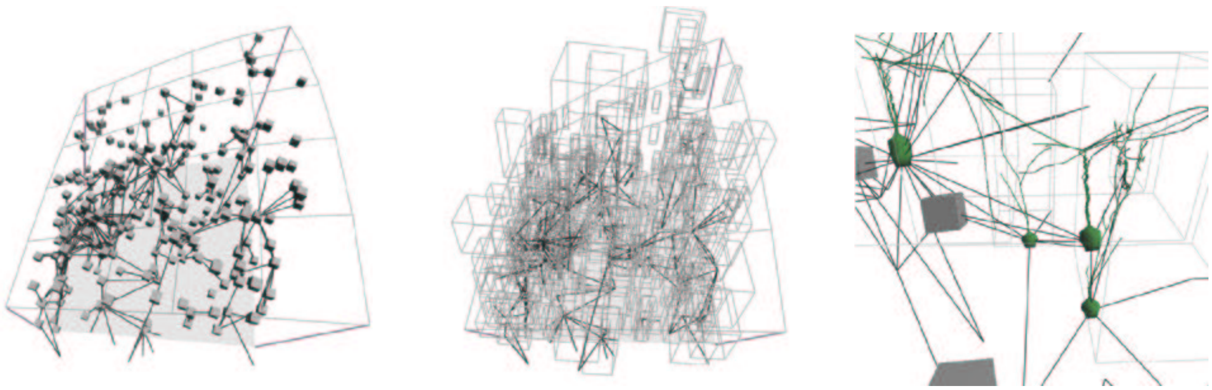
To reduce the number of connection tests, we only attempt to connect a neuron to neurons in the FEs which that neuron intersects. Moreover, most neuron pairs are quickly rejected by a filtering test which checks for intersection of their bounding volumes. Thus, detailed distance computations between the neuronal segments will only be performed for those neuron pairs that pass the bounding volume test.

Three different bounding volumes were tested, bounding sphere, bounding box, and convex hull. The intersection tests for bounding spheres and boxes can be computed very quickly, while the time for convex hulls depends on the number of vertices in the hulls. Note that the edges of the roadmap are directed, although the bounding volumes contain no directional information. Directional connections will be made during synapse discovery.

Neuron connections based on intersections of the neurons' bounding boxes are shown in Figure 6. Figure 6(a) shows the positions of the neurons and the connections. The denseness of the network is seen in Figure 6(b), and a close-up view of the symbolic connections between the somas of synthetic neurons is shown in Figure 6(c).

Next, a method using detailed segment information will more precisely compute synapse locations. We implemented a simple algorithm based on tendencies known for (real) synapse generation. In particular, we use the fact that synapses are usually found between segments and spines (see Figure 3(b)). The synapse discovery algorithm is described in Figure 7.

The synapse gap in line 5 is a user specified value which defines the maximum gap size between the pre-synaptic



(a) (b) (c)

Figure 6: (a) FE extracted from mouse cortex. (b) Bounding boxes of neurons in FE. (c) Connection of synthetic neurons.

SYNAPSE DISCOVERY

```
for (each neuron, n1)
 for (each neighbor neuron, n2, of n1 )
  for (each microsegment, m1, on n1 and, m2, on n2)
     Compute distance from endpoint of m1 to spine of m2
     if ( Distance < Synapse Gap )
       Report (m1,m2) as a synapse
```

Figure 7: Simple synapse discovery algorithm.

and post-synaptic sites. In current work we are adding more sophisticated criteria to our synapse discovery algorithm.

6 Experiments

To test our framework, we generated neurons in the mouse cortex. The model of the mouse cortex shown in Figure 4 was built in Maya and has 112 FEs. Each FE is modeled as upper and lower NURBS patches with 16 control vertices on each patch. To fill this model, 40,000 neurons, both abstract and synthetic neurons, were generated and distributed in the mouse cortex. Generation of these nodes took an average of 476.55 seconds over 10 runs. For the connection phase, we first tested bounding sphere, bounding box. and convex hull bounding volumes, with 40,000 neuron nodes. Data about the running times and the discovered edges (abstract synapses) is collected in Table 1.

The running time for building these volumes, as expected, was much faster for the bounding sphere and the bounding box than for the convex hull. Since the bounding sphere has larger volume than the bounding box and the convex hull, there were many more potential connections identified using the spheres than were found using the other two methods. Also, due to the overly conservative bounding sphere, almost all the neurons were contained in one connected component while almost a third of the neurons are isolated using the convex hull method.

After constructing the roadmaps with abstract synaptic connections, we applied our simple synapse discovery algorithm to identify real synapses. We used 0.015 as the synaptic gap size (neuron height is between 0.3 and 0.9). Some results are shown in Table 2. The roadmaps from the bounding sphere, the bounding box, and the convex hull are tested separately. The running time for discovery is linear in the number of edges of the input roadmap, so the convex hull is the fastest. Note that the differences in the number of synapses found is very small between the three roadmaps. This is because many of the abstract synapses found with the bounding sphere and bounding box methods contain very few, or no real synapses. The average number of synapses between two connected neurons, and the error rate illustrate this point. The error rate is:

$$\frac{N_{\mathrm{empty-abs-syn}}}{N_{\mathrm{total-abs-syn}}} \qquad (1)$$

Here, $N_{\mathrm{empty-abs-syn}}$ is the number of abstract synapses which did not contain any real synapse, and $N_{\mathrm{total-abs-syn}}$ is total number of abstract synapses in the given roadmap. This measure gives us an indication of the accuracy of the bounding volume methods. Although all methods have error rates higher than 50%, we believe this will decrease when we are able to more densely pack the cortex with neurons. Here, we generated only 40K and in reality there are some 16M neurons in the mouse cortex.

7 Conclusion and Future Work

In this paper we presented a prototype system that will eventually be used to (re)construct an entire mouse cortical network containing on the order of 16M neurons. Here,

Bounding volume strategies (Node Connection)			
Method	**Bounding Sphere**	**Bounding Box**	**Convex Hull**
Time for creating bounding volume	23.5 sec.	22.3 sec.	222.1 sec.
Time for connection	271.4 sec.	207.3 sec.	1,682.5 sec.
Number of edge found	1,353,083	307,494	87,110
Average node degree	33.8	7.7	2.2
Max node degree	374	100	38
Number of CC (Connected Component)	15	2,878	15,562
Number of Small CC (less than 4 nodes)	13	2,777	14,86
Number of isolated neuron	12	2,358	13,055
Max CC Size	39,982	35,941	5,957

Table 1: Bounding volume strategy labels for connecting 40000 neurons.

Cortical Network Refinement (Synapse Discovery)			
Input roadmap from	**Bounding Sphere**	**Bounding Box**	**Convex Hull**
Time for finding synapse	36,896.6 sec.	9,838.6 sec.	3,481.2 sec.
Number of synapse found	417,277	416,895	400,065
Max Number of synapses in single neuron	267	267	267
Avg Number of synapses between two neurons	0.3	5.5	16.0
Max Number of synapses between two neurons	241	241	241
Error Rate	97.4%	88.0%	61.5%

Table 2: Synapse discovery for connecting 40000 neurons.

we built a coarse cortical network using N-PRM and L-System neuron generators. We tested three bounding volume methods for synapse identification. We determined that the approximate convex hull bounding volume was the fastest overall (considering both the cost of the abstract and the actual synapse discovery phases); its drawback is the increased storage requirements for the convex hull as compared to the bounding sphere or box.

References

[1] N. M. Amato, O. B. Bayazit, L. K. Dale, C. V. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, pages 155–168, 1998.

[2] G. Ascoli and J.L. Krichmar. L-Neuron: a modeling tool for the efficient generation and parsimonious description of dendritic morphology. *Neurocomputing*, 2000.

[3] V. Boor, M. H. Overmars, and A. F. van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 1018–1023, 1999.

[4] J.M. Bower and D. Beeman. *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System*. Springer-Verlag, 1998.

[5] Duke/Southampton. Archive of neuronal morphology. `http://www.cns.soton.ac.uk/~jchad/cellArchive/cellArchive.html`.

[6] A. Hodgkin and A.Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.*, 117:500–544, 1952.

[7] D. Hsu, L. Kavraki, J-C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, 1998.

[8] L. Kavraki, P. Svestka, J. C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Automat.*, 12(4):566–580, August 1996.

[9] T. Lozano-Pérez and M. A. Wesley. An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacle. *Communications of the ACM*, 22(10):560–570, October 1979.

[10] B.H. McCormick. Brain tissue scanner enables brain microstructure surveys. *Neurocomputing*, 2002, In press.

[11] B.H. McCormick and K. Mulchandani. L-system modeling of neurons. In *Proc. Visualization in Biomedical Computing, SPIE Vol. 2359*, pages 693–705, 1994.

[12] D.D Mott, D.A. Turner, M.M Okazaki, and D.V. Lewis. *J. Neurosci.*, 17:3990–4005, 1997.

[13] Papaly et al. *J. Comp. Neurol*, 391:335–352, 1998. Recorded in G. Buszaki's laboratory at Rutgers University and reconstructed at Duke University.

[14] G. Song and N. M. Amato. A motion planning approach to folding: From paper craft to protein folding. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 948–953, 2001.