Neuron PRM: A Framework for Constructing Cortical Networks [1]

Jyh-Ming Lien, Marco Morales, and Nancy M. Amato

PARASOL Lab, Department of Computer Science, Texas A&M University,
{neilien,marcom,amato}@cs.tamu.edu

Abstract

The brain has extraordinary computational power to represent and interpret complex natural environments is essentially determined by the topology and geometry of the brain's architectures. We present a framework to construct cortical networks which borrows from probabilistic roadmap methods developed for robotic motion planning. We abstract the network as a large-scale directed graph, and use L-systems and statistical data to 'grow' neurons that are morphologically indistinguishable from real neurons. We detect connections (synapses) between neurons using geometric proximity tests.

Key words: Cortical Networks. PRM. BTS. L-System. Rectangle Tree.

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) [5], will enable an entire mouse brain to be imaged and reconstructed at the neuronal level of detail. 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 will be used to reconstruct the three-dimensional neuronal structure of the scanned tissue. While scientists have mapped small areas of cortex, we are not aware of any work mapping the entire

cortical network of any mammalian species at the neuronal level of detail. Since only a small percentage (less than 10%) of neurons will be stained, 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. The geometry and connectivity of the resulting cortical network will be studied theoretically and, in collaboration with the IBM Blue Gene project, through simulation on massively parallel machines.

In this paper, we start with geometric models of reconstructed neurons, and concentrate on the generation and connection of the anatomically realistic synthetic neurons needed to complete the cortical network. Our general strategy borrows from the *probabilistic roadmap methods* (PRMs) [4] first developed for robotic motion planning, but which have been applied to applications ranging from robotics to computational biology [7]. PRMs construct a network (roadmap) in which nodes represent feasible configurations (neurons) and edges represent valid connections (synapses). In this paper, we use L-systems [6,1] to grow synthetic neurons and describe techniques for identifying geometrically and anatomically realistic synapses. A simple prototype system, called NEURON PRM (N-PRM), for growing a synthetic cortical network has been implemented.

2   Neuron PRM Framework

Our ultimate goal is to map and understand the connectivity and geometry of the cortical network. The construction process borrows from the PRM framework by using simple, local techniques to construct a representative model of a much larger unknown space. We begin our modeling by partitioning the cerebral cortex into a set of finite elements (FEs). The general system architecture is described in Figure 1(b). 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 the FE's neuron database. During *synapse generation*, neurons are connected using the geometric models constructed during the *neuron generation*, and their known affinities for synapse creation. Finally, a roadmap representing cortical connectivity is ready for analysis and simulation.

We now briefly describe the basic building blocks of our method: neurons, FE model, statistical databases, and the roadmap graph. A *neuron* is represented as a tree of neural components. 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. The *finite element model of the cortex* partitions the cerebral cortex into finite elements (FEs) which contain local information about types, population distribution, and spatial distributions of neurons in that FE (upper right corner in Figure 1(b).) FE models also help in visualization and speed up the process of finding connections between neurons. Several *databases* contain information for growing and distributing neurons (see center part of Figure Figure 1(b).)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) statistical data for the neuron model. The information in the first two databases is unique to each each FE, and so is stored with the FEs. The
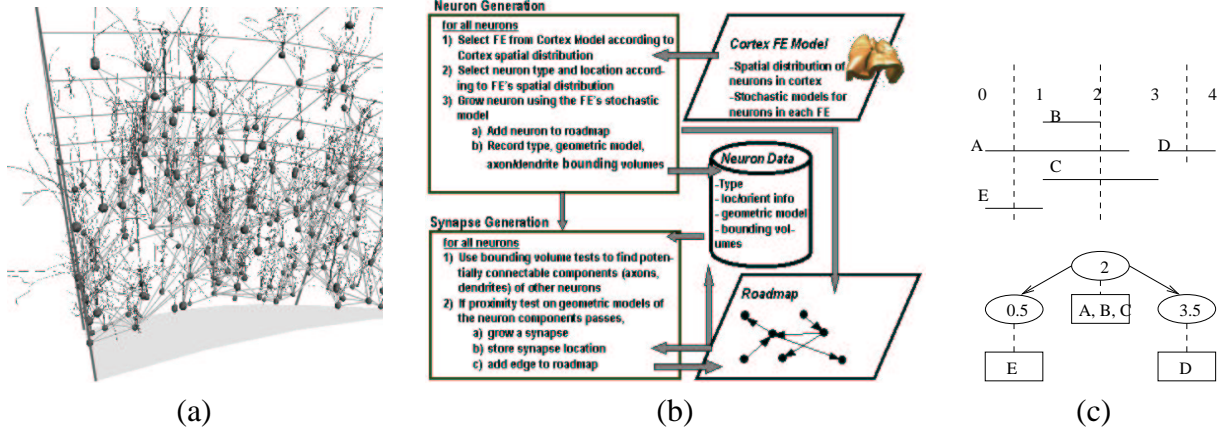
Fig. 1. (a) Synthetic neurons generated and connected in the cortex. (b) Neuron PRM Framework: Finite elements, Neuron generation/connection, roadmap graph, and database. (c) A 1-fold rectangle tree.

third database is associated with each neuron type. A *roadmap graph* (lower-right in Figure 1(b).) is a directed graph representing an abstract cortical network. Each vertex in the graph corresponds to a neuron and each directed edge represents one or more connections (or synapses) between two neurons. Thus, the roadmap provides a hierarchical representation facilitating analysis, simulation and visualization of the cortical network.

Generation of Neurons. 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. In [6] the use of L-systems which had been used to model plants was proposed to represent neuron morphology. L-Neuron [1], developed by Ascoli and Krichmar, uses an L-system to generate neurons. It stochastically samples parameters from experimental distributions stored in a neuroanatomical database to be used in the generation of a virtual neuron.

Neuron PRM uses a similar approach to grow neurons. An abstract neuron contains information regarding position, type, associated FE, and a unique ID. Position and type are generated randomly based on some given statistical distribution stored in the FE's database. Next, an L-system is used to create a morpholigically correct synthetic neuron based on the information contained in the abstract neuron and on statistics provided by the database in the host FE for that neuron type. Currently, the statistics are calculated from a sample of SWC neurons publically available [2]. In the future, we will use reconstructed neurons provided by the BTS to extract statistics. An abstract neuron and synthetic neuron pair are stored as a vertex of the cortical network (see Figure 2(a,b,c).)

Generation of Synapses. After the FE model has been populated with neurons as described above, the cortical network is assembled by identifying potential synapses and connecting the neurons. Statistically each neuron has thousands of synapses and there are on the order of sixteen million neurons in the cortex of the mouse. Thus, a brute force strategy which tests all pairs of neurons is not feasible. To address this issue, we define simple metrics to reject neuron pairs and identify potential synapses very quickly. 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. When the intersection of a pair of neurons is discovered, we connect their associated vertices
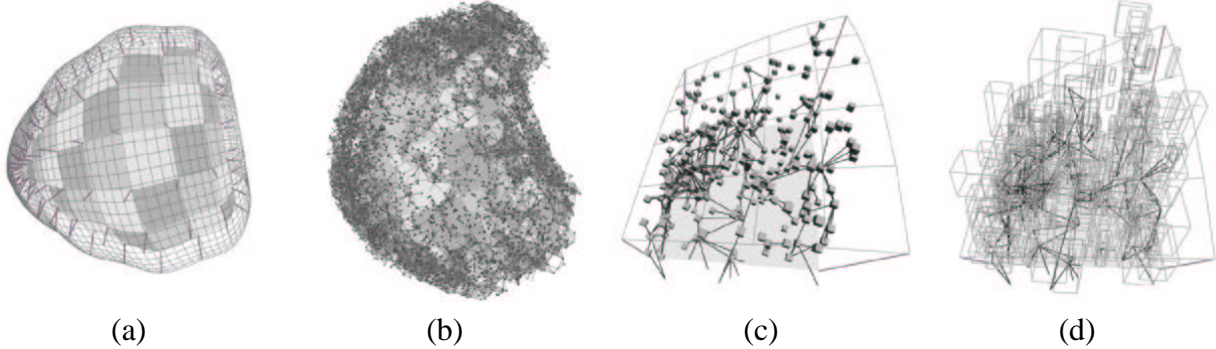
Fig. 2. (a) Empty cortical model of mouse. (b) Cortical model filled with abstract neuron nodes and connections. (c) A closer view of cortical network. (d) Neurons in one FE surrounded by bounding volumes.

Bounding Volume Tests (Node Connection)					
		Bounding Box			
Method	Bounding Sphere	Brute Force	RT Naive	RT Improved	Convex Hull
Time for creating bounding volume	47.37 sec	47.53 sec.	47.88 sec.	47.93 sec.	56.89 sec.
Time for connection	344.46 sec	360.1 sec.	93.17 sec.	64.1 sec.	604.23 sec.
Number of edges found	59,158	19,994	19,994	19,994	6,094
Average node degree	11.8	3.9	3.9	3.9	1.2
Connected Components (CC)	16	709	709	709	5,089
Isolated neurons	14	540	540	540	3,440

Table 1
Bounding volume tests for identifying (potential) connections for 10,000 neurons.

in the roadmap (Figure 2(d)). Three different bounding volumes were tested: spheres, boxes, and convex hulls. The bounding sphere and bounding box provide fast construction and intersection tests, but convex hull is more accurate in terms of approximating the shape of the target neuron. The strategy sketched above performs $O(n^2)$ bounding volume intersection tests. For the bounding box representation, we reduced this to $O(n \log n + k)$, where k is the actual number of intersections, using a method based on Edelsbrunner's *rectangle tree* [3] (see Fig. 1(c)). We implemented two versions of the rectangle tree query, one described by Edelsbrunner [3] that initiates each query from the root of the tree, and a variation we designed to solve the *all-intersection query* in a more efficient way.

After a neuron pair's bounding volumes are found to intersect, a more detailed computation 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. That is, for each pair of connected neurons, we compute distances from spines of one neuron to the segments of the other neuron. If the distance is less than a user specified value for synapse gap, this spine, segment pair is marked as synaptic site.

3   Experiments

To test our framework we generated a model of the mouse cortex (Figure 2). A summary of the results is shown in Table 1. The running time for building these volumes, as expected, was much

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
Avg # of synapses between two neurons	0.3	5.5	16.0
Error Rate	97.4%	88.0%	61.5%

Table 2
Synapse discovery for 40,000 neurons.

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. For the bounding box tests, we found that the versions using the rectangle tree take 16%-25% of the time required for the brute force method, depending on whether the standard query or the optimized query for computing all intersections was used.

For the connection phase, we tested 10,000 neurons with the bounding sphere, bounding box, and the convex hull. The bounding box was tested using the brute force approach, the original version of the rectangle tree, and our optimization to compute all intersections. To identify real synapses, we used 0.015 as the synaptic gap size (neuron height is between 0.3 and 0.9). Some results for 40,000 neurons are shown in Table 2. The roadmaps from the bounding sphere, the bounding box, and the convex hull were 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. Thus, since synapse identification requires both abstract connection and synapse discovery, the convex hull method is the most efficient – the first phase is more expensive, but it pays off by filtering out more neuron pairs for the second phase, in which each test is significantly more expensive. The error rate in Table 2 is defined as : $\frac{N_{\mathrm{empty-abs-syn}}}{N_{\mathrm{total-abs-syn}}}$ where $N_{\mathrm{empty-abs-syn}}$ 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 40,000 and in reality there are some 16,000,000 neurons in the mouse cortex.

4   Conclusion

We have developed a prototype system that will eventually be used to (re)construct an entire mouse cortical network containing over 15 million neurons. Our PRM-based framework, N-PRM, for constructing a hierarchical model of the cortical network uses L-System neuron generators. In this paper, we study various bounding volumes for reducing the cost of synapse identification. We found that data structures, such as the rectangle tree, which use spatial information to restrict the number of intersection tests, can lead to significant improvements in execution time. In future work, we will further investigate our proposed optimization of the rectangle tree, including its applicability

to additional bounding volumes.

References

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

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

[3] H. Edelsbrunner. A new approach to rectangle intersections. *Int. J. Comp. Math.*, 13:209–229, 1983.

[4] 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.

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

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

[7] 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.

Jyh-Ming Lien is a Ph.D. student working with Dr. Nancy Amato in the PARASOL Lab in the department of Computer Science at Texas A&M University. He received his B.S. in Computer Science from National ChengChi University in 1999.

Marco Morales is a Ph.D. student working with Dr. Nancy Amato in the PARASOL Lab in the department of Computer Science at Texas A&M University. He received his B.E. in Computer Engineering and his M.E. in Electrical Engineering from the Universidad Nacional Autónoma de México in 1996 and 1998 respectively. His main interests include motion planning, robotics, computational neuroscience, and computational geometry.

Nancy M. Amato is an Associate Professor of Computer Science at Texas A&M. She received her M.S. and Ph.D. in Computer Science from UC Berkeley and the University of Illinois at Urbana-Champaign, resp. She is an Associate Editor of the IEEE Transactions on Robotics and Automation and of the IEEE Transactions on Parallel and Distributed Computing. Her research interests include motion planning, parallel computing, and computational biology, geometry, neuroscience, and physics.