Realistic Modeling of Brain Structures with Remote Interaction Between Simulations of an Inferior Olivary Neuron and a Cerebellar Purkinje Cell

Jason Leigh, Erik De Schutter, Maurice Lee, James M. Bower, Thomas A. DeFanti
Electronic Visualization Laboratory at the University of Illinois at Chicago
and the California Institute of Technology

ABSTRACT

This paper discusses the modeling and visualization of the synaptic interaction between two mammalian neurons. The olivary neuron makes a climbing fiber synapse on the cerebellar Purkinje cell. This is a very strong synapse, which evokes a typical complex response involving the opening of calcium channels in the Purkinje cell dendrite. Both neurons were modeled by realistic compartmental models (about 500 compartments for the olivary neuron and 4500 for the Purkinje cell), with voltage dependent ionic channels in both the soma and dendrites. This included the modeling of submembrane calcium concentrations. The complexity of the models, required that we simulate them on separate processors; an Intel Touchstone Delta (a parallel supercomputer) performed all computations for the Purkinje cell simulation, and a unix workstation for the olivary neuron. The synaptic interaction was carried as a boolean signal over a T3 network communication line, between the Delta at Caltech and the unix workstation in Chicago. A high performance graphics workstation was used to provide interaction with the simulations as well as visualization for the collected results. A prototype of this system was demonstrated at Showcase'92 at the SIGGRAPH'92 conference. In addition, a visualization of this interaction was demonstrated in the CAVE virtual reality environment at the SIGGRAPH'92, Supercomputing'92, and RSNA'92 conferences. The goal of this research is to allow computational neuroscientists to connect neuronal models at geographically distant locations while providing interactive visualization of the results.

1 INTRODUCTION

In recent years the greater availability of workstations has resulted in significant increases in modeling in many scientific disciplines. In Computational Neuroscience, there has been an increase in the number, and complexity of models of single neurons, and neural networks (Bower and Koch 1992). Modeling is attractive because it provides a deeper understanding of what is still unknown about the system, and thus helps us to guide our experiments so that we avoid generating massive amounts of unconnected and uninterpretable data (Bower 1992).

In this paper we will discuss modeling of two neurons (the Purkinje and the Inferior Olivary Neuron) in the cerebellum. Up until now, modeling of single neurons and neural networks has been relegated to single workstations or parallel computers. Our work involves the interaction between, and visualization of two neurons at geographically distant locations, computed on heterogeneous computer systems. At Caltech, the Intel Touchstone Delta (a roughly 500 processor MIMD supercomputer) simulated a cerebellar Purkinje cell, while in Chicago, a Sun Sparc station simulated an inferior olivary neuron. The two neurons were connected via a T3 line through Argonne National Laboratory, and all data generated during the interaction was simultaneously visualized on a Silicon Graphics Crimson on the exhibit floor of SIGGRAPH '92, in Chicago. This prototype is the first application of the Metacomputing paradigm (Smarr and Catlett 1992) in Computational Neuroscience.

We will begin by describing the mathematical model for characterizing neuronal compartments, and then describe the Purkinje and Inferior Olivary neurons themselves. In addition we will discuss the issues involved in parallelizing and visualizing neuronal simulations.

1.1 The Mathematical Model

Cable Theory and the application of its equations describes the flow of current in an electrical cable. Similarly, it characterizes the flow of heat in a rod and the diffusion of substances in a solute (Shepherd 1976), but more profoundly to neuroscientists, it describes the passage of current in dendritic neurons. The systematic adaptation of the cable equations to dendritic neurons was first carried out by Rall (Rall 1957, 1959, 1960). The Cable Equation is a partial differential equation typically expressed as:

$$\lambda^2(\delta^2 V/\delta x^2) - V - \tau(\delta V/\delta t) = 0$$

where V represents the difference between interior and exterior membrane voltages; x represents the distance along the length of the membrane cylinder, and λ is the length constant of the core conductor. t represents time and τ is the membrane time constant (Rall 1989).

Although linear cable theory has made exact quantitative formulations of neurophysiological events feasible, its use in modeling real neurons is limited. This is because, for every electrical event, specific solutions of this equation have to be obtained. The complexity of the solution is very sensitive to the elaborateness of the model so that in practice only simple, passive models with simple branching structures can be handled (De Schutter 1989).

1.2 Compartmental Modeling

A recent extension to the cable theory, where the lengths of dendrites and their branches are decomposed into individual compartments, has made it unnecessary to describe an entire neuron with one complex equation (Segev, Fleshman and Burke 1989). Instead, each compartment is kept small enough so that they may be represented by only one simple equation. For passive compartments the equation is always the same. For excitable compartments, terms describing the ion conductances have to be included.

Diagram 1 shows the electrical circuits for passive and excitable compartments. Common to both types of compartments is the part of the circuit that describes current flow through the cell membrane and to other compartments over a cytoplasmic resistance R_i. The membrane assumes some capacitance CM, and there is a potential source E (which is the difference between the membrane potential and the resting membrane potential) coupled to a membrane resistance RM. Values for these parameters depend on the size of the compartment and on neuron specific, equivalent cable parameters (De Schutter 1989).

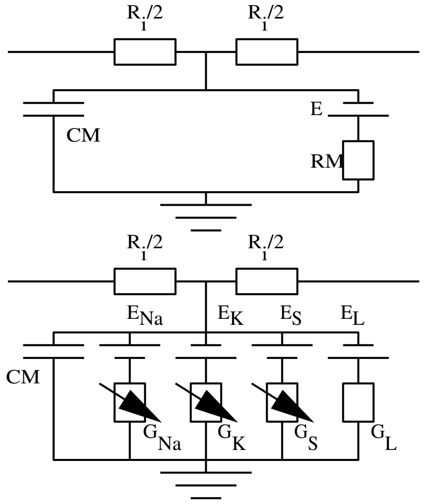

Diagram 1: Equivalent circuits for a compartment with passive (top) and excitable (bottom) membranes.

For excitable membranes, the circuit is somewhat more complex for it typically includes three basic categories of ionic channels: leak or passive, synaptic, and active channels (Segev, Fleshman, Burke 1989). The leak channel, is represented by a time and voltage independent transmembrane conductance G_L. The synaptic channels are represented by time-dependent but voltage-independent conductances $G_S(t)$; and the active channels are represented by both time- and voltage-dependent conductances. As in the Hodgkin-Huxley model of the squid giant axon (Hodgkin and Huxley 1952), our model includes active channels for Potassium and Sodium conductances (G_K and G_{Na} respectively).

The change in membrane potential δE in a passive compartment i connected to one or more compartments j, with a synaptic current $G_S(t)(E_i - E_S)$ and an injected current I_e is described by a single differential equation:

$$\frac{\delta E_i}{\delta t} = \frac{E_i}{RM_i CM_i} + \frac{\sum_j ((E_i - E_j)/R_{i,j})}{CM_i} + \frac{G_S(t)(E_i - E_S)}{CM_i} + \frac{I_e}{CM_i}$$

where $R_{i,j}$ is the axial resistance between compartment i and j.

In excitable compartments the first term in the above equation is replaced by:

$$\frac{\sum_u G_u(E_i, t)(E_i - E_u)}{CM_i}$$

where u enumerates all the active ionic channels (in our example K and Na channels). A summary of the properties of many different ionic channels may be found in Hille (Hille 1984; Yamada, Koch and Adams 1989; De Schutter 1989).

2 MODELING THE TWO NEURONS

Detailed computer models of neurons and of the synaptic interactions in neuronal networks have become an invaluable tool to neuroscientists. Models of cerebellar and olivary neurons can be useful in helping us understand the function of the cerebellum, as despite almost a century of research, the exact operation of this large part of the vertebrate brain remains a mystery.

2.1 Inferior Olivary Neuron Model

The inferior olivary nucleus is an aggregate of neurons on the ventral side of the brain stem. It is convoluted and parcellated into several subnuclei. The neurons of this nucleus seem to be extensively interconnected by means of low-resistance current shunts (gap junctions) between dendritic spines and branches. A single axon arises from each neuron and projects to the cerebellar cortex, where it divides into as many as ten branches, each contacting one Purkinje cell in the adult animal.

The model of a neuron from the inferior olivary nucleus used in this simulation consists of approximately 500 compartments. Five main dendritic trunks arise from the neuronal soma and subdivide to form the dendritic tree. Three active ionic conductances (two involved in initial spike generation and one involved in oscillatory activity) are simulated in the soma; each dendritic compartment contains two active conductances. The actions of these conductances contribute to the characteristic membrane potential response of inferior olive neurons upon stimulation, a fast initial spike followed by a longer depolarization and a still longer hyperpolarization, these latter two effects arising in the dendrites. Often this is followed by a rebound or oscillatory response.

The single-cell model is being used to study the effects of interconnecting networks of such cells through gap junctions on the synchronization and desynchronization over both space and time of activity in this nucleus.

2.2 Purkinje Cell Model

The cerebellar Purkinje cell is a unique neuron. It receives more synaptic inputs than any other neuron in the brain and has one of the most active dendrites, generating massive calcium spikes in response to climbing fiber synaptic input (Ito 1984). We simulated the Purkinje cell with a very detailed compartmental model (Rall. 1962). The model reproduced the exact morphology (with 1600 separate compartments) and included ten different voltage dependent channels (over 8000 in total in the model) and more than 3000 synaptic channels. The dendritic membrane contained two different types of calcium channels and two types of calcium activated potassium channels. We also modeled the submembrane calcium concentration. The model reproduced the response of Purkinje cells to current injection in vitro and to synaptic inputs in vivo completely. It has been shown to be robust to changes in channel densities and morphology. This model was run on the Intel Touchstone Delta to examine the response of the Purkinje cell to its numerous synaptic inputs from granule cells (De Schutter and Bower 1993). We found that the Purkinje cell is insensitive to the exact location of synchronous inputs in its large dendritic tree. This would not have been true if the dendrite had passive membrane, but in the Purkinje cell the P-type calcium channels differentially amplify distal inputs. Also, asynchronous excitatory and inhibitory inputs controlled how large the response of the Purkinje cell would be to any synchronous input. This effect was global (for synchronous inputs located anywhere) and occurred over a slow time scale, and hence the asynchronous input pattern in the preceding 10 to 20 milliseconds was important. Assuming that synchronous and asynchronous inputs to the Purkinje cell model correspond to different elements in the cerebellar circuitry, these results have profound implications for our understanding of how the cerebellar neuronal networks operate.

2.3 The Climbing Fiber Synapse

The Purkinje cell receives two kinds of synaptic inputs (Ito 1984). The parallel fiber inputs from cerebellar granule cells are numerous (up to 175,000) but have a small synaptic conductance. On the contrary, each Purkinje cell receives just one climbing fiber synapse from an olivary neuron. Activation of the climbing fiber synapse results in a massive response, called the complex spike. The complex spike is caused by the activation of calcium channels in the dendrite during a prolonged dendritic spike. The complex spike changes the immediate sensitivity of the Purkinje cell to other synaptic inputs and can induce long term depression of the synaptic conductance of parallel fiber inputs. In our model, the synaptic transmission is simulated as a simple boolean operation. A threshold element in the olivary neuron model detects action potentials, which are transmitted with a time delay to the Purkinje cell model where the climbing fiber synaptic channels are opened at the appropriate time. Figure 1 shows snapshots of the various stages of the complex spike event. The Purkinje cell is shown at the top and the olivary neuron is shown below it. The pseudocolor represents the membrane potential in each compartment, and the timer in each snapshot, shows the elapsed simulation time in seconds. Plots of the membrane potential at the soma for each neuron are also shown.

3 PARALLEL SIMULATION SOFTWARE

GENESIS (GEneral NEtwork SImulation System) is currently the only software that supports parallelized modeling of single neurons and networks on MIMD parallel computers (De Schutter 1992) and it is the system with which all our modeling work was performed. It was designed primarily for the interactive development of single neuron and neural network models. The system is an object-based interpreter which is extendible by the addition of C subroutines as new objects. All objects communicate with one another through message passing and each object contains sets of methods specific to that object; class inheritance is not supported.

The interpreter makes it simple for users to quickly develop and test neuronal models without having to engage in the tedious ritual of coding and compiling. However, the system is extendible by either writing procedures in the GENESIS scripting language or C. If C is used the code is encapsulated as a new GENESIS object and the entire GENESIS system is recompiled to produce a newer version. This was the mechanism with which the INET networking object (discussed below) was designed.

In addition to having a rich collection of objects for detailed neuronal modeling, GENESIS supports XODUS (X-based Output and Display Utility for Simulators), which provides users with easy-to-develop graphical interfaces for outputting results obtained during simulations, as well as inputting new parameters between simulations. However, since XODUS runs under GENESIS, its ability to offer high speed visualization is severely limited. Consequently external visualization tools (GENESIS Visualizer and GENESIS listener) have been developed to take advantage of currently available high performance graphics engines.

3.1 Simulating Neurons on Parallel Platforms

It is simply the case that there is no computer now conceived that is powerful enough to capture the complexity of realistic models of large neural networks. Hence parallelization of single or multiple neuronal models, on a single multiprocessor computer or over several workstations and multiprocessor computers, must be exploited.

3.2 Parallelizing Single Neuronal Models

A fundamental issue in optimally parallelizing single neuronal models is that of determining a distribution of the task that incurs the minimum communication overhead and load imbalance (Nelson, Fumanski and Bower 1989). However, these factors are related such that it is not possible to minimize both simultaneously (Fotouhi, Leigh, and Rana 1991). As a result we need to find some compromise between the two. Load imbalance can be described as:

$$LoadImbalance = \frac{T_{calc_{max}} - T_{calc_{avg}}}{T_{calc_{avg}}}$$

where $T_{calc_{avg}}$ is the average calculation time per processor. Communication overhead is then given by:

$$CommOverhead = \frac{T_{comm_{max}}}{T_{calc_{max}} + T_{comm_{max}}}$$

where $T_{comm_{max}}$ is the maximum communication time and $T_{calc_{max}}$ is the maximum calculation time for the busiest processor (Nelson, Fumanski and Bower 1989).

3.2.1 Estimating Load Imbalance

In order to estimate Load Imbalance for a given multicompartmental model we need estimates for $T_{calc_{max}}$ and $T_{calc_{avg}}$. $T_{calc_{max}}$ is defined as:

$$T_{calc_{max}} = \max_i T_{calc(i)}$$

and $T_{calc_{avg}}$ as:

$$T_{calc_{avg}} = \frac{1}{numProcrs} \sum_{i=1}^{numProcrs} T_{calc(i)}$$

where i represents the processor number and $numProcrs$ is the total number of processors utilized. $T_{calc(i)}$ is given by:

$$T_{calc(i)} = T_{calc_{update(i)}} + T_{calc_{comm(i)}}$$

where $T_{calc_{update(i)}}$ is the total calculation time devoted to updating processor i, which is dependent on the number of compartments assigned to that processor. $T_{calc_{comm(i)}}$ is the computation time involved in intercompartmental communication and is dependent on the number of messages that need to be processed.

An estimate on $T_{calc_{update(i)}}$ can be obtained by determining the time needed for a single processor to execute one simulation step for a single compartment and scaling this value according to the number of compartments assigned to processor i. If such timing information is not available then an estimate can be derived from analysis of the number of floating point operations involved in the simulation step. A similar analysis can be applied to estimate $T_{calc_{comm(i)}}$ with the exception that we scale the time taken to process a single message, by the number of expected messages. The number of expected messages may be constant or may vary with time, as in the case of communication between neurons via action potentials. This is further discussed by Nelson (Nelson, Fumanski, and Bower 1989).

3.2.2 Estimating Communication Overhead

An accurate estimate of the communication overhead for a model will require examination of the architecture of the parallel computer. A low communication overhead is generally preferred since we want all the processors to be able to compute continuously without blocking for incoming messages. A low communication overhead can be achieved if we distribute the compartments of the neuron so that most of the communication occurs within the processor while maintaining a low load imbalance.

In what follows we will use the Intel Touchstone Delta, a 576 processor MIMD mesh-connected parallel supercomputer, as the architecture to model our estimates, although the formulations can easily be generalized for other parallel architectures. Let $1 \leq XSize \leq 16$ represent one of the user designated dimensions of the Delta's mesh and let $1 \leq YSize \leq 32$ be the other. This only totals a maximum of 512 processors. The remaining 64 processors of the Delta are used to support I/O between disk drives, tape drives, ethernet and HIPPI network connections (Concurrent Supercomputing Consortium 1991). In anycase we designate $numProcrs = XSize \times YSize$ as the number of processors assigned to a simulation.

The routing of messages on the Delta is such that a message originating from a node at mesh coordinate (x_{start}, y_{start}), destined for a node at (x_{end}, y_{end}) first travels along row y_{start} of the mesh until $x_{start} = x_{end}$. Then it travels along column x_{start} until $y_{start} = y_{end}$. Hence the maximum distance traveled by any message is $XSize + YSize$ and the minimum is zero.

Nelson (Nelson, Fumanski, and Bower 1989) describes a Router algorithm in which connected compartments, that may or may not be distributed over different processors, communicate with one another by each sending messages to the others. The problem with this scheme is that it results in a rather large number of messages sent over the mesh, especially if connectivity between the compartments are high. An alternative scheme is to use what are called Postmasters (Nelson, Fumanski, and Bower 1989) and we will see later its benefits over the naive Router algorithm. Regardless of the algorithm the maximum communication time is clearly:

$$T_{comm_{max}} = \max_i T_{comm(i)}$$

where i is each processor out of $numProcrs$ processors, and $T_{comm(i)}$ is the total communication time for processor i.

For the Router algorithm $T_{comm(i)}$ is given by:

$$T_{comm(i)} = t_{commNTN} \sum_c Distance(c, i)$$

where c is each compartment in the set of compartments assigned to processor i and $t_{commNTN}$ is the time to pass a message between neighboring nodes. $Distance(c, i)$ is the cumulative distances traveled by all the messages m sent by compartment c from processor i:

$$Distance(c, i) = \sum_m |x_{end_m} - x_i| + |y_{end_m} - y_i|$$

Using the above formulation we can derive estimates for all the neuronal structures described by Nelson (Nelson, Fumanski, and Bower 1989).

3.2.3 Postmaster

Nelson (Nelson, Fumanski, and Bower 1989) points out that the Router algorithm is not particular efficient in terms of minimizing interprocessor communication time. The reason for this is that for each compartment c connected to compartments S on other processing nodes, $|S|$ messages must be sent.

An alternative is to use a Postmaster. In this scheme each compartment in a processor sends its messages to a postmaster on that processor. The postmaster gathers all messages destined for a particular processor into a single message and sends out only one message to that processor, hence each compartment sends at most $numProcrs - 1$ messages into the mesh (Nelson, Fumanski, and Bower 1989).

3.3 Algorithms for Efficient Compartmental Assignment in Branching Models

Now that we have some means of estimating load imbalance and communication overhead, we can discuss optimal means of assigning compartments to processors that minimize these factors. Fox (Fox 1987) describes a recursive bifurcation scheme for distributing the compartments in branching neuronal models. The dendritic tree of the neuron is repeatedly split into two approximately equal parts based on estimates of load imbalance and communication overhead, until there are as many parts as there are processors. The distribution created is not perfect, but as with most heuristic solutions, a near optimal solution is usually good enough.

Another solution which yields somewhat better distributions, is to interpret the load imbalance and communication overhead as an energy function. A downward descent method can then be used to find a minimum for this function. Fox and Furmanski (Fox 1988) have used a Hopfield-type network to generate solutions (Hopfield and Tank 1985) with generally better results than the bifurcation scheme. Additional heuristics such as simulated annealing (Kirkpatrick 1983) can be used to prevent the Hopfield net from settling in local minima.

For the Purkinje model, an algorithm similar to the bifurcation scheme was used. The load was calculated as a function of both the number of compartments and number of ionic channels in each compartment. Emphasis was given to finding a distribution that minimized the number of messages sent between compartments on different processors.

3.4 Networking Neuronal Models on Heterogeneous Computer Systems

It is conceivable that in the near future researchers will wish to develop more complex multi-neuronal networks spread over networks of computer systems. This will allow researchers at geographically distant locations to collaborate in their simulations work as well as to tap into distant computing resources to run their simulations more quickly.

The first attempt at this took place at the Showcase exhibit, at the SIGGRAPH'92 conference in Chicago. We connected an Inferior Olivary neuron, running on a Sparc station at the University of Illinois at Chicago, to a cerebellar Purkinje cell, running on the Delta at Caltech, via a T3 network line. We were able to interactively inject a small current into the soma of the olivary neuron which caused it to undergo an action potential that in turn evoked a complex spike in the Purkinje cell.

The networking model is designed so that all data gathered during the simulation is sent to a central workstation on the SIGGRAPH exhibit floor. From a control panel, the user could inject small currents into the olivary neuron. An INET (short for internet) object was developed for GENESIS which allowed the transfer of information between the neurons and GENESIS Listener shown in Figure 2 (a visualization tool that animated a three dimensional model of the two neurons). To minimize the amount of data transfered over the T3 line, the INET object used a user-defined color table to encode the floating point membrane voltage values as color information. The color data was then sent to the listener and mapped onto the compartments of the three dimensional model. When the olivary neuron underwent an action potential, a single boolean signal was sent to the listener which relayed this to the Purkinje cell to trigger the complex spike event.

Simulating the olivary neuron on a Sparc station took approximately 0.03 seconds for each iteration step. We applied a number of different mesh sizes to the parallelized Purkinje model and we found that a grid size of about 4 by 4 was optimal for the compartmental assignment scheme we used. This resulted in each iteration step taking approximately 0.062 seconds (compared to 0.36 seconds on a Sparc 2). Despite this high load imbalance between the two neurons, we found that the network delay on the T3 line was the major bottleneck in our prototype. This was because both neurons had to run synchronously and hence no matter how quickly the two machines were able to complete a simulation step, they still had to wait for synchronization before continuing with the next step.

4 INTERACTIVE VISUALIZATION

With the increasing speed and affordability of graphics workstations, many scientific disciplines are fast making computer graphics a standard tool in their research. Visualization offers a holistic understanding of the information that allows scientists to quickly identify areas of interest in their data. Interpreting all the data gathered from neuronal modeling is a formidable task that can afford the assistance of advanced visualization techniques.

The interpretive nature of GENESIS, coupled with the computationally intensive task of calculating thousands of differential equations in the compartmental models (29000 in the Purkinje model), makes rapid visualization of the data difficult. As a result we developed tools that allowed GENESIS data to be visualized at the accelerated speeds offered by today's high performance graphics engines.

4.1 GENESIS Listener

As described briefly in the previous section, the visualization of the collected data was performed on a Silicon Graphics workstation using GENESIS Listener. The listener buffered all incoming color values while allowing the user to perform instant replays of the data collected so far. The color values were mapped onto the corresponding compartments in the two neurons and rendered as a three dimensional model that can be arbitrarily rotated and translated by the user.

4.2 GENESIS Visualizer

Another means of visualizing the collected data is with GENESIS Visualizer. The Visualizer is a tool that allows the user to load multiple neuronal morphologies and data, and key frame animate them. The Visualizer also provides selection of alternative color mapping schemes as well as design of graphs that plot data at selected compartments. This tool allows GENESIS users to rapidly animate their collected data as well as record their animations on film or video for documentation purposes. Whereas neurons in GENESIS are rendered as lines under XODUS' limited graphical interface, GENESIS Visualizer offers gouraud shaded models of the neurons which give it an added degree of realism.

4.3 CAVE Visualization

The interaction between the Purkinje cell and the Olivary neuron was further demonstrated in the CAVE at SIGGRAPH 92, as well as Supercomputing'92 and RSNA'92 (Radiological Society of North America). The CAVE is a non-intrusive virtual reality environment developed by the Electronic Visualization Laboratory at the University of Illinois at Chicago (Cruz-Neira et al. 1992). The CAVE consists of four, ten foot projection screens (left, right, front and floor screens) on which alternating stereoscopic pairs are displayed (see Figure 3). Stereo is mediated by LCD shutter glasses and tracking is via Ascension technology's, Flock of Birds. The CAVE is driven by five Silicon Graphics Crimsons, one for each screen and one to coordinate the other four machines. The CAVE offers a richer experience than other existing virtual reality environments, such as the head mounted display and the NASA Boom, in that its panoramic view gives the user a greater sense of immersion.

Our first implementation of the interaction between the two neurons involved a stripped down version of GENESIS Listener where data computing under GENESIS was sent over an Ethernet connection to the CAVE for viewing. Using a wand (a device used to interact with applications in the CAVE), the user was able to inject a current into the soma of the olivary neuron by touching it. In the future we intend to enhance this by allowing greater degrees of interaction between the user and the models as well as the visualization of large neural networks. We believe that the kind of immersive experience offered by the CAVE will be invaluable when we begin to visualize highly complex neural networks.

5 CLOSING REMARKS

We successfully developed a prototype for the networked interaction and visualization of multiple neurons on heterogeneous computing systems. This is an important first step towards building complex, distributed neural networks that take advantage of all available computing resources at possibly geographically distant locations.

We are still continuing to develop efficient means of interconnecting heterogeneous computer systems to model neural networks (currently a port of GENESIS to the CM-5 is in progress), and we are looking into new ways of visualizing this data in the CAVE. Finally we are designing a database for GENESIS that will allow computational neural scientists to share their models with other researchers.

REFERENCES

Bower, J. M. 1992. "Modeling the Nervous System." *Trends in Neuroscience 15*, no. 11[173] (Nov.): 411-412.

Bower, J. M. and C. Koch. 1992. "Experimentalists and modelers: can we all just get along?" *Trends in Neuroscience 15*, no. 11[173] (Nov.): 458-461.

Concurrent Supercomputing Consortium. 1991. *Intel Touchstone Delta User Manual.* Caltech Concurrent Supercomputing Facilities, Pasadena, California.

Cruz-Neira, C.; D. J. Sandin; T. A. DeFanti; R. V. Kenyon; and J. C. Hart. 1992. "The Cave Audio Visual Experience Automatic Virtual Environment", *Communications of the ACM 35*, no. 6 (June): 64-72.

De Schutter, E. 1989. "Computer Software for Development and Simulation of Compartmental Models of Neurons." *Comput. Biol. Med. 19*, no. 2: 71-81.

De Schutter, E. 1992. "A Consumer Guide to Neuronal Modeling Software." *Trends in Neuroscience 15*, no. 11[173] (Nov.): 462-465.

De Schutter, E. and J. M. Bower. 1993. "Integration of synaptic inputs in a model of the Purkinje cell." In *Computation and Neural Systems 1992*, J. M. Bower and F.H. Eeckman, eds. Kluwer Academic, Norwell, Massachusetts. (in press).

Fotouhi, F.; J. Leigh; and S. P. Rana. 1991. "Optimal Parallel Scheduling of M-way Join Queries." *Information Systems 16*, no.6, (Jan.): 627-639.

Fox, G. C. 1987. "A Graphical Approach to Load Balancing and Sparse Matrix Vector Multiplication on the Hypercube." In *Numerical Algorithms for Modern Parallel Computer Architectures*, Springer Verlag, New York.

Fox, G. C. and W. Furmanski. 1988. "Load Balancing Loosely Asynchronous Problems with a Neural Network." In *Proceedings of the Third Conference on Hypercube Concurrent Computers and Applications.* ACM, New York, 241-278.

Hille, B. 1984. *Ionic Channels of Excitable Membranes.* Sinauer, Sunderland, Massachusetts.

Hodgkin, A. L. and A. F. Huxley. 1952. "A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in a Nerve." *J. Physiol. 117*: 500-544.

Hopfield, J. J. and D. W. Tank. 1985. "Neural Computation of Decisions in Optimization Problems." *Biological Cybernetics 52*, 141-152.

Ito, M. 1984. *The Cerebellum and Neural Control.* Raven Press, New York.

Kirkpatrick, S.; C. D. Gelatt; and M. P. Vecchi. 1983. "Optimization by simulated annealing." *Science 220*, 671-680.

Nelson, M. E.; W. Fumanski; and J. M. Bower. 1989. "Simulating Neurons and Networks on Parallel Computers." In *Methods in Neuronal Modeling From Synapses to Networks*, C. Koch and I. Segev, eds. MIT Press, Cambridge, Massachusetts, 397-438.

Nelson, M. 1990. "Brain Maps and Parallel Computers." *Trends in Neuroscience 13*, no. 10.

Rall, W. 1957. "Membrane Time Constant of Motoneurons." *Science 126*: 454.

Rall, W. 1959. "Branching Dendritic Trees and Motoneuron Membrane Resistivity." *Exp. Neurol.*, no. 1, 491-527.

Rall, W. 1960. "Membrane Potential Transients and Membrane Time Constant Of Motoneurons." *Expt. Neurol. 2*: 503-532.

Rall, W. 1962. "Theory of Physiological Properties of Dendrites." *Ann. N.Y. Acad. Sci. 96*: 1071-1092.

Rall, W. 1989. "Cable Theory for Dendritic Neurons." In *Methods in Neuronal Modeling From Synapses to Networks*, C. Koch and I. Segev, eds. MIT Press, Cambridge, Massachusetts, 9-62.

Segev, I.; J. W. Fleshman; and R. E. Burke. 1989. "Compartmental Models of Complex Neurons." In *Methods in Neuronal Modeling From Synapses to Networks*, C. Koch and I. Segev, eds. MIT Press, Cambridge, Massachusetts, 63-96.

Shepherd, G. M. 1976. *The Synaptic Organization of the Brain.* Oxford University Press, New York.

Smarr, L. and C. E. Catlett. 1992. "Metacomputing." *Commu-*

nications of the ACM 35, no. 6 (June): 44-52.

Yamada W. M.; C. Koch; P. R. Adams. 1989. "Multiple Channels and Calcium Dynamics." In *Methods in Neuronal Modeling From Synapses to Networks*, C. Koch and I. Segev, eds. MIT Press, Cambridge, Massachusetts, 97-133.