Synchronized views for exploring populations of neurons

Kay A. Robbins[1,2] Igor Grinshpan[1] Kevin Allen[1] David M. Senseman[1,2]

[1]University of Texas at San Antonio, [2]Cajal Neuroscience Research Center

Abstract

Davis (Data Viewing System) is a general-purpose data viewer designed for the simultaneous display of a large number of dynamic data sets. Davis was inspired by the need to explore computational models of the cerebral cortex. These systems are distinguished by complex dynamic elements interconnected in irregular patterns. Neuroscientists study the detailed behavior of individual elements and how these elements interact to achieve cortical function. This paper describes Davis and its use in cortical visualization.

Davis is written in Java and can be run from a browser or as a standalone application. Users must provide an XML description of their data, which Davis uses for its menus, browsing and visualization. Davis visualizations can be applied to any collection of space-time data sets, and Davis allows visualizations to be added easily.

Keywords: Cortical visualization, KL decomposition, multi-view visualization, XML input specification

Introduction

The cerebral cortex is composed of multiple large-scale neuronal populations that are interconnected in complex, irregular and dynamically shifting patterns. To understand the computations underlying cognition, neuroscientists must explore both the detailed behavior of individual neurons and the ensemble behavior of multiple neurons during sensory perception. The need to seamlessly combine the "microscopic" and the "macroscopic" behavior of neurons in studying how the cortex processes visual information motivated the development of Davis.

The neuron is the fundamental building block of the cortex. Cortical neurons can be classified as either excitatory or inhibitory depending on whether they increase or decrease the activity of neurons to which they are synaptically connected. These neurons can be of several distinct types that differ in their anatomical shape (morphology), location within the cortex, and their physiological behavior. From a modeling perspective, there is a tradeoff between accuracy and computational efficiency. How much complexity is needed to reproduce essential function? One of the simplest neuronal models, integrate-and-fire, consists only of a cell body that integrates its inputs and generates an action potential if the result is greater than a threshold. Dendrites and axons are sometimes modeled analytically by transmission lines or by a series of interconnected compartments (often cylinders), each of which is assumed to consist of tissue at the same potential. Compartmental models for an individual neuron can range from a few compartments to thousands of compartments (De Schutter and Bower, 1994a,b).

Efforts to understand the function of the cortex lead to many questions that can be approached through visualization. First level questions address basic cortical behavior. What is the structure of the initial response and how does it evolve over time? How are experimental variables related during different phases of the response? For example, how does the soma voltage track calcium or sodium currents? What are the temporal relationships among various channel variables during the response? When does inhibition due to $GABA_A$ become significant? Which excitatory receptors are important in the generation of the initial response wave? What is the role of excitatory receptors that don't appear to contribute to the initial response? What part of the response is due to the input (LGN excitation) as opposed to recurrent excitation?

More integrative questions can be also asked. How does the response change as the stimulus is varied? Can the change be parameterized? Is this change related to some internal cortical mapping? Other classes of questions address the influence on cortical response of structural change in the model or the experiment. What happens to the response, for example, if the NMDA receptors are weakened or removed entirely? What if a chemical that blocks $GABA_A$ is applied to a cortical area? Another class of question relates different indicators of encoding. It is widely held that information is encoded by the timing of spikes generated by individual neurons. However, the relationship between traditional spike coding analysis and population measurements produced by voltage sensitive dye imaging or multi-electrode recording is not understood. Visualizations that compare both types of measurements may further understanding of cortical function.

Data model

Rather than hardwiring the visualizations for particular biological problems, we modeled the problem space more generally. The data model has identifiable groups of elements called coordinate groups. Examples of coordinate groups include neurons of a particular type, calcium channels, grid points for interpolated data, photodiodes, or even virtual points that have been placed to obtain a particular visual effect. Each element in a coordinate group has a spatial position and is associated with one or more time series. The coordinate group forms an organizational unit of elements that are to be treated in the same way by visualization. A single data set can be partitioned into coordinate groups in many different ways. For example, stellate neurons may be placed in a single coordinate group, indicating that each stellate neuron is to be treated as a unit. Alternatively, data from individual compartments of the stellate neurons can be dumped separately. In this case, each type of compartment in the stellate neurons would form a distinct coordinate group with its own set of positions.

We formalized the data model in an XML schema (XML-W3C Homepage), whose top level is shown in Fig. 1. Each problem space or domain is defined by an environment that has coordinate groups, parameters and experiments. Different numerical models or different experimental setups are organized into distinct environments.

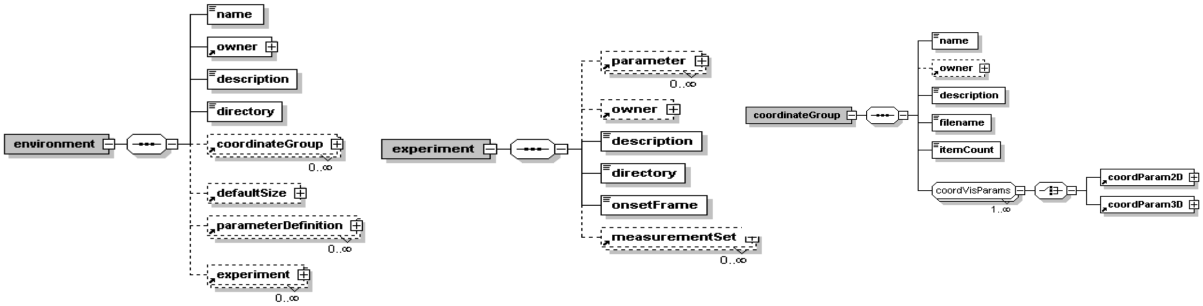


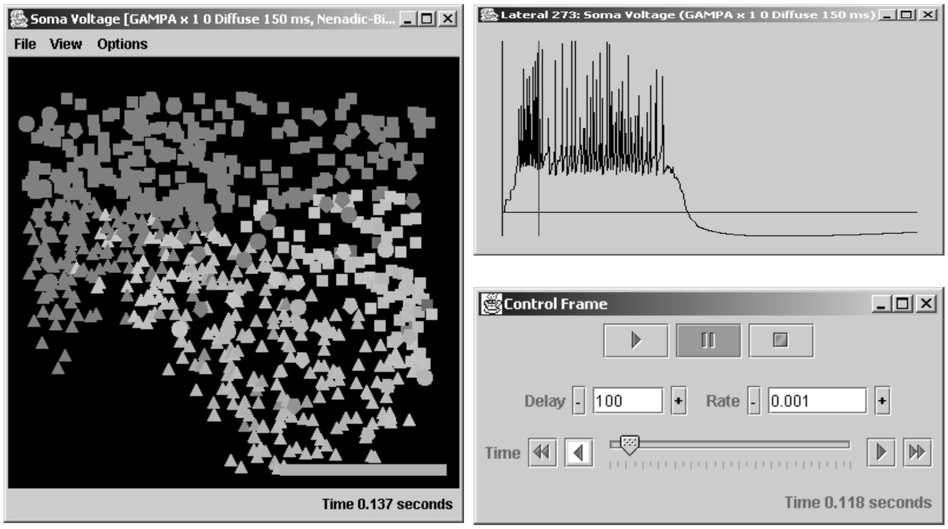
Figure 1. Top-level XML schema for the Davis data model.

The same data can be organized in many ways. The visualization tool is then free to display them differently. The XML schema is quite general. Visualizations based on this schema can be used for many types of data from scientific models or experiments.

Davis visualization

Because of the extensive development of model environments such as Genesis (Bower and Beeman, 1998) and Neuron (Hines and Carnevale, 2001), it was somewhat surprising that even the most basic general tools for viewing populations of neurons were not available. Davis (Data Viewing System) is written in Java and can be used to explore data on the World Wide Web through an ordinary web browser such as Internet Explorer or Netscape. Alternatively, the data viewer can be run locally on a researcher’s machine to examine output from an experiment or from a simulation.

A measurement set represents a physical quantity such as a current or a voltage. Measurement groups contain the values of those sets that are associated with particular coordinate groups. Fig. 2 shows a snapshot of an example visualization display that represents coordinate groups by geometric shapes and the values of the variable associated with the measurement set by a color. The data set used for the visualization was generated by a numerical model of the turtle visual cortex (Nenadic et al., 2000, 2002, 2003) run in Genesis. The measurement set displayed by this visualization is the soma voltage. Five measurement groups are shown, corresponding to the five different types of neurons represented in this model. Each shape in Fig. 2 represents a neuron of a particular type. The color of the shape corresponds to the magnitude of its soma voltage. The particular display of Fig. 2 is set so that the color scale for each type of neuron is determined by the maximum voltage attained by the neurons of that type. The user could also chose to scale the color maps by the global maximum. All of the shape and color attributes are user-settable in this visualization, as well as the color maps used for the different coordinate groups.


Soma Voltage [GAMPA x 1 0 Diffuse 150 ms, Nenadic-Bi...
File
View
Options
Lateral 273: Soma Voltage (GAMPA x 1 0 Diffuse 150 ms)
Control Frame
Time

Time 0.137 seconds

Delay 100

Rate 0.001

Time 0.118 seconds

Figure 2: A sample visualization display.

Since the visualization of Fig. 2 is intended to be used with many simultaneous windows, several optimizations are applied. The display keeps track of the previous color of each element and does not redraw a pixel unless it is in the foreground and the color changes. The visualization also uses a settable quantization to reduce the number of possible colors and therefore also the number of pixels that are redrawn on each frame. These simple optimizations reduce the computational load by a factor of 10 or more for some of the neural data we studied. The display of Fig. 2 uses tool tips to identify individual elements and their current values as the user moves the cursor over the figure. If the user double clicks on a neuron, a plot of the response for an individual neuron appears and a radio line moves across the display to mark the current time.

After an initial prototype of the visualization tool was built, it became clear that a good synchronization mechanism was needed to simultaneously display data sets corresponding to different variables in the same experiment and data from different experiments. Data sets often have different sampling rates, but scientists need to see them on the same time scale. We also wanted the synchronization of the displays and the timing to operate independently of the views. Two rates come into play during display: the nominal rate that data is sampled and the rate at which frames are actually displayed. Suppose the sampling frequency of the original data set is 1000HZ. The interval between successive samples of the data set is then 0.001 seconds. The user may not want to see every frame, but rather to see every other frame. By setting the timer rate to 0.002, the visualization displays every other frame. The user-settable rate does not have to be an integral multiple of the underlying experimental sampling rates of any of the data sets. The viewing rate is not directly related to the sampling rate, and is limited by the time it takes to draw and visualization as well as by the user's ability to see changes. We developed a general-purpose timer class and a VCR control to handle navigation and synchronization. By dragging a slider with the mouse, the user can manually step through the frames. The right and left single arrow buttons allow the user to single step through the frames at the specified sampling rate. The double arrows move the time to the beginning or the end.

Insights from visualization

In this section, we present some examples of how Davis can be used to understand the behavior of the response, both in simulation models and in experimental data. Fig. 3 shows different aspects of the behavior of the NGU model of the turtle visual cortex for a diffuse light flash visual stimulus using the published values of the physiological parameters (Nenadic et al., 2003). The left block of four windows shows the voltages of all the neurons 97 ms after stimulation. The upper left window in this block shows the voltages of the different types of neurons. The visualization distinguishes different neurons by different display shapes. Lateral pyramidal cells, the dominant excitatory neurons in this system are shown with triangles. Medial pyramidal neurons, which show a much smaller response, are displayed with squares. The inhibitory horizontal and stellate neurons are shown using circles and pentagons, respectively. The bar in the lower right actually consists of overlapping LGN neurons that provide the input.

Phase plane plots, based on KL decomposition (Robbins, 1998; Robbins and Senseman, 1998; Kirby, 2001; Senseman and Robbins, 2002), provide a caricature of the overall wave behavior. If we think of each frame as a vector, the data set forms a curve in a high-dimensional space (a space with as many dimensions as there are data elements). We can take the projection of that curve onto the plane formed by finding the two directions in this space that contain the most data set energy using PCA or KL decomposition. The resulting curve gives a caricature of the wave motion. Davis allows the user to select a measurement group on which to perform KL decomposition. The tool computes the two principal directions in the data element vector space and then projects each time step on these vectors to obtain a point in a two-dimensional space. The points thus derived are displayed as small dots. The projection corresponding to the current time is shown as a large dot. The current frame may be interpolated if the timer is not running at a rate commensurate with the sampling frequency of the measurement group.

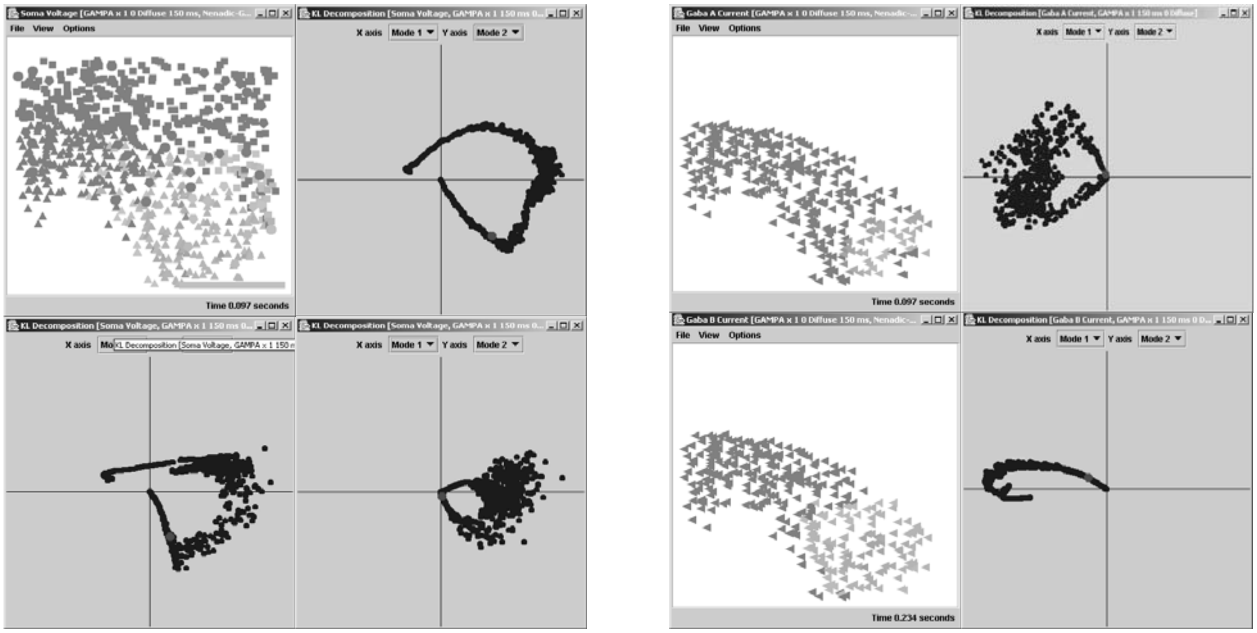

Soma Voltage [GAMPA x 1 0 Diffuse 150 ms, Nenadic-G...
File View Options
Time 0.097 seconds
KL Decomposition [Soma Voltage, GAMPA x 1 150 ms 0...
X axis Mode 1 Y axis Mode 2
KL Decomposition [Soma Voltage, GAMPA x 1 150 ms 0...
X axis Mo KL Decomposition [Soma Voltage, GAMPA x 1 150 m
KL Decomposition [Soma Voltage, GAMPA x 1 150 ms 0...
X axis Mode 1 Y axis Mode 2
Gaba A Current [GAMPA x 1 0 Diffuse 150 ms, Nenadic-...
File View Options
Time 0.097 seconds
KL Decomposition [Gaba A Current, GAMPA x 1 150 ms 0 Diffuse]
X axis Mode 1 Y axis Mode 2
Gaba B Current [GAMPA x 1 0 Diffuse 150 ms, Nenadic-...
File View Options
Time 0.234 seconds
KL Decomposition [Gaba B Current, GAMPA x 1 150 ms 0 D...
X axis Mode 1 Y axis Mode 2

Figure 3: NGU model response to diffuse light flash.

The three other windows of the left block of Fig. 3 show phase plane plots based on KL decompositions of the lateral (upper right), medial (lower left) and horizontal (lower right) neurons. The large dot in each KL window indicates the position of the current frame in the caricature. In each picture the origin corresponds to t=0, since the first frame was subtracted from all of the other frames to move the origin before doing the KL decomposition.

The lateral neuron KL decomposition (upper right window of the left block) has distinctive characteristics. The initial curve is fairly linear, signifying an expanding wave of initial response. The first bend in the curve is reached when inhibition first becomes significant, limiting the propagation. The upper branch of the lateral KL curve represents the post-stimulus decay of the response. The data set spans 1.5 seconds. The KL curve does not return to 0 because the lateral neurons are still hyperpolarized 1.5 seconds into the response. The KL decomposition of the medial pyramidal neurons (lower left window) has a caricature that is similar to the lateral decomposition, but the overall response level is lower and less coherent. The KL decomposition of the horizontal cells (lower right window) shows they have not exerted any significant inhibitory restraint at this time in the response. The stellate inhibitory neurons (whose KL decomposition is not shown here), activate much earlier in the response and have a very smooth caricature, suggesting that these cells play a different, more coherent inhibitory role.

Inhibition plays a crucial role in the structure of the response, and modelers need to know whether the action of the inhibitory neurons matches experimentally measured effects. The NGU model incorporates two types of inhibition, $GABA_A$ and $GABA_B$. The right block of visualization in Fig. 3 demonstrates how visualization can be used to understand the timing of the inhibition in the response. The upper left window displays the amount of the $GABA_A$ inhibition on the lateral neurons at 97 ms after the stimulus. Although the initial response wave has almost reached the left boundary, inhibition has just started to build. This behavior was not expected and suggested to the modelers the need to introduce an additional type of inhibitory neuron that was not included in the initial model. The lower pair of windows in the right block of Fig. 3 shows the $GABA_B$ inhibition of the lateral neurons 234 ms into the response. At 97 ms, $GABA_B$ has no effect. The slower $GABA_B$ inhibition eventually brings the hyperpolarized neurons back to their resting levels.

Davis works on data from animal or other types of experiments as well from model data. Fig. 4 shows the response to a diffuse flash in an animal experiment on the turtle visual system measured using voltage sensitive dye imaging onto a photodiode array (Senseman, 1999). The hexagonal shape of Fig. 4 reflects the arrangement of the photodiode detectors. Each detector measures an average of response in a 300 μm x 300 μm area of cortical tissue. The response shows the characteristic linear growth as the initial response sweeps across the cortex, followed by a turn in the KL caricature as the response reaches the edge of the cortex. The stimulus was administered at 50 ms and the initial response has a longer delay than in the model because of the time it takes to travel the visual pathway to the cortex.

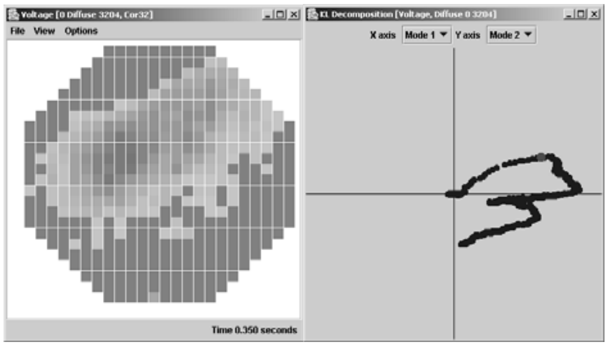

Voltage [0 Diffuse 3204, Cor32]
File
View
Options
Time 0.350 seconds
KL Decomposition [Voltage, Diffuse 0 3204]
X axis
Mode 1
Y axis
Mode 2

Figure 4: Response to diffuse light flash in vitro.

Related work

Related work occurs in several distinct areas. Modeling environments such as Neuron (Hines and Carnevale, 2001) and Genesis (Bower and Beeman, 1998) provide facilities for visualizing responses from individual neurons. Catacomb2 (Cannon et al., 2003) provides a graphical design environment for building neural models. The environment provides basic graphical widgets for plotting curves and panning and zooming, but its emphasis is on model development rather than visualization. The Neosim environment (Neosim Homepage) is a framework based on parallel discrete event simulation techniques which allows modules from different simulation packages and visualizations to be plugged in. This software is still in preliminary development.

XPP (Ermentrout, 2002) is a package for analyzing and animating behavior of small-scale dynamical systems. This package is widely used by modelers working with models on the order of tens rather than hundreds or thousands of neurons. Visu_nbc is another tool designed to visualize neural activity (XNBC Homepage). This tool provides some matrix visualizations for one-dimensional systems. Neural systems have also been visualized using general-purpose visualization environments such as SciRun (Johnson, et al. 2000; SciRun Homepage) or Volvis (Sobierajski et al., 1995). The emphasis in this work has been to reproduce the complex neuronal morphology rather than to display dynamical relationships between variables.

In compact, well-defined problem domains, an encompassing XML specification makes it possible to develop general tools and to compare data produced by different researchers (Holladay, et al., 2002). Several research efforts are underway to design a specification for neural models and data that would allow researchers to exchange data and to develop general-purpose analysis tools. NeuroML (Goddard et al., 2001) is an effort to standardize the specification of neural models by an XML schema. NeuroML also provides a GUI Kit for building Java Widgets and displays. The displays mainly focus on single time series.

The complexity and variation of neural models have been difficult to capture completely in XML schema, and the standard neural modelling environments do not yet support XML-based model input specifications. Cannon et al. (2002) have proposed more flexible, non-curated distributed databases for models and data. NeuroSys (Pittendrigh and Jacobs, 2002) is another neural data management environment based on semi-structured specifications of the data. Each of these systems, as well as ModelDB (Migliore et al., 2003; ModelDB Homepage), has the potential for providing supporting visualization, but none yet provides these capabilities.

Discussion

Davis is conceptually simple and we found it surprising, given its usefulness, that such a viewer was not available already. We began with a simple prototype viewer, which was specific for the turtle numerical model. As the viewer went through refactoring, we were able to generalize it. The data model of experiments, measurement sets and coordinate groups can be used for other scientific applications, and we have examined other types of data such as experimental measurements of flame fronts using this tool.

The KL decomposition gives a simple guide to the overall structure of a particular response and delineates where the response changes its character. These caricatures are useful in narrowing down sections of the data sets where interesting behavior occurs. The current implementation calculates the KL decomposition when the user opens a KL decomposition window. This calculation is somewhat slow. KL spatial basis functions may be pre-computed and read in. Pre-computation is useful not only for efficiency, but also for projecting on other coordinate systems such as those computed from concatenating data sets to form a global coordinate system (Robbins, 1998; Robbins and Senseman, 1998; Senseman and Robbins, 2002).

Davis is effective for looking at data sets that consist of hundreds or perhaps a few thousand elements for a few thousand frames. These data sets can be held in memory and can be effectively examined without sophisticated multiscale viewing techniques. While recognizing that this is a substantial limitation, we note that many published neuronal models fall into this size range.

Acknowledgements

We would like to express our gratitude to our collaborators on the turtle brain project, Philip Ulinski, Bijoy Ghosh and Zoran Nenadic, for sharing their models and insight. This work was partially supported by NIH (G12 RR13646), NSF (ACI-9721348; EIA-0217884) and ONR (N00014-97-0029).

References

[1] J. Bower and D. Beeman. The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System, 2nd ed. Springer-Verlag, 1998.

[2] R. Cannon, M. Hasselmo and R. Koene. From biophysics to behavior: Catacomb2 and the design of biological plausible models for spatial navigation. Neuroinformatics, 1(1), 2003.

[3] R. Cannon, F. Howell, N. Goddard and E. De Schutter. Non-curated distributed databases for experimental data and models in neuroscience. Network: Computation in Neural Systems, 13:415-428, 2002.

[4] B. Ermentrout. Simulating, Analyzing, and Animating Dynamical Systems: a Guide to XPPAUT for Researchers and Students. SIAM Software Environments and Tools 14, 2002.

[5] E. De Shutter and J. Bower. An active membrane model of the cerebellar Purkinje cell I. Simulation of current clamps in slice, J. Neurophysiology, 71:375-400, 1994a.

[6] E. De Schutter and J. Bower. An active membrane model of the cerebellar Purkinje cell II. Simulation of synaptic responses, J. Neurophysiology, 71:401-419, 1994b.

[7] N. Goddard, M. Hucka, F. Howell, H. Corenelis, K. Shankar, and D. Beeman. Towards NeuroML: Model description methods for collaborative modeling in Neuroscience. Philosophical Transactions of the Royal Society, 29:1209-1228, 2001.

[8] M. Hines and N. Carnevale. NEURON: a tool for neuroscientists. The Neuroscientist, 7:123-135, 2001.

[9] K. Holladay, K. Robbins and D. Varner. An XML schema for digital communication signal research. XML '2002, 2002.

[10] C. Johnson, S. Parker and D. Weinstein. Large-scale computation science applications using SCIRun problem solving environment. Supercompter 2000, 2000.

[11] M. Kirby. Geometric Data Analysis. John Wiley & Sons, 2001.

[12] ModelDB Homepage: senselab.med.yale.edu/senselab/ModelDB/default.asp.

[13] NeuroML Homepage: www.neuroml.org/index.html.

[14] Neosim Homepage: www.neosim.org/.

[15] M. Migliore, T. Morse, A. Davison, L. Marenco, G. Sheperd, and M. Hines. ModelDB: Making models publicly accessible to support computational neuroscience. Neuroinformatics, 1:131-134, 2003

[16] Z. Nenadic, B. Ghosh and P. Ulinski. Spatiotemporal dynamics in a model of turtle visual cortex. Neurocomputing, (32-33):479-486, 2000.

[17] Z. Nenadic, B. Ghosh and P. Ulinski. Modeling and estimation problems in the turtle visual cortex. IEEE Trans Biomed. Eng., 49:753-762, 2002.

[18] Z. Nenadic, B. Ghosh, and P. Ulinski. Propagating waves in visual cortex: A large-scale model of turtle visual cortex. J. of Computational Neuroscience, 14(2):161-184, 2003.

[19] NeuroSys Homepage: cns.montana.edu/research/neurosys/index.html.

[20] C. Pittendrigh and G. Jacobs. NeuroSys: An electronic laboratory notebook and semi-structured database. *Neuroinformatics*, in press

[21] K. Robbins. Visualization of scientific video data using KL decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 4(4):330-343, 1998.

[22] K. Robbins and D. Senseman. Visualizing differences in movies of cortical activity. *IEEE Visualization '98*, 217-224, 1998.

[23] D. Senseman. Spatiotemporal structure of depolarization spread in cortical pyramidal cell populations evoked by diffuse retinal light flashes. *Visual Neuroscience*, 16:65-79, 1999.

[24] D. Senseman, and K. Robbins. Modal behavior of cortical neural networks during visual processing. *J Neuroscience*, 19:RC3, 1-7, 1999.

[25] D. Senseman and K. Robbins. High-speed VSD imaging of visually evoked cortical waves: Decomposition into intra- and intercortical wave motions. *J. Neurophysiology*, 87(3):1499-1514, 2002.

[26] SciRun Homepage: software.sci.utah.edu/.

[27] L. Sobierajski, R. Avila, D. O'Malley, S. Wang and A. Kaufman. Visualization of calcium activity in nerve cells. *IEEE Computer Graphics and Applications*, July, 55-61, 1995.

[28] XML-W3C Homepage: www.w3c.org/.

[29] XNBC Homepage: www.b3e.jussieu.fr/xnbc/.