VISUALIZING NERVE CELLS WITH VOLVIS

ARIE KAUFMAN

Center for Visual Computing and Computer Science Department
State University of New York, Stony Brook, NY 11794-4400

VolVis is a high-performance volume visualization software system. It integrates numerous visualization methods within a comprehensive system, providing a flexible tool for the scientist, engineer, and physician, as well as the visualization developer and researcher. Visualizing nerve cells has been a primary application of *VolVis*, and has served as the driving force behind the development of *VolVis*. This paper describes *VolVis*, its unique features, and its various components, as well as several case studies of utilizing *VolVis* in visualizing nerve cell datasets acquired using confocal microscopy. These studies include visualizing surface structures using both iso-surface and volume rendering, with a specific focus on visualizing calcium.

1. Introduction

Neurobiologists have long appreciated the significance of 3D structures when investigating nerve cells. With the advent of confocal laser scanning microscopy, which dramatically increased the resolving power of the microscope, 3D datasets of nerve cells can be routinely acquired. The challenge is now one of eliciting 3D structure from large confocal datasets through the careful application of visualization methods.

The confocal microscope measures light intensity at a specific point within a specimen. This allows the microscope to extract high quality optical sections without the out-of-focus blur that normally accompanies the images produced by light microscopes. In order to provide a significant emitted light response to the scanning laser beam, a fluorescent dye is introduced into the specimen. This provides great scanning flexibility since there are numerous fluorescent dyes, each capable of binding to different cellular compounds. During one experiment the outer surface of a fixed ganglia may be investigated, while in another the distribution of calcium throughout a complex network of neurons may be examined.

Visualizing the diverse array of datasets acquired using confocal microscopy can present a formidable challenge to a visualization system. The system must handle accurate isosurface and volume rendering in addition to providing interactive manipulation and rendering tools. Additionally, the system should provide several visualization methods to aid the scientist while investigating the data. This may include cut planes, animation facilities, filters, stereo projections, texture mapping, and global rendering effects such as shadows and reflections.

In response to these requirements, over the past six years we have developed a comprehensive volume visualization system called *VolVis* [1, 2]. *VolVis* has been disseminated world-wide to over 1,000 active registered users. It is a general-purpose system and is of potential use in any field which acquires 3D sampled data. However, in this paper we focus on *VolVis* applications in neurobiology.

2. The VolVis System

The visualization of volumetric data has aided many scientific disciplines. The diversity of these fields coupled with a growing reliance on visualization has spawned the creation of a number of specialized visualization systems – usually limited by machine and data dependencies and typically not flexible or extensible. A few visualization systems have attempted to overcome these dependencies (e.g., AVS, SGI Explorer, Khoros) by taking a data-flow approach. However, the added computation associated with data-flow systems results in poor performance. In addition, these systems require that the user invests a large amount of time understanding the capabilities of each module and how to effectively link them together.

The *VolVis* system has been designed to meet the following key objectives [2]:

Diversity: *VolVis* supplies a wide range of functionality with numerous methods provided within each functional component. For example, *VolVis* provides various projection methods including ray casting, ray tracing, and marching cubes.

Ease of use: Unlike data-flow systems, the *VolVis* user interface is organized into functional components, providing an easy to use visualization system.

Extensibility: The structure of *VolVis* allows a visualization programmer to easily add new representations and algorithms, using an extensible and hierarchical abstract model [1] which contains definitions for all objects in the system.

Portability: The *VolVis* system, written in C, is highly portable, running on Unix workstations supporting X/Motif or OpenGL. The system has been tested on SGI, Sun, HP, DEC, and IBM workstations and PCs.

Availability: The high cost of visualization systems and difficulties in obtaining their source code often lead researchers to write their own tools for specific visualization tasks. *VolVis* is freely available as a source code for non-profit use.

2.1. System Overview

The *VolVis* pipeline processes input data (volumetric or geometric) to produce visualization output. The input is first processed by the Modeling and Filtering components of *VolVis* (see Sec. 2.2) to generate either a 3D volume model or a 3D geometric surface model of the data. The Measurement component obtains quantitative information from the data. Surface area, volume, histogram and distance information can be extracted from volumes in various ways. Isosurface volume and surface area measurements can be taken either on an entire volume or on a surface-tracked section. Areas and volumes can be computed using either a simple non-interpolated voxel counting or marching cubes [8] based measurement.

Most of the interaction in *VolVis* occurs within the Manipulation component (see Sec. 2.3). The user may modify object parameters such as color, texture, and segmentation, and viewing parameters such as image size and field of view. Within the Navigator of the Manipulation component, the user can interactively modify the position and orientation of volumes, lights, and views. It is closely connected to the

Animator of the Manipulation component, permitting the user to specify animation sequences either interactively or with a set of object transformations.

The Rendering component (see Sec. 2.4) encompasses several rendering algorithms, including geometry-based (e.g., marching cubes) and global illumination methods (e.g., ray tracing). The Input Device component (see Sec. 2.5) maps physical input device data into a device independent representation that is used by various algorithms requiring user interaction. As a result, *VolVis* is input device independent.

2.2. Modeling

A primary responsibility of the Modeling component is the voxelization of geometric objects into volumetric model representations, supporting *volume graphics* [6]. Voxelizing a continuous model into a volume buffer of voxels is a sampling process which determines the voxel values in the volume buffer. To reduce object space aliasing, we adopt volume sampling that estimates density contribution of the geometric objects to the voxels [12]. The density of a voxel is determined by a filter weight function proportional to the distance between the voxel center and the geometric primitive. Since the voxelized geometric objects are represented as volume buffers of density values, we can essentially treat them as sampled volume datasets, and employ one of many volume rendering methods for image generation. One advantage of this approach is that volume rendering carries the smoothness of the volume-sampled objects from object space over into image space. Hence, the silhouette of the objects, reflections, and shadows are smooth. Furthermore, by not performing any geometric ray-object intersections or geometric surface normal calculations, a large amount of rendering time is saved. In addition, constructive solid modeling operations between two volume-sampled geometric models are accomplished at the voxel level during voxelization, using a Boolean operation between pairs of voxels.

2.3. Manipulation

The Manipulation component consists of three sections: Object Control, Navigator, and Animator. The Object Control section is extensive, allowing the user to manipulate parameters of objects, such as modifications to the color, texture, and shading parameters of each volume, as well as more complex operations (e.g., positioning of cut planes and data segmentation). The color and position of light sources can be interactively manipulated. Also, viewing parameters (e.g., final image size) and global parameters (e.g., ambient lighting and background color) can be modified.

The Navigator allows the user to interactively manipulate objects. The user can translate, scale and rotate volumes, light sources, and the view itself, as well as interactively manipulate the view in a manner similar to a flight simulator. To provide interactive navigation speed, a fast rendering algorithm was developed which projects reduced resolution representations of the scene. This task is relatively simple for geometric objects. However, the cost considerably increases for volumetric isosurfaces. A simple, memory efficient method creates a reduced resolution representation of an isosurface by uniformly subdividing the volume into subvolumes and projecting the outer faces of those that contain a portion of the isosurface. These subvolumes serve a dual purpose since they are also used by the PARC (Polygon Assisted Ray Casting) acceleration method [1] during ray casting and ray tracing.

Although the PARC subvolume representation can be stored as a compact list of subvolume indices, the resulting images are boxy and uninformative for many datasets. To overcome this, another method is provided utilizing a reduced resolution marching cubes representation of an isosurface. To reduce the amount of data required for this representation, edge intersections used to compute triangle vertices are restricted to one of four possible locations. This results in much smoother images which are typically more informative than the uniform subdivision method.

Unlike the Navigator which is used to apply a single transformation at a time, the Animator can be used to specify a sequence of transformations to produce an animation. The user can preview the animation using the Navigator, and then select a more accurate and time consuming rendering method, such as volumetric ray tracing, to create a high quality animation. In addition to simple rotation, translation and scaling animations, the Navigator can be used to interactively specify a "flight path", which can then be passed to the Animator and rendered to create an animation.

2.4. Rendering

Rendering is the most important and extensive component of *VolVis*. Speed and accuracy are both critical, yet often conflicting aspects of the rendering process. For this reason, various rendering methods have been implemented in *VolVis*, ranging from the fast, rough approximation of the final image, to the comparatively slow, accurate rendering within a global illumination model. Also, each rendering algorithm itself supports several levels of accuracy, giving the user an even greater amount of control. In this paper we detail only the volumetric ray tracing.

Volumetric ray tracing is built upon a global illumination model. Standard volume rendering methods, also supported by *VolVis*, typically employ only a local illumination model for shading, and therefore produce images without global effects. Including a global effect within a visualization system has several advantages. First, they are often desirable in scientific applications. For example, by placing mirrors in the scene, a single image can show several views of an object in a natural, intuitive manner leading to a better understanding of the 3D nature of the scene. Also, complex surfaces are often easier to render when represented volumetrically than when represented by high-order functions or geometric primitives.

The volumetric ray tracer provided in *VolVis* handles classical geometric objects as well as volumetric data. It is intended to produce very accurate, informative images [10], but strict adherence to the laws of optics is not always desirable. For example, a scientist may wish to view the maximum value along a ray passing through a volume, instead of the optically-correct composited value. Fig. 1 illustrates the importance of including global effects in a maximum-value projection of a

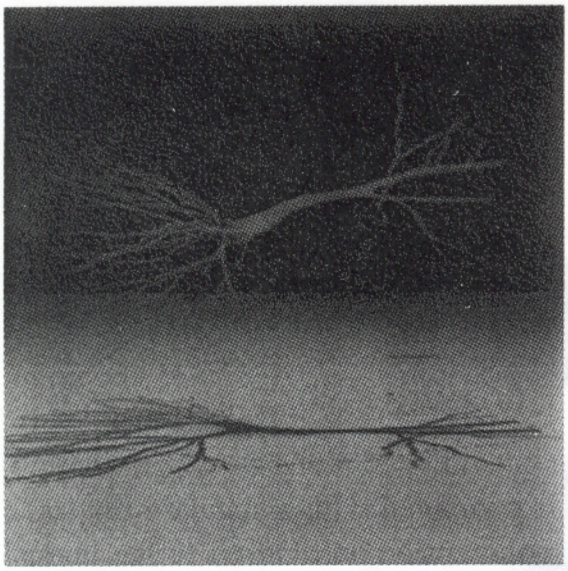

Fig. 1: Ray-traced max-value projection of a hippocampal pyramidal neuron

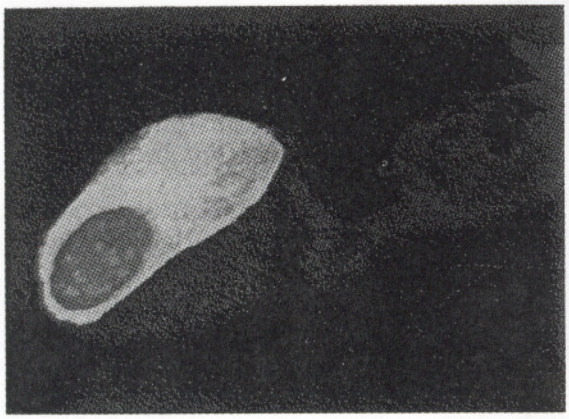

Fig. 2: Cut plane of outer isosurface of bullfrog sympathetic ganglia reveals the nucleus

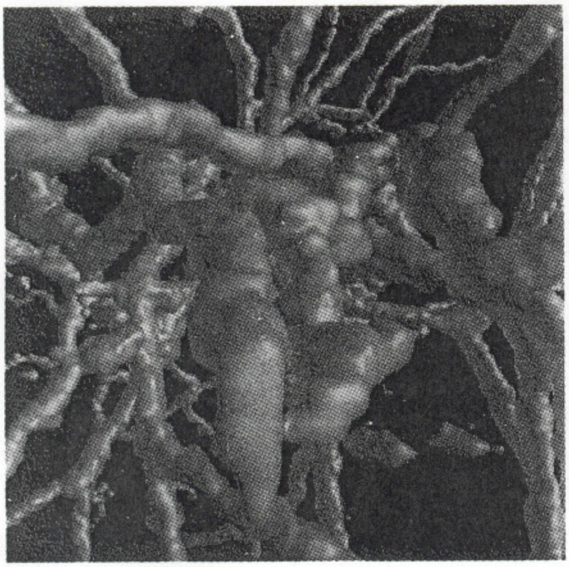

Fig. 3: Ray-traced close-up view of a lateral geniculate nucleus neuron

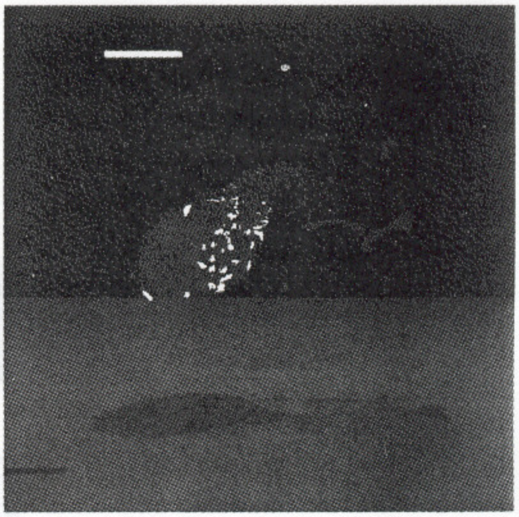

Fig. 4: Volume rendering of bullfrog sympathetic neuron showing subcellular structures

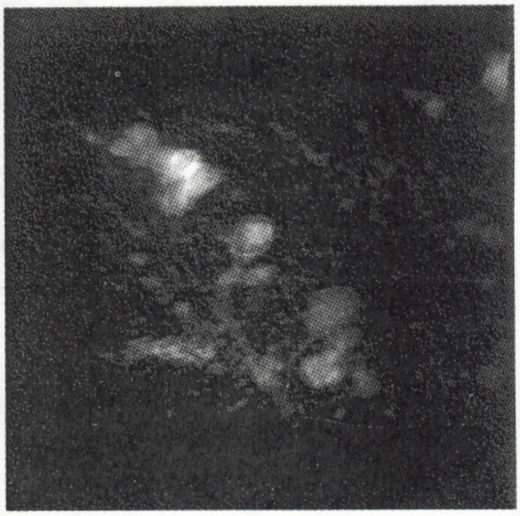

Fig. 5: Volume rendering of zebrafish spinal cord segment

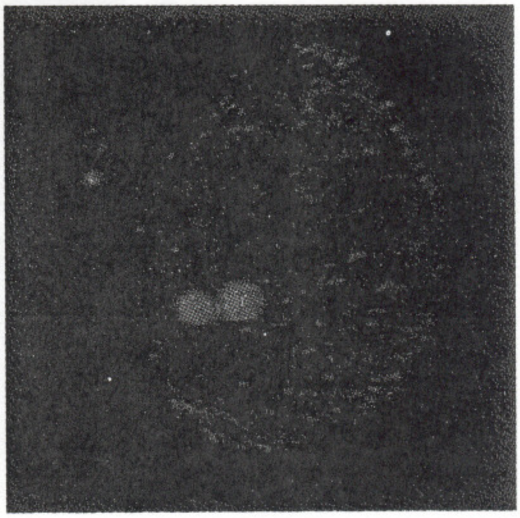

Fig. 6: Volume Rendering of a potassium channel in a dissociated bullfrog ganglion

hippocampal pyramidal neuron dataset which was obtained using a confocal microscope. Since maximum-value projections do not give depth information, a floor is placed below the cell, and a light source above it, producing a shadow of the cell on the floor, adding back the depth information.

In order to incorporate both geometric and volumetric objects into one scene, the classical ray tracing intensity equation – evaluated only at surface locations – has been extended to include volumetric effects. The intensity of light, $I_{\lambda}(x, \vec{\omega})$, for a given wavelength λ, arriving at a position x, from a direction $\vec{\omega}$, is computed by:

$$I_{\lambda}(x, \vec{\omega}) = I_{v\lambda}(x, x') + \tau_{\lambda}(x, x') I_{s\lambda}(x', \vec{\omega}) \tag{1}$$

where x' is the first surface intersection point encountered along ray $\vec{\omega}$ originating at x. $I_{s\lambda}(x', \vec{\omega})$ is the light intensity at x', and is computed with a standard ray tracing illumination equation [13]. $I_{v\lambda}(x, x')$ is the volumetric contribution to the intensity along the ray from x to x', and $\tau_{\lambda}(x, x')$ is the attenuation of $I_{s\lambda}(x', \vec{\omega})$ by any intervening volumes. These values are determined using volume rendering methods, based on a transport theory model of light propagation [7]. It is basically similar to classical ray tracing – rays are cast from the eye into the scene, and surface shading is performed on the closest surface intersection point. The difference is that shading must be performed for all volumetric data that are encountered along the ray while traveling to the closest surface intersection point.

For photo-realistic rendering, the user typically wants to include all shading effects that can be calculated within a given time limit. However, visualization users may find it necessary to view volumetric data with no shading effects, as when using a maximum-value projection. In *VolVis*, the user has control over the illumination equations for both volumetric and geometric objects, and can specify, for each object, which shading effects should be computed. For example, in Fig. 1 no shading effects were included for the maximum-value projection of the cell, while all parts of the illumination equation were considered when shading the geometric polygon.

2.5. Input Devices

The Input Device component of the *VolVis* system allows the user to control various input devices in an input device independent environment. For example, to control the Navigator, the user can utilize various physical input devices such as a keyboard, mouse, Spaceball, and DataGlove. To achieve this, we have developed the *device unified interface* (DUI) [5], a generalized and easily expandable protocol for communication between applications and input devices.

DUI converts raw data received from different input sources into unified format parameters of a "virtual input device." The parameters may include a number of 3D positions and orientations as well as abstract actions – including direct and simple actions (e.g., mouse/Spaceball button clicks) and complex dynamic actions (e.g., two hand gestures or "snap-dragging"). The conversion from real device operations to abstract actions is performed by simulation methods incorporated into DUI.

The primary advantage of employing the virtual input device paradigm is input device independence. In the DUI, each application is interactively assigned a virtual input device, whose configuration is also interactively decided. However, in order to fully utilize the capability of different devices, a virtual input device should not totally hide device dependent information since different devices are suitable for different applications. For example, it is harder to control the Navigator with the Spaceball than with the DataGlove, since the six degrees of freedom provided by the Spaceball are not as entirely independent as they are in the DataGlove [5].

2.6. VolVis Implementations

Two major concerns in implementing *VolVis* have been to ensure that it could be expanded to include new functionality, and that the system would be relatively easy to port to new platforms. Therefore, the development of the *VolVis* system required the creation of a comprehensive, flexible, and extensible abstract model [1]. The model is organized hierarchically; for example, low-level objects such as vectors and points can be combined to create a coordinate system, while at the highest level the World contains the state of every object in the system. The World includes Lights, Volumes, Views, global cut planes, and global shading parameters. Each Volume includes color and texture, local shading parameters, local cut planes, and data which may be either geometric or rectilinear gridded data.

VolVis requires only Unix and X/Motif or OpenGL to run. For portability we implemented all viewing transformations, shading, and hidden surface removal in software. This greatly reduces the rendering speed for geometry-based projection routines used in the Navigator, and in turn reduces the overall interactivity of the system. Since many Unix workstations now include graphics hardware, interactivity can be maintained by utilizing the graphics language of the workstation.

3. Visualizing Surface Structures

When investigating 3D structures of nerve cells a neurobiologist often uses dye which pervades the entire cell. Once the dataset is acquired, it may be rendered as a set of isosurfaces by specifying the intensity values of, for example, the outer isosurface of the cell and the nucleus. Generally, if two isosurfaces are rendered, one is completely contained within the other. It is thus essential for a visualization system to provide cut planes and translucent rendering of isosurfaces to reveal inner structures. Fig. 2 shows the outer surface and nucleus of a bullfrog sympathetic ganglia which was filled with a pervasive dye. To show the internal nucleus a cut plane was applied, and a color table was specified for displaying scalar values on the cut plane. Rendering was performed with the *VolVis* ray caster utilizing PARC [1].

Accurate rendering of isosurfaces (e.g., Fig. 2) is paramount in investigating cell structures. Unfortunately, accurate rendering requires from many seconds to several minutes, depending on factors such as image resolution, data size, and number of isosurfaces. Therefore, a visualization system must provide interactive rendering methods that update the image while the user manipulates parameters. Fig. 3 illustrates an accurate rendering of a lateral geniculate nucleus (LGN) neuron. It shows a close-up of the outer surface of the LGN rendered with the *VolVis* ray tracer [3].

Often a neurobiologist wishes to see a projection of the data without having to interpret the effects of complex compositing or lighting interactions. Such a technique is the *maximum-value projection*, where for each ray cast through the data, only the maximum value found along the ray is stored in the ray pixel. As a result of not performing shading on the data values, this projection technique does not provide any depth information. Incorporating global shading effects such as shadows and reflections into the scene can provide the essential cues necessary for an understanding of the 3D nature of the data. Fig. 1 shows a maximum-value projection of a hippocampal pyramidal cell casting a shadow on a "floor" that was added to the scene. Casting a shadow is a natural, intuitive way to add 3D cues to a scene.

4. Visualizing Calcium

The ability to visualize calcium within nerve cells provides biologists with an effective tool for investigating the function of neural systems. Calcium plays a variety of critical roles in signaling that occurs between and within nerve cells. For example, calcium directly gates synaptic transmission, regulates the excitability of neural networks and also plays a principal role in the formation and storage of memories. In addition, when nerve cells are overstimulated, as occurs during epilepsy or stroke, calcium initiates a neurodegenerative cascade that can lead to neuronal cell death.

We acquire calcium activity data by injecting a fluorescent calcium indicator into the nerve cells. When the cells are active, calcium flows into the cell and the fluorescence increases in proportion to the amount of calcium influx. With the confocal microscope we capture a sequence of regularly-spaced digital images throughout the specimen. In each image, bright pixels correspond to areas of high concentration of either calcium or of the fluorescent indicator itself. We then reconstruct this sequence of 2D images to form a 3D volumetric dataset.

We applied our calcium visualization method in two experiments. First we studied the distribution of calcium influx into individual bullfrog sympathetic neurons to evaluate the levels of calcium in different subcellular compartments. This is important not only for understanding subcellular calcium fluxes, but also for studying in-vivo behavior of the calcium indicators themselves. We applied several visualization techniques to this data including cut planes, volume rendering, and measurements. We then applied *VolVis* to the study of populations of neurons inside larval zebrafish. Because these fish are transparent, we have been able to visualize activity in two networks of neurons in the brain and spinal cord that initiate an escape behavior.

4.1. Bullfrog Ganglia

In the first set of experiments, we examined the dynamics of calcium influx into individual bullfrog sympathetic neurons. Their relatively large size enables us to study calcium changes in the nucleus and cytosol independently. This allows us to conclude that the greater nuclear fluorescence is due to greater amounts of fluorescent indicator in the nucleus rather than a higher level of calcium [9]. This result differs from most previous calcium imaging studies and was facilitated by both visualization of the different compartments of the cell and 3D reconstruction of the minimum diffusion path that could contribute to the observed calcium dynamics [11].

One drawback of isosurfaces is that they do not show the internal distribution of the indicator in the cell. Since this internal structure is often what we want to examine, we need to use a method that shows internal 3D structures in a 2D image. One often-used method is to apply a cut plane to the dataset (see Fig. 2).

Although we use cut planes to reveal multiple surfaces and internal values on the these planes, the foregoing would only render a small subset of the scalar values in the data. To view the entire 3D internal structure, we can employ a composited volume rendering. Fig. 4 shows a transparent rendering of a bullfrog sympathetic neuron. Now we can see the nucleus as well as several other large internal organelles.

To analyze our data, we used the *VolVis* measurement tools to measure distance by interactively selecting two points. These points could exist within the same dataset or belong to two separate datasets. We find this useful when multiple datasets represent neighboring sections of the specimen, which we then align within *VolVis*.

Three-dimensional distance measurements are important when interpreting the time course of calcium dynamics observed in either 1D or 2D imaging experiments. For example, when comparing the time course of the calcium increase in the nucleus and an "equivalent" region of cytoplasm, we need to measure the minimum diffusion path to different regions of the cytoplasm in order to select a region that is truly equivalent to the nucleus in terms of its distance to the plasma membrane.

Volume measurements from 3D datasets can be applied to a number of physiological problems. To illustrate, we used volume measurements of the whole cell and the nucleus to aid us in understanding the nature of the observed calcium signals. By combining an electrophysiological measurement of the total amount of calcium entering the cell with the measured volume of the cell, we can predict how the concentration of calcium will increase. Deviations away from the predicted value are indicative of cellular processes such as extrusion and calcium buffering.

4.2. Zebrafish

Recently a technique was developed to label neuron populations in the brain and spinal cord of larval zebrafish [4]. The indicator calcium green-dextran had been injected into the muscle to label motoneurons in the spinal cord. These spinal motoneurons have axons that extend into the muscle, which in turn take up the indicator and transport it back to the cell body in the spinal cord. The indicator has a basal level of fluorescence that provides enough morphological detail to individually identify labeled neurons inside these fish. Brainstem neurons are similarly labeled by injecting the indicator into the spinal cord where their axons project. When these cells are stimulated, the fluorescence increases. This allows us to simultaneously image activity in neuron populations and to identify which neurons are involved in different behaviors exhibited by these fish, such as swimming and escaping. This is the only method that allows the direct visualization of activity in populations of identified neurons inside the brain or spinal cord of an intact vertebrate. The use of *VolVis* has assisted the development of this method in several important ways.

Labeled neurons take up different amounts of indicator and so their basal fluorescence varies over a very wide range. Our desire to see the internal structures within a spinal cord necessitates the use of a form of volume rendering. Fig. 5 shows a population of motoneurons in a segment of the spinal cord. We used a transparent rendering to help us visualize all of the labeled neurons in this segment. The three larger neurons are primary motoneurons that are active during fast swimming, while the smaller cells are secondary motoneurons used in slower swimming. We wanted to know which groups of motoneurons and how many were active during an escape behavior. This transparent rendering allows us to clearly identify both dimly and brightly labeled cells, whose activity we have monitored in 1D or 2D images. From this work we demonstrated that all motoneurons, both primary and secondary, are synchronously activated during an escape behavior [4].

After injecting the indicator into the spinal cord near the tail of the fish, we observed labeled neurons along the length of the spinal cord and in the hindbrain and midbrain. For optical reasons, it is impossible to image the entire nervous system in a single z-series, but we find that it is often of interest to view the full array of labeled neurons. Our approach was to collect sequential stacks of images along the length of the spinal cord up to the brainstem and to render the nervous system using alignment tools and volume rendering within *VolVis*. Alignment of the datasets was performed by choosing landmarks in the overlapping datasets and transforming the data to the correct location. At times it was helpful to render the datasets as isosurfaces in order to align them more carefully. Once aligned, cut planes were introduced to avoid contributing double the signal to the overlapping areas. Because only those cells whose axons project through the injection site are labeled, this method gives us an immediate view of a population of neurons projecting to a particular segment of the spinal cord. The zebrafish spinal cord has a relatively small number of distinct cell types and these can be individually identified in reconstructions. This composite projection allows us to directly view the distribution of different neuronal types, both in the spinal cord and the brain-stem, that project to a particular segment of the spinal cord, permitting us to examine the geometric relationships.

5. Conclusions

The increasing capabilities of confocal microscopy have placed challenging requirements on visualization systems. The key requirement has been to provide a wide range of methods within an organized system. We have developed *VolVis* to meet these requirements, as well as future challenges that will arise as confocal microscopy improves. One of the more exciting research areas is extending the definition of a dataset to include a fourth (temporal) dimension, leading to the visualization of changes over time in the cell structure or the location of compounds in a cell.

The use of *VolVis* provided substantial assistance in the study of cellular calcium dynamics and in the monitoring of neural network activity. It yielded geometric details that facilitated our understanding of the behavior of calcium indicators inside nerve cells and led to a new view of the calcium permeability of the nuclear envelope. It has also produced anatomical details that significantly facilitated the development of a method that now allows us to directly visualize activity in a population of neurons inside of a behaving vertebrate.

Future advances in imaging technology and activity indicators will enable us to image other ions, such as chloride, sodium, or potassium, that play functional roles distinct from calcium's. A recent study is depicted in Fig. 6, showing the localization of a voltage-gated potassium channel (KV2.1) in a dissociated bullfrog sympathetic ganglion neuron. KV2.1 is a delayed rectifier channel that contributes to the repolarization of the action potential. An antibody to the C-terminus of the channel was conjugated to Texas Red and visualized in a fluorescence laser confocal microscope. A set of optical sections was then obtained and analyzed with *VolVis*. Previously it had been assumed that this channel is found throughout the somatic and axonal membranes. Instead, these data show that the channel is arranged in small clusters that are confined to the soma. Both the somatic localization and clustering are probably important for electrical functioning of the cell. Staining inside the cell (primarily on the nucleolus) is probably artefactual.

Optical developments will also allow us to image deeper into less transparent neural structures such as mammalian brains. Ultimately, it should be possible to acquire 3D activity maps of the many different neural networks that make up the nervous system of the zebrafish and other animals. This will require the development of rendering, navigation, animation, and measurement tools for such "4D" data.

Acknowledgements

VolVis development has been supported by NSF grants CCR-9205047 and MIP-9527694, Office of Naval Research, Department of Energy, Howard Hughes Medical Institute, and the Center for Biotechnology. Data for the figures is courtesy of Howard Hughes Medical Institute, Stony Brook, NY. Thanks are due to Don O'Malley, Kingsley Cox, Paul Adams, Barry Burbach, Pat Lu, and Martin

Wessendorf for providing the case studies, datasets and images. For information on how to obtain *VolVis*, visit http://www.cs.sunysb.edu/ ~volvis, or send e-mail to volvis@cs.sunysb.edu. Thanks are due to Lisa Sobierajski and Rick Avila being the primary developers of *VolVis* and its neurobiology applications, and to Taosong He, Lichan Hong, Claudio Silva, Hanspeter Pfister and Pat Tonra for help with *VolVis*.

References

1. Avila, R., Sobierajski, L. and Kaufman, A., "Towards a Comprehensive Volume Visualization System", *Visualization '92 Proc.*, Oct. 1992, 13-20.
2. Avila, R., He, T., Hong, L., Kaufman, A., Pfister, H., Silva, C., Sobierajski, L. and Wang, S., "VolVis: A Diversified Volume Visualization System", *Visualization '94 Proceedings*, Washington, DC, October 1994, 31-38.
3. Avila, R., Sobierajski, L. and Kaufman, A., "Visualizing Nerve Cells", *IEEE Computer Graphics & Applications*, **14**, 5 (September 1994), 11-13.
4. Fetcho, J. and O'Malley, D., "Visualization of Active Neural Circuitry in the Spinal Cord of Intact Zebra Fish", *J. Neurophys.*, **73**, (Jan. 1995), 399-406.
5. He, T. and Kaufman, A., "Virtual Input Devices for 3D Systems", *Visualization '93 Proc.*, San Jose, CA, Oct. 1993, 142-148.
6. Kaufman, A., Cohen, D. and Yagel, R., "Volume Graphics", *IEEE Computer*, **26**, 7 (July 1993), 51-64.
7. Krueger, W., "The Application of Transport Theory to Visualization of 3D Scalar Data Fields", *Computers in Physics*, July/Aug. 1991, 397-406.
8. Lorensen, W. and Cline, H., "Marching Cubes: a High Resolution 3D Surface Construction Algorithm", *Computer Graphics*, **21**, 4 (July 1987), 163-169.
9. O'Malley, D. M., "Calcium Permeability of the Neuronal Nuclear Envelope: Evaluation using Confocal Volumes and Intracellular Perfusion", *J. Neuroscience*, **14**, (October 1994), 5741-5758.
10. Sobierajski, L. and Kaufman, A., "Volumetric Ray Tracing", *Volume Visualization Symposium Proceedings*, Washington, DC, October 1994, 11-18.
11. Sobierajski, L., Avila, R., O'Malley, D., Kaufman, A. and Wang, S., "Visualization of Calcium Activity in Nerve Cells", *IEEE Computer Graphics & Applications*, **15**, 4 (July 1995), 55-61.
12. Wang, S. and Kaufman, A., "Volume Sampled Voxelization of Geometric Primitives", *Visualization '93 Proc.*, San Jose, CA, Oct. 1993, 78-84.
13. Whitted, T., "An Improved Illumination Model for Shaded Display", *Communications of the ACM*, **23**, 6 (June 1980), 343-349.