



ACADEMIC
PRESS

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Methods 30 (2003) 94–105

METHODS

www.elsevier.com/locate/ymeth

Automated reconstruction of three-dimensional neuronal morphology from laser scanning microscopy images

Alfredo Rodriguez,^{a,b} Douglas Ehlenberger,^{a,b} Kevin Kelliher,^{a,b} Michael Einstein,^{a,c,d}
Scott C. Henderson,^{a,e,f} John H. Morrison,^{a,c,d,f} Patrick R. Hof,^{a,c,d,f}
and Susan L. Wearne^{a,b,d,f,*}

^a *Computational Neurobiology and Imaging Center, Mount Sinai School of Medicine, New York, NY 10029-6574, USA*

^b *Department of Biomathematical Sciences, Mount Sinai School of Medicine, New York, NY 10029-6574, USA*

^c *Kastor Neurobiology of Aging Laboratories, Mount Sinai School of Medicine, New York, NY 10029-6574, USA*

^d *Fishberg Research Center for Neurobiology, Mount Sinai School of Medicine, New York, NY 10029-6574, USA*

^e *Brookdale Department of Molecular, Cell and Developmental Biology, Mount Sinai School of Medicine, New York, NY 10029-6574, USA*

^f *Advanced Imaging Program, Mount Sinai School of Medicine, New York, NY 10029-6574, USA*

Accepted 31 December 2002

Abstract

Experimental and theoretical studies demonstrate that both global dendritic branching topology and fine spine geometry are crucial determinants of neuronal function, its plasticity and pathology. Importantly, simulation studies indicate that the *interaction* between local and global morphologic properties is pivotal in determining dendritic information processing and the induction of synapse-specific plasticity. The ability to reconstruct and quantify dendritic processes at high resolution is therefore an essential prerequisite to understanding the structural determinants of neuronal function. Existing methods of digitizing 3D neuronal structure use interactive manual computer tracing from 2D microscopy images. This method is time-consuming, subjective and lacks precision. In particular, fine details of dendritic varicosities, continuous dendritic taper, and spine morphology cannot be captured by these systems. We describe a technique for automated reconstruction of 3D neuronal morphology from multiple stacks of tiled confocal and multiphoton laser scanning microscopy (CLSM and MPLSM) images. The system is capable of representing both global and local structural variations, including gross dendritic branching topology, dendritic varicosities, and fine spine morphology with sufficient resolution for accurate 3D morphometric analyses and realistic biophysical compartment modeling. Our system provides a much needed tool for automated digitization and reconstruction of 3D neuronal morphology that reliably captures detail on spatial scales spanning several orders of magnitude, that avoids the subjective errors that arise during manual tracing with existing digitization systems, and that runs on a standard desktop workstation.

© 2003 Elsevier Science (USA). All rights reserved.

Keywords: Automated; Reconstruction; Morphology; Pyramidal neurons; Dendrites; Spines; Confocal; Multiphoton laser scanning microscopy

1. Introduction

Neuronal morphology plays a central role in shaping the neural firing patterns and biochemical signaling that underlie synaptic plasticity and memory-related cognitive functions. Fine gradations in spine morphology control biochemical and electrical compartmentalization within

the dendritic tree, while variations in dendritic branching complexity modulate neural integration and the efficacy of action potential backpropagation [1–10]. Neocortical pyramidal cells display a wide range of dendritic morphologies and a correspondingly diverse range of intrinsic firing patterns. Recent compartment models of reconstructed cortical neurons have demonstrated that the full spectrum of observed firing patterns can be reproduced in a range of model neurons with common ion distributions, differing only in the extent of arborization of their dendritic geometries [6–8]. These studies suggest a causal

* Corresponding author. Fax: 1-212-860-4630.

E-mail address: susan@camelot.mssm.edu (S.L. Wearne).

relationship for the observed correlations between dendritic geometry and neural firing properties [11–14]. Importantly, both macroscopic branching topology [2,6,8,10] and microscopic surface irregularities including dendritic varicosities [15] and spine density, shape, and distribution [16–19] contribute to the excitable properties of neurons. Obtaining accurate 3D morphological reconstructions of pyramidal neurons is therefore a crucial first step in evaluating the role of structure in normal and pathologic cognitive function.

Many pathologies of cognitive function, including age-related deficits of short-term memory and disorders of learning and memory, are accompanied by subtle changes in the geometry of dendritic arborizations [20,21] and altered spine densities and distributions [22–27]. Determining exactly how these geometric characteristics relate to neuronal function requires biophysical modeling that can reliably represent both global dendritic structure and detailed spine geometry. Although *in principle* currently available compartment modeling software can discretize neuronal morphology at arbitrary resolution, in practice this is limited by the resolution and accuracy of the digitization method used to capture neuronal morphology.

Currently available digitization software for acquiring neuron morphology relies on manual tracing from a computer screen using custom packages such as NeuroZoom [28] and Neurolucida (MicroBrightField,

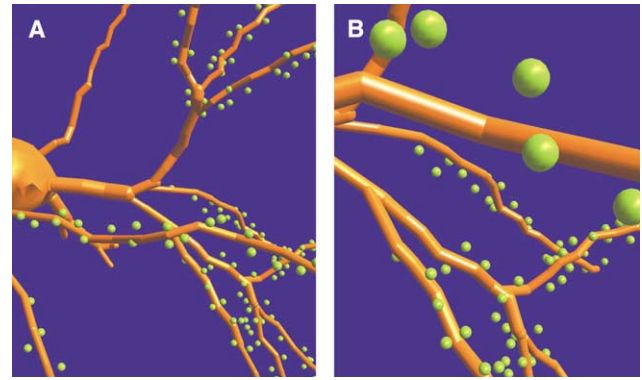


Fig. 1. (A) Layer III pyramidal neuron from macaque monkey prefrontal cortex digitized with standard manual tracing software (NeuroZoom). (B) Zoomed view of (A), showing details of dendritic shape and spines that can be captured by this system. Compare the absence of fine dendritic structure and spine geometry in these manually traced data with the high-resolution reconstructions based on imaged data (Figs. 6; 7A; 8B, inset).

Wiliston, VT, USA) [29]. Such manual tracing methods introduce systematic inaccuracies depending on the individual performing the tracing, and fail to capture fine dendritic structure. In particular, spines are represented as spheres of uniform size; relative diameters of spine heads and necks cannot be resolved. An example of a Layer III pyramidal neuron from macaque monkey digitized with NeuroZoom is shown in Fig. 1. The

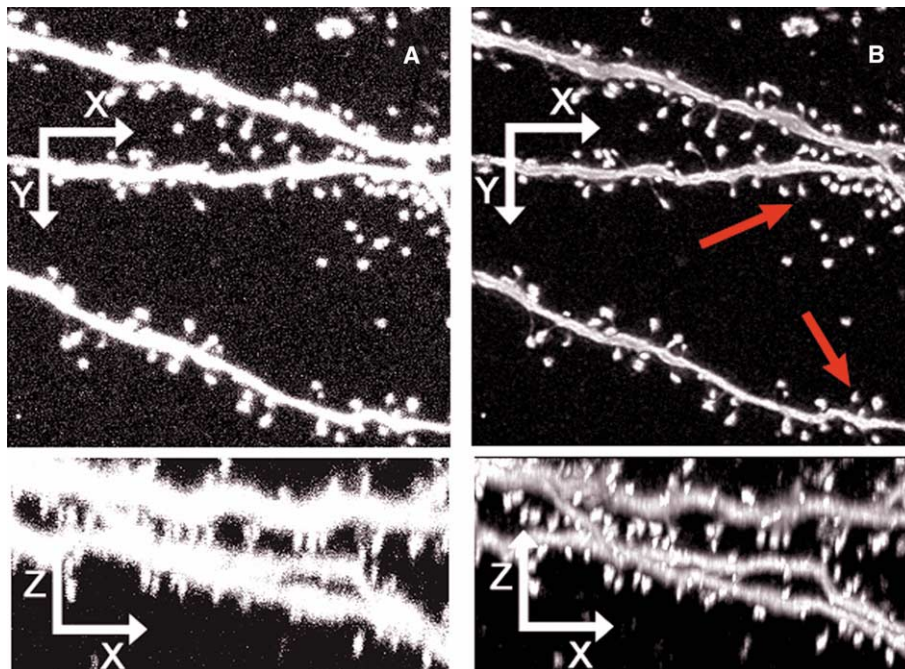


Fig. 3. (A) Raw CLSM data before deconvolution. The two views show maximal intensity projections onto XY (top) and XZ (bottom) planes. Before deconvolution, image spread is greatest in the Z direction, such that 3D spine morphology cannot be precisely ascertained (XZ projection). (B) Data from (A) after blind deconvolution. Removal of image spread is extremely good in all dimensions. Of particular note is the excellent reduction of blurring along the Z axis (XZ plane), permitting morphology of individual spines superimposed on previously oversaturated dendritic trunks to be resolved. Red arrows in the XY plane show loss of spine neck information, which is typical of images from intracellularly injected neurons.

limited resolution of fine dendritic detail and poor representation of spine morphology are apparent from the reconstruction of Fig. 1A. At higher resolution (Fig. 1B), the artifactual articulated structure of the reconstructed dendrites is clearly visible, as is the absence of continuous taper, varicosities, and spine morphology. A first attempt to reconstruct neurons directly from digitized image stacks has been made by Bitplane's NeuronTracer (Bitplane AG, Zurich, Switzerland). The system relies on extensive manual image editing following basic segmentation, which is not feasible for the large datasets that are generated when imaging and reconstructing entire pyramidal neurons at high resolution.

We have developed an automated system to reconstruct neurons in three dimensions from laser scanning microscopy (LSM) images, which is capable of resolving both gross tree topology as well as fine structural detail in dendrites and spines. By reconstructing entire neurons on a microscopic scale these tools will allow model-based simulation and evaluation of the coincident effects of dendritic branching topology and spine geometry and distribution, both of which are critical in regulating neuronal dynamics and synaptic plasticity [2,5,30–34].

In this report, we demonstrate the techniques used to reconstruct and visualize, in three dimensions, Layer III neocortical pyramidal neurons from macaque monkey, imaged entirely with a 100×1.4 NA objective lens. The neurons were visualized by injection of Lucifer Yellow. Image stacks were collected as sets of overlapping tiles by LSM, and reconstructed on a standard laboratory desktop workstation. The elements of the automated image alignment, stitching of adjacent tiles, volume integration, and efficient visualization routines are described in detail, and algorithms for the central elements of each stage are outlined.

2. Materials and methods

The following sections describe techniques used to image the pyramidal neuron shown reconstructed from multiple tiled image stacks using confocal laser scanning microscopy (CLSM) in Fig. 8. Three pyramidal neurons from macaque monkey and one from mouse have been reconstructed using this methodology. The entire XY extent of the neuron in Fig. 8 was imaged with a $100\times$ objective and reconstructed. The combined constraints of loading neurons close to the surface of a tissue block to ensure visibility (see *Cell Loading Techniques*) and the restricted working distance of approximately $100\ \mu\text{m}$ for the high-NA lens of the CLSM system, however, limit the Z extent of these neurons to $100\ \mu\text{m}$ (see *Imaging Techniques*). This creates artifactual truncation of the dendritic tree in Z . Current studies are using deeply loaded neurons at depths of $\sim 200\ \mu\text{m}$ in $400\text{-}\mu\text{m}$ -thick tissue blocks, to capture the full Z extent of the dendritic

tree. Using the longer-working-distance lenses of multiphoton laser scanning microscopy (MPLSM), deep tissue blocks scanned to depths of $200\ \mu\text{m}$ from either surface can be imaged, allowing entire cells to be automatically tiled. These deep-loading and MPLSM imaging techniques are described below. The automated stack alignment, integration, and visualization routines described below are entirely transferable to the resulting deeper image stacks.

2.1. Cell loading techniques

Following protocols conducted according to National Institutes of Health (NIH) guidelines for animal research and approved by the Institutional Animal Care and Use Committee (IACUC) at Mount Sinai School of Medicine, macaque monkeys are deeply anesthetized with ketamine hydrochloride ($25\ \text{mg/kg}$ im) and sodium pentobarbital ($20\text{--}35\ \text{mg/kg}$ iv), intubated, and mechanically ventilated. The descending aorta is clamped at a level just below the heart apex, the right auricle is cut open and the animal is perfused transcardially with cold 1% paraformaldehyde in phosphate buffer (pH 7.4) for 1 min followed by cold 4% paraformaldehyde and 0.125% glutaraldehyde for 12 min at about 250 ml per minute. The brains are removed from the skull, cut into 4-mm-thick coronal blocks or according to experimental requirements, and postfixed no longer than 2 h in 4% paraformaldehyde at 4°C . These blocks are placed in phosphate-buffered saline (PBS) with 0.01% sodium azide at 4°C .

For intracellular injection, 200- to $400\text{-}\mu\text{m}$ -thick slabs are cut on a Vibratome. These thick sections are mounted on nitrocellulose filter paper and immersed in PBS (pH 7.4). The sections are incubated with 4,6-diamidino-2-phenylindole (DAPI; Sigma, St. Louis, MO, USA), a fluorescent nucleic acid stain, for at least 20 min, to reveal the cytoarchitecture under UV excitation. Using DAPI staining as a guide, neurons are impaled with sharp micropipets under visual control, and loaded with 5% Lucifer Yellow (LY; Molecular Probes, Eugene, OR, USA) in distilled H_2O under a dc current of 3–8 nA for 10–12 min, or until the dye has filled distal processes and no further loading is observed [20,35,36]. Generally, up to 15 cells can be injected per slice, with injections sufficiently far apart to avoid overlapping of the dendritic trees. Alternatively, to prevent artifacts due to artificial sectioning of dendrites at either surface of the tissue slab, which occurs in most cases for average-sized pyramidal neurons loaded within the first $100\ \mu\text{m}$ of the surface, cells can be loaded blindly deep in the tissue, aiming with the micromanipulator at depths of about $200\ \mu\text{m}$. This permits a fuller loading of all dendritic branches. After neuronal labeling, sections are fixed again in 4% paraformaldehyde and 0.125% glutaraldehyde in PBS for 4 h at 4°C , washed, and stored in

PBS. Sections are then mounted on uncoated slides and loaded neurons are visualized by confocal laser scanning microscopy. Neurons loaded deeply in the tissue must be imaged using a long-working-distance lens and MPLSM as described below.

2.2. Imaging techniques

2.2.1. Confocal laser scanning microscopy: high-resolution imaging of dendritic spines

Brain slices containing Lucifer Yellow-loaded neurons are mounted in VectaShield (Vector Laboratories, Burlingame, CA, USA) on glass slides beneath No. 1 1/2 coverslips and sealed with clear nail varnish. Prior to overlaying the coverslip, a layer of 0.17 μm red fluorescent beads (Molecular Probes) is applied to the upper surface of the section underneath the coverslip, providing reference points in a separate channel to facilitate manual translation of the stage in the XY plane. Slices are imaged with a TCS-SP (UV) confocal laser scanning microscope (Leica Microsystems, Heidelberg, Germany), equipped with four lasers (Ar–UV, Ar, Kr, HeNe) and four confocal detector channels [with 8-bit photomultiplier tubes (PMTs)]. The spectrophotometer scan head is tuned to optimize the collection of Lucifer Yellow fluorescence while minimizing the detection of background autofluorescence. At the top of each column of sections, a second detector channel is activated and the sample illuminated with the 568-nm line of a Kr laser to collect an image of the red fluorescent beads. Remaining samples are illuminated with the 488-nm line of an Ar laser for LY imaging. Images are collected using a 100 \times 1.4 NA PlanApo oil immersion objective lens and scanned at a resolution of 1024 \times 1024 pixels. At this magnification, without any optical zooming, the images have a field size of 100 \times 100 μm and pixel dimensions of 0.098 \times 0.098 μm . For through-focus imaging, to approximate cubic voxels, optical sections are collected every 0.08 μm using a scan stage with a step resolution of 40 nm and saved to the hard drive. Given the relatively short working distance of the high-NA oil immersion lens (approximately 100 μm), only 1000–1200 sections can be collected per column. Once a column has been imaged through-focus, the stage is refocused to the upper surface of the tissue section and, using the randomly positioned red fluorescent beads as a reference, translated in the XY plane to the adjacent column allowing for a 10–20% overlap to ensure accurate registration when aligning and stitching adjacent stacks. The through-focus imaging is repeated until the entire area occupied by a neuron is imaged. Manual translation of the stage during tiling and accurate registration of adjacent tiles are the most time-consuming features of this system. Tiling can be significantly automated on the current generation of microscopes equipped with motorized XY stages to automate translation and regis-

tration of adjacent tiles in the XY plane, and with larger RAM capacities to permit collection of an entire column 100 μm deep of optical images at a time.

2.2.2. Multiphoton laser scanning microscopy: deep tissue imaging of entire dendritic arbors of pyramidal neurons

Current studies are exploiting the advantages provided by MPLSM to image neurons at greater depths than possible by CLSM. The pulsed infrared (IR) laser penetrates deeper into the tissue, as it is scattered less than shorter-wavelength light. Furthermore, since the illumination occurs only at the focal plane, a pinhole aperture is not required for confocality and external (direct) detectors are used to capture the fluorescence emission (including scattered signal), thus providing greater sensitivity at greater depths than achieved by confocal microscopy. For these studies, we use a Radiance 2000 MP system (Bio-Rad, Hercules, CA, USA), equipped with a Mira 900F Ti:sapphire laser (tuned to 860 nm) and a Verdi 10-W pump laser (Coherent, Santa Clara, CA, USA). This system feeds into an Olympus BX50WIF fixed-stage microscope equipped with LUMPLFL 40 \times 0.8 NA W/IR and LUMPLFL 60 \times 0.9 NA W/IR2 water immersion objective lenses (for deep imaging of the branching patterns) and a UPLAPO 60 \times 1.2 NA W/IR objective lens (for higher-resolution imaging of dendritic spines).

There are a number of considerations and compromises to be made when using MPLSM for these studies. As the illuminating wavelength is greater than that used for confocal microscopy, the lateral resolution is slightly poorer by MPLSM. To exploit fully the advantages provided by MPLSM, objective lenses with the greatest IR transmission efficiency and the largest practical working distance should be used. The dipping lenses used in our MPLSM studies have better IR performance and longer working distances, but poorer resolution due to the lower NA. The higher-NA objective (UPLAPO 60 \times 1.2 NA W/IR2) provides better resolution with good IR transmission but has a shorter working distance (approximately 250 μm). Furthermore, although the illuminating light is less scattered than that for confocal microscopy, permitting greater penetration of the exciting light, the emission is still scattered by the sample, thus limiting the working distance. This can be overcome by inverting the sample and imaging the neuron from the other side. For the long-working-distance objectives, this involves inverting the tissue slice and remounting on a bed of 1% agarose under PBS. For higher-resolution imaging of the dendritic spines, the tissue slice may be sandwiched (in mounting medium) between two coverslips and imaged from both sides.

Fig. 2 shows an example of a pyramidal neuron imaged using MPLSM, which illustrates the feasibility of this approach. The data were collected at low magnification and were volume rendered without deconvolution,

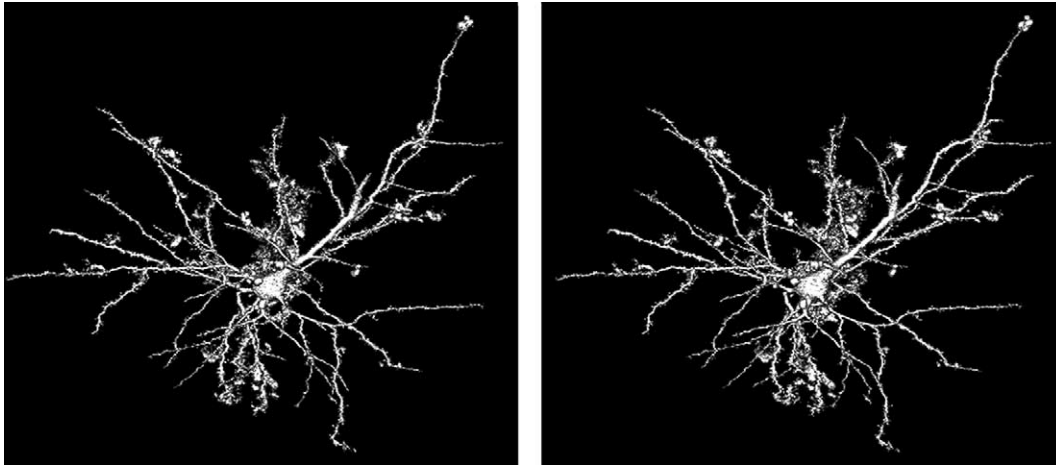


Fig. 2. Stereoscopic image of a pyramidal neuron from a macaque monkey. A Lucifer Yellow-loaded neuron was imaged by multiphoton laser scanning microscopy (using a Bio-Rad Radiance 2000 MPLSM system and a $40\times$ objective lens). Images were collected every $4\ \mu\text{m}$ through a depth of $200\ \mu\text{m}$. The data set was volume-rendered and seed-filled using *VoxelView* (Vital Images, Fairfield, IA, USA) and projected as a stereo pair ($\pm 6^\circ$).

limiting resolution of fine dendritic structure and individual spines, but demonstrating clearly the gross topology of the dendritic tree and absence of dendrite truncation that can be achieved using this system (see figure legend).

2.3. Image analysis

The image processing techniques used in our system are tailored to our combined requirements for deep imaging and high resolution. Image stacks are filtered initially to remove high-frequency noise from the PMT while retaining sharp edges, and deconvolved before volume integration and reconstruction.

2.3.1. Deconvolution

An extended object can be represented as a set of individual point sources of light. If imaging conditions were ideal, the image of each point source would be a point. Diffraction effects due to the finite aperture of the objective and optical aberrations, however, cause a spreading of the 3D image of a point object, which is characterized by the point spread function (PSF) [37,38]. Assuming linearity and shift invariance, we can describe the image of any object by the convolution of the object and the PSF of a single point source of light [39]. In theory, by knowing the PSF of an imaging system comprising both microscope and sample, and without the influence of noise, the convolution operation could be inverted and the object recovered exactly. In practice, noise cannot be entirely eliminated, and experimental measurement of the PSF is difficult. Particularly important for large, deeply imaged samples such as the neurons shown in this report, the PSF is not shift invariant, but is affected by both imaging depth and XY variation in tissue characteristics.

During pilot studies we compared the performance of deconvolution using an experimentally determined PSF, measured from subresolution fluorescent beads inserted in the tissue, with a montaged, blind deconvolution system (AutoQuant Imaging Inc, Watervliet, NY), which uses an estimated and iteratively refined theoretical PSF that can be adjusted locally to account for variations in refractive characteristics of the tissue with depth and sample characteristics. For automated use with large, tiled tissue blocks, the adjustable blind deconvolution was faster, was more accurate, and produced very high resolution results, removing much of the image spread in the Z direction. Fig. 3 shows XY and XZ projections of image stacks before (Fig. 3A) and after (Fig. 3B) blind deconvolution. Prior to deconvolution, image spread in the Z direction precludes accurate recovery of dendritic diameters and 3D spine morphology. After deconvolution (Fig. 3B), dendritic diameters and spine shapes are very well recovered, permitting the morphology of individual spines superimposed on previously oversaturated dendritic trunks to be resolved. Following noise filtering and deconvolution, adjacent image stacks are stitched into a single volume using the Volume Integration and Alignment System (VIAS) developed in our laboratory and described below.

2.4. Volume integration techniques

The objective of VIAS is to create a single volume of dimensions $L \times M \times N$ with a coherent coordinate system $[X, Y, Z]$, $(x_1, \dots, x_L, y_1, \dots, y_M, z_1, \dots, z_N)$ from multiple 2D image stacks, each of variable dimensions $l \times m \times n$, where x and y denote row and column coordinates within an image, and z denotes the image coordinate or slice number. Let dx , dy , and dz denote

the dimensions of a voxel in physical units determined by the imaging system. In the application to confocal or multiphoton images described below, all physical units are in micrometers. Real-time alignment of large volumes is possible by defining a workspace in memory, in which 2D projections of all subvolumes can be manipulated interactively. Alignment and integration proceed as a two-step process: (1) alignment of the workspace, and (2) volume integration.

2.4.1. Workspace alignment

Consider a typical image stack of dimensions 1024×1024 pixels, 1000 images deep. Then $l = m = 1024$, $n = 1000$. Maximal intensity projections map the 3D columns onto 2D planes, which are then used to align adjacent columns. The 3D data are projected along the X , Y , and Z axes, creating maximal intensity projection arrays P_X , an $m \times n$ array in the YZ plane, P_Y , an $l \times n$ array in the XZ plane, and P_Z , an $l \times m$ array in the XY plane. Corresponding to each element (x, y, z) in the 3D volume is an intensity or gray-scale value $g(x, y, z) \in [0, 255]$ for 8-bit images and $g(x, y, z) \in [0, 65\ 535]$ for 16-bit images. For each pixel in the projection arrays $P_X = (p_{jk})$, $P_Y = (p_{ik})$, $P_Z = (p_{ij})$, a sequence of intensity values, $\{g_{x_i}\}$, $i = 1, \dots, l$, $\{g_{y_j}\}$, $j = 1, \dots, m$, and $\{g_{z_k}\}$, $k = 1, \dots, n$, respectively, is generated from consecutive points along the projection axis, in the direction orthogonal to the projection plane. To compute P_Z , the projection along the Z axis onto the XY plane, for example, first a sequence $\{g_{z_k}\}$ of gray-scale values for each pixel in the XY plane is generated. Then the i, j th element in the projection array P_Z is given by

$$P_Z = (p_{ij}) = \max_k \{g_{z_k}\}, \quad k = 1, \dots, n. \quad (1)$$

For any two adjacent stacks, only two of the planar projections are used for alignment.

Alignment is performed interactively on pairs of projections with click and drag tools inside the VIAS graphical user interface (GUI). The user manually positions adjacent projections into approximate alignment with the mouse (Fig. 4A), then uses the autoalign feature to fine-tune the adjustment (Fig. 4B). This procedure is repeated sequentially with all adjacent projections until the entire object is reconstructed.

Graphical user interface. The image stacks to be aligned are opened, and maximal projections are automatically created in the workspace and drawn into the VIAS window. The intrinsic coordinate system and message passing routines of the Windows (Microsoft Corp.) Operating System are used to provide an animated on-screen view of positional adjustments made to projections in the workspace. Windows defines 2D screen coordinates for any window, indexed by the position of the cursor, with origin at the upper left corner of the window and positive directions of X and Y axes increasing rightward and downward from the origin,

respectively. The position of the cursor in screen coordinates is relayed as a continuously updated message stream to the VIAS program. As the user translates projections on the screen, VIAS maps the new screen coordinates to workspace coordinates, adjusting the relative positions of the projections within the workspace. Concurrently, the projections are redrawn at their new locations, providing an animated view of the changing workspace.

Autoalignment. Fig. 4 demonstrates complete 3D alignment of two image substacks, A and B, using planar projections P_Z for alignment in the XY plane and projections P_Y for alignment in the XZ plane. The origin of the workspace is defined by selecting one projection as reference ($P_Z^{(A)}$ in Fig. 4A) and aligning the second projection ($P_Z^{(B)}$ in Fig. 4A) with respect to this reference. All coordinates are assigned relative to the upper left-hand pixel in each projection (red pixels in Fig. 4), called the *projection origin*. The origin of the reference projection is denoted as $(0, 0, 0)$ in workspace coordinates. This reference projection remains fixed at $(0, 0, 0)$ while the second projection is manually positioned into approximate alignment (Fig. 4A). Manual selection of the optimal placement of projection $P_Z^{(B)}$ is tedious and error-prone, so an autoalign feature, initiated from the VIAS toolbox, fine-tunes the adjustments in each plane. The autoalignment algorithm minimizes the scaled ℓ_1 norm of intensity differences computed on a subset of the overlapping pixels within a predefined search window around the initial guess from the manual alignment. To expedite the computation, the autoalignment is performed in two passes: the first as a coarse-grained search on a subset of the overlapping pixels defined by a larger search window, and the second as a fine-grained search, using the full set of pixels defined by a restricted search window centered on the current position of the origin of $P_Z^{(B)}$.

To align two XY projections, $P_Z^{(A)}$ and $P_Z^{(B)}$, for example, $P_Z^{(A)}$ is anchored at the origin of the workspace, and $P_Z^{(B)}$ moved into approximate alignment. Then a coarse-grained search window Ω_C , is defined, with $\pm\alpha$ denoting the number of positions searched in each cardinal direction from the origin of $P_Z^{(B)}$, such that the window has length $2\alpha + 1$ pixels on a side, and is centered on the origin of $P_Z^{(B)}$. This results in $(2\alpha + 1)^2$ candidate positions within Ω_C . The $(2\alpha + 1)^2$ candidate positions generate varying amounts of overlap between the two projections (dark regions in Fig. 4). To minimize processing time, only a subset of overlapping pixels are used to compute the degree of mismatch between the two projections at each candidate position. The overlap region is denoted Ω_0 , with X and Y extents $\hat{l} \leq l$, $\hat{m} \leq m$ and a subsampling step size s is defined. Then the number of pixels sampled per row is $\hat{l}_s = [(\hat{l} - 1)/s] + 1$, where the term in square brackets indicates the integer

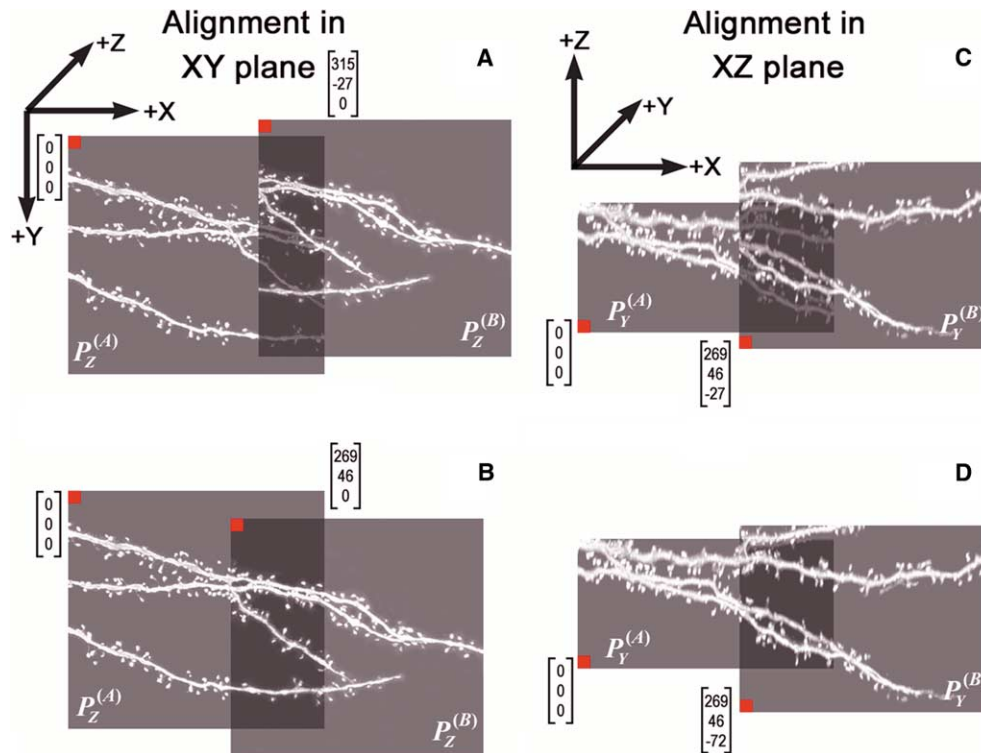


Fig. 4. Alignment of two substacks using projections P_Z (A, B) to constrain the (x, y) coordinates, and P_Y (C, D) to constrain the z coordinate. Red pixel in the upper (A, B) and lower (C, D) corners indicates the origin of each projection. The relative positions (x, y, z) of the projection origins in workspace coordinates are indicated by the vectors adjacent to each point of origin. (A) Manual alignment of adjacent projections onto XY plane, $P_Z^{(A)}$ and $P_Z^{(B)}$. Projection $P_Z^{(A)}$ is the reference; projection $P_Z^{(B)}$ is manually adjusted into approximate alignment. (B) After autoalignment, the overlapping dendritic sections in (A) and (B) are perfectly aligned in the XY plane. (C) To align in the XZ plane, projections $P_Y^{(A)}$ and $P_Y^{(B)}$ need only be aligned in the Z direction. The X and Y coordinates of $P_Y^{(B)}$ are fixed, and the user manually moves the projection into approximate alignment. (D) Autoalignment in the XZ plane determines the optimal Z coordinate, and the 3D alignment is complete. The projections shown are from substacks (dimensions $l^{(A)} = 424$, $m^{(A)} = 394$, $n^{(A)} = 214$, $l^{(B)} = 418$, $m^{(B)} = 392$, $n^{(B)} = 308$) cropped from their original columns.

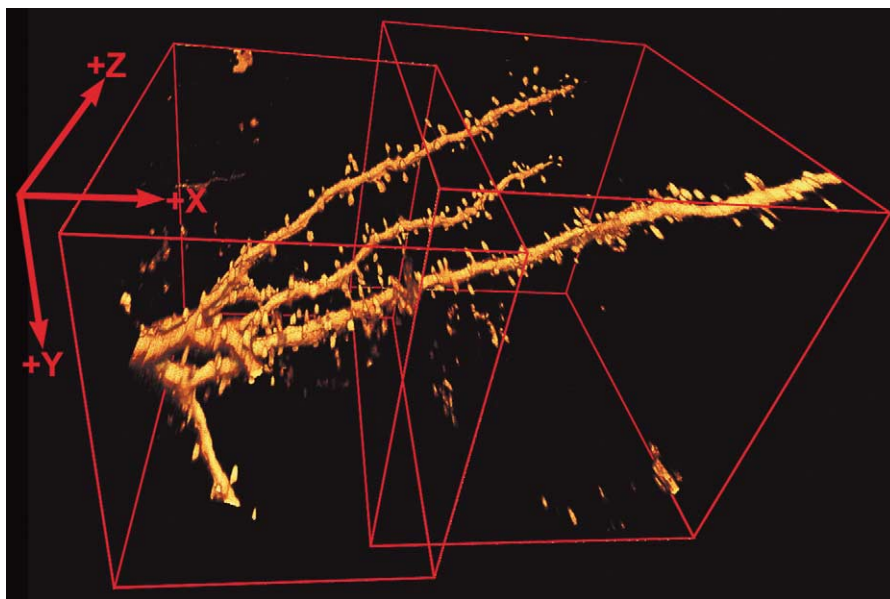


Fig. 6. Volumetric rendering of reconstructed dendritic segments from two substacks following alignment and integration with VIAS system. The integrated volume shows excellent alignment in all three dimensions.

part, and the number sampled per column is $\hat{m}_s = [(\hat{m} - 1)/s] + 1$. Coordinates (\hat{i}, \hat{j}) , $\hat{i} = 1, s + 1, 2s + 1, \dots, \hat{l}_s$; $\hat{j} = 1, s + 1, 2s + 1, \dots, \hat{m}_s$ are assigned to the sampled pixels within Ω_0 . For each candidate position within Ω_C , a penalty function,

$$\delta = \frac{1}{\hat{l}_s \hat{m}_s} \left\| P_{ij}^A - P_{ij}^B \right\|_1 = \frac{1}{\hat{l}_s \hat{m}_s} \sum_{i=1}^{\hat{l}_s} \sum_{j=1}^{\hat{m}_s} |P_{ij}^A - P_{ij}^B|, \quad (2)$$

representing the ℓ_1 norm of overlapping intensity differences, normalized by the number of sampled pixels in the overlap region, is computed. The minimum value of δ determines the optimal alignment. Computation of δ based on a subsampled overlap region with resolution s will ignore any features of scale less than or equal to s , producing a coarse estimate of the optimal alignment. A second fine-grained search extending $\pm s$ pixels in each cardinal direction from this initial estimate, and producing $(2s + 1)^2$ candidate positions, is performed by computing a penalty, δ , using every pixel in Ω_0 . This determines the optimal alignment of $P_Z^{(B)}$ relative to $P_Z^{(A)}$ (Fig. 4B).

Alignment in any one of the three planes, XY , XZ , or YZ determines two of the three translational degrees of freedom required to specify location in a 3D volume. To determine the third degree of freedom, only one of the two remaining orthogonal projections is needed. In Figs. 4A and B, for example, the relative placement of projections $P_Z^{(A)}$ and $P_Z^{(B)}$ in the XY plane completely determines their x and y coordinates, while the z coordinate is determined by aligning projections in the XZ plane, $P_Y^{(A)}$ and $P_Y^{(B)}$ (Figs. 4C, D). The VIAS interface makes use of this constraint to simplify the final alignment: the first two directions (X and Y) are “locked” in place, and translation only along the final (Z) axis is allowed for both manual and autoalign processes. The coordinates of subvolume B are now completely determined by its top left-hand element, located at (269, 46, -72) relative to the reference volume, column A (Fig. 4D). The combined projection ($P_Z^{(A)} + P_Z^{(B)}$) becomes the new reference, and subvolumes adjacent to columns A and B are sequentially aligned and assigned workspace coordinates.

2.4.2. Volume integration

The 3D extent of the volume is computed by taking upper and lower bounds of the aligned workspace along each of the X , Y , and Z axes. The point $(\min(X), \min(Y), \min(Z))$ is designated the origin of the array at $(0, 0, 0)$, and adjoining subvolumes are sequentially inserted into the array using the workspace-to-array coordinate transformation described in pseudocode in Fig. 5. For inserted pixels that overlap in-place pixels, either simple replacement by the inserted value, replacement with the maximum of inserted and in-place values, or replacement by the average of inserted and in-place values can be implemented. This procedure is

```

minX = min(stack[1:n].minX);
maxX = max(stack[1:n].maxX);
minY = min(stack[1:n].minY);
maxY = max(stack[1:n].maxY);
minZ = min(stack[1:n].minZ);
maxZ = max(stack[1:n].maxZ);
dest = alloc((maxX-minX+1)*(maxY-minY+1));
count = alloc((maxX-minX+1)*(maxY-minY+1));

for(z = minZ:maxZ){
  clear(dest);
  clear(count);

  for(i=1:n)
    if(stack[i].minZ <= z && z <= stack[i].maxZ){
      src = read(stack[i],z);
      x = stack[i].minX-minX;
      y = stack[i].minY-minY;
      for(r=y:y+stack[i].height-1)
        for(c=x:x+stack[i].width-1){
          dest[r][c] += src[r][c];
          count[r][c]++;
        }
    }
  average(dest, count);
  write(dest);
}

```

Fig. 5. Pseudocode for volume integration demonstrating the Workspace-to-array coordinate transformation required to integrate aligned substacks.

repeated at sequential Z levels, requiring an array of size only $\max(X) - \min(X) + 1$, $\max(Y) - \min(Y) + 1$ to be retained in memory during the integration process. As each Z level is transformed from workspace to array coordinates, it is written to disk to construct the integrated volume.

Because the alignment and integration are performed on $2D$ projections rather than the volumetric data, the system is extremely efficient and can be run on a standard desktop workstation. The size of the workspace is limited only by available disk and memory space. Each projection is saved to disk, and stored in memory in RGB format for fast screen rendering. To align two typical columns of dimensions $1024 \times 1024 \times 1000$ pixels collected on our system, only 4.74 MB RAM is required to store the three planar projections of each of the two columns. To perform the equivalent alignment using volumetric data (see Fig. 6, for example), the entire 3D datasets of 1 GB each must be held in memory. Storing each dataset in RGB format for fast screen animation during manual alignment would require three copies of the two volumes, or 6 GB RAM for only two columns, clearly prohibitive on standard laboratory workstations. The integrated volume that results from this simple method of alignment from projections is shown in Fig. 6.

VIAS Version 1.0 is available for download at <http://www.mssm.edu/cnic/tools.html>. Full user details are available in the Help Manual that ships with the program, which features illustrated step-by-step instructions for use of the various tools and features. All future

releases or upgrades of the VIAS system will be added to the web site.

2.5. Reconstruction and visualization in three dimensions

Using traditional 3D volume rendering software the integrated volume could in principle be volume rendered and displayed as in Fig. 6. The memory constraints of standard desktop workstations, however, require a significant reduction in the size of the image to enable the entire reconstructed neuron to be viewed and manipulated interactively. To reduce the size of the dataset to be viewed, we extract a polygonal surface from the volumetric data using a variation of the Marching Cubes algorithm [40,41], which defines piecewise linear isosurfaces in a 3D lattice of data points by a set of polygons, typically triangles. Standard Marching Cubes can introduce discontinuities in surfaces exhibiting high curvature, which arise from ambiguities in defining the intersection of the polygonal surface with individual cubes in the 3D lattice. The ambiguous cases originate from Marching Cubes' reduction of the 23 distinct ways of classifying the intersections to only 15 topologically distinct patterns [40]. This reduction introduces ambiguous saddle cases, which can lead to holes in the surface if not properly disambiguated. In theory, additional constraints from adjacent cubes could be used to resolve these cases [41]. In practice, we have found it simpler and more efficient to ignore the case reduction, and use all 23 original distinct patterns [42]. For surfacing highly curved objects such as the spiny dendrites of Figs. 7A and D, this method produces an excellent-quality result at low computational cost.

Real-time inspection of entire neurons in three dimensions is possible using a combination of *object culling* and *level of detail* routines [43]. In the brute-force approach to rendering, every polygon generated by the surfacing algorithm is sent to the rendering system, slowing performance dramatically for large and complex models. Culling techniques optimize performance by rendering only those objects that contribute to the final image for a given frame. We have implemented *view frustum culling* to reduce the number of polygons rendered by discarding entire objects that fall outside the view frustum, or field of view [43]. The view frustum is a truncated pyramid with apex at the eye or camera position from which a scene is viewed, which defines those parts of the scene that are visible to the camera and thus should be rendered. Elements to be rendered can be selected from the scene on the basis of an intersection test between each object and the view frustum. For our application to dendritic morphology, in which the entire volume can be represented as a single, connected surface, the object is subdivided into sections that are each compared with the view frustum using OpenGL's Selection Mode [44] to implement a hardware accelerated intersection test. These intersection tests represent only a small fraction of the rendering time, resulting in significant performance increases for large or complex objects whenever culling is implemented.

Level of detail (LOD) routines store several versions of the object at different precomputed resolutions, using the highest resolution when the viewpoint is close to the object and lower resolutions as it moves further away. A commonly used technique for reducing the LOD is mesh simplification or decimation [45–47], which reduces

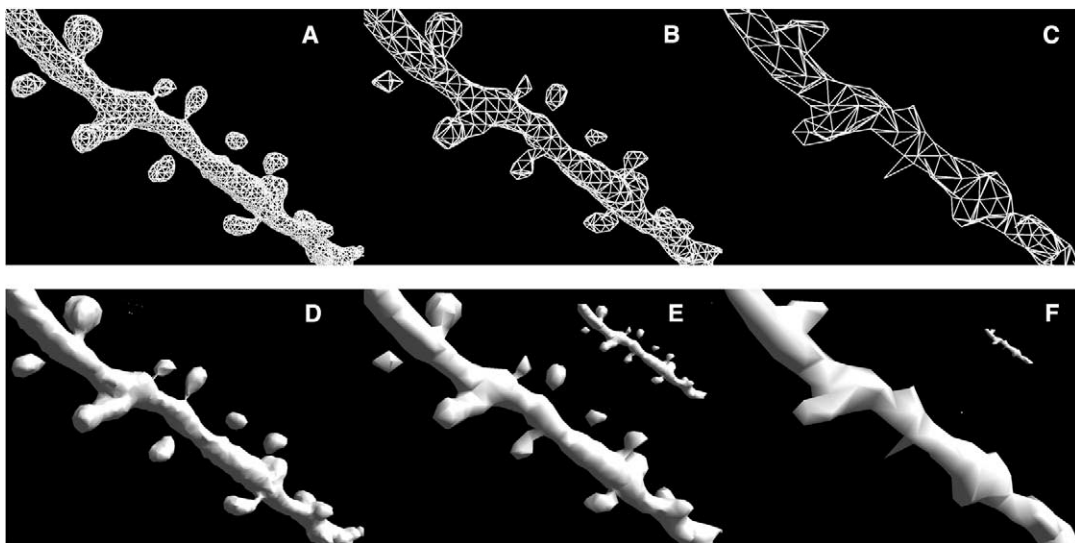


Fig. 7. Spiny dendrite rendered in wireframe (A–C) and surfaced with polygons (D–F) at three successive levels of detail (LOD). (A, D) LOD=0, highest-resolution model. (B, E) LOD=1, an 8-fold reduction in the number of polygons relative to (A). (C, F) LOD=2, a 64-fold reduction in the number of polygons relative to (A). The insets in (E) and (F) show the reduced models at the distance that they would normally be rendered. The reduction in detail, which is obvious in the closeup images, is not noticeable at a distance.

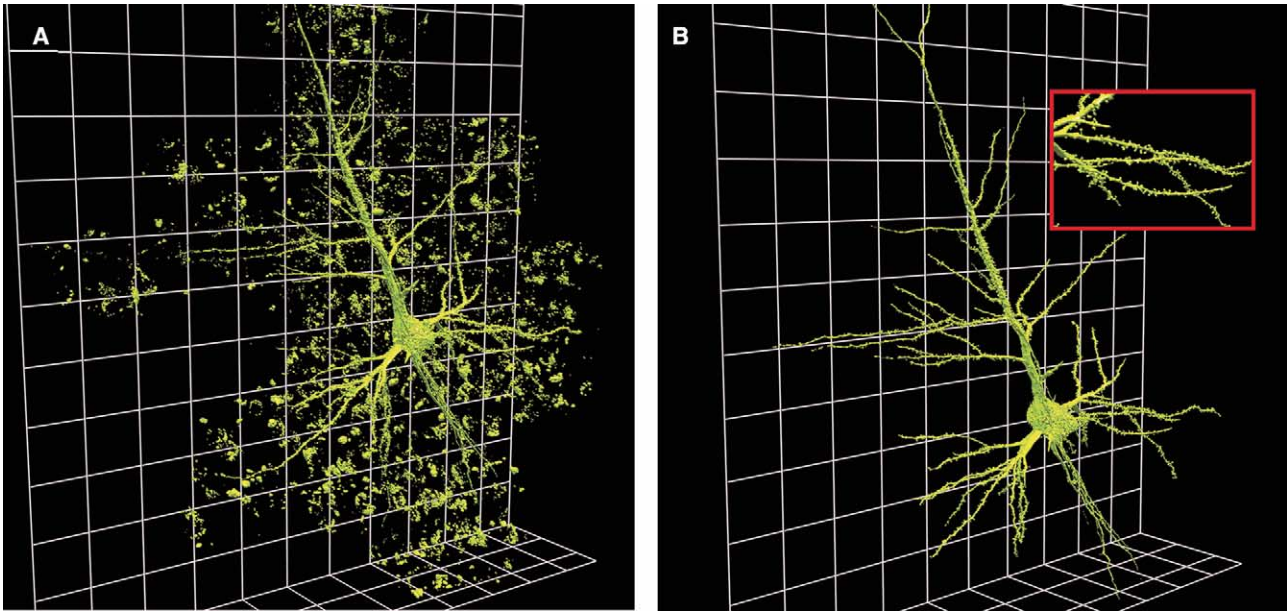


Fig. 8. Fully reconstructed and surfaced neocortical pyramidal neuron from layer III of macaque monkey. (A) Results of surfacing algorithm applied to the entire volume before deconvolution and cell extraction. Dendritic morphology and spine geometry are obscured by image spread and extraneous fluorescent material in the background. (B) After deconvolution and cell extraction, the fine geometry of the dendrites and spine details is clearly visible. A zoomed view of the basal dendrites (inset, red box) shows excellent recovery of the geometry of individual spines.

polygon counts in a model by selectively merging vertices when the surface curvature is relatively low. For spiny dendritic surfaces with inherently high and variable surface curvature, these routines do not produce satisfactory results at low LODs. Instead, we use pixel averaging to subsample the original volumetric dataset before surfacing. Averaging every two adjacent pixels in each of the three spatial dimensions produces a fixed subsampling ratio of $2^3 = 8:1$, effectively halving the linear resolution for each successive LOD. The full-resolution model is defined as level 0, and progressively more coarse resolution models of the object as levels $1, \dots, N$. The subsampled datasets at each LOD are surfaced to produce several models of the object, which are precomputed and read into memory at rendering time. Fig. 7 shows a wireframe rendering of a spiny dendritic segment at three successive levels of detail. When viewed at close range, the reduced LOD in Fig. 7C and F degrades the appearance of the dendrite significantly. At a distance, however (insets, Fig. 7E, F), the degradation in appearance is not noticeable, and higher LODs are unnecessary.

Several methods are commonly used to select an appropriate LOD for an object at any given position relative to the viewpoint. These include analytical expressions based on projected areas [46] and user-defined range-based functions [43]. Our system uses a range-based method in which the user assigns a maximum and minimum viewing distance for each LOD. The Euclidean distance, d , from the camera to the object is first computed, and the appropriate LOD selected for

distances falling within predefined ranges. In Fig. 7 for example, LOD 0 is used whenever d falls between zero and some user-defined value r_1 . LOD 1 (Fig. 7B) is used whenever $r_1 \leq d \leq r_2$, and the coarsest level of detail, LOD 3 (Fig. 7C) is used whenever $r_2 \leq d \leq r_3$. Using these optimization techniques, our current version of PVIEW allows quasi-interactive frame rates in the range 5–15 frames per second (fps). These rates can be improved with further optimization, and as new rendering hardware becomes available. PVIEW is available for download from the CNIC web site at <http://www.mssm.edu/cnic/tools.html>.

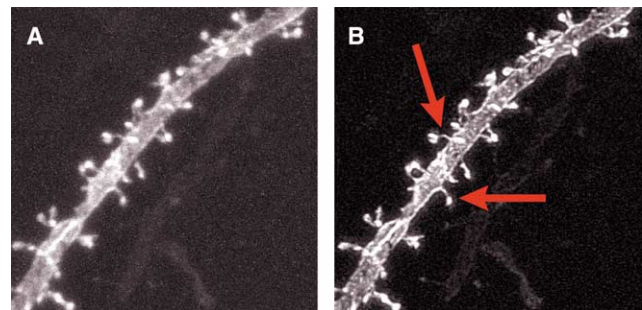


Fig. 9. Spiny dendrite from macaque monkey cortical pyramidal neuron, labeled using DiOlistic technique. (A) Raw data prior to deconvolution. (B) Deconvolved data. Spine morphology and, in particular, spine necks are clearly visible using this technique. Compare recovery of fine spine necks (red arrows in (B), above) with loss of spine neck information from intracellularly labeled neurons (red arrows, Fig. 3B).

2.6. Cell extraction

Extraction of neurons from the background is complicated by the presence of naturally fluorescent structures in the tissue (Fig. 8A) which are difficult to distinguish from spines using automated algorithms. In spiny neurons filled by intracellular injection of fluorescent markers, spine necks are often poorly illuminated and some spines may appear detached from the dendrites (Figs. 7A; 3B, red arrows), so that standard flood-fill algorithms cannot be used directly. We have developed a semiautomated cell extraction algorithm based on a two-pass procedure. Prior to cell extraction, a minimal threshold for the neuronal surface is chosen. Threshold selection is performed on the deconvolved projections by interactively adjusting the threshold using a 2D contouring algorithm (PhotoShop, Adobe Systems) until the contour fits exactly over the projected outline of the spines. In the first pass, the extraction routine marks all connected pixels from an initial seed inside the cell. A threshold that just overestimates the extent of the cell is selected, defining a 3D mask around the cell boundary. Overestimating ensures that spines maintain connectivity to the dendrites. All pixels outside the mask are set to zero intensity, effectively eliminating all unattached background structures. A second pass resurfaces the resulting dataset, at the original, minimal threshold value. This effectively isolates the cell from the background, resulting in a finely detailed dataset that efficiently captures spines while rejecting extraneous noise (Fig. 8B).

Fig. 8 shows a Layer III neocortical pyramidal neuron from the superior temporal cortex of a macaque monkey, reconstructed using the VIAS system from 30 individual subvolumes of ~ 1 GB each, and visualized using PVIEW. The system can reconstruct the entire dendritic arbor at 0.1-mm resolution with the morphology of individual spines clearly discernible. Using these data, realistic biophysical models can be constructed to assess the functional role of variations in fine dendritic structure and spine morphology.

3. Future directions

Further progress in resolving dendritic morphology with LSM is constrained by available technology. Currently we are using deep cell loading and MPLSM to acquire more complete reconstructions of global dendritic arbors (see *Imaging Techniques*). Recovery of entire trees without truncation of branch tips is critical for complete and accurate 3D reconstruction of neuronal geometry. Since pyramidal neurons can occupy volumes up to $400\text{--}600\mu\text{m}^3$, development of deep-cell loading and imaging techniques is an important aspect of this project.

At the local level, alternate methods are being developed for visualizing dendritic morphology that better resolve spine geometry and spine necks, using ballistic impregnation of the cellular membranes by diffusion of lipophilic fluorescent compounds (“DiOlistics” [48]). This technique has been used in live tissues but can be applied to postmortem materials for which it is more reliable than cell loading. Cell loading in human tissue is limited by a number of factors such as fixation and postmortem autolysis time, whereas DiOlistics provide excellent labeling of neurons in paraformaldehyde-fixed materials (4%, 48 h) with postmortem intervals of less than 12 h. Ballistic labeling of tissues using a Helios gene gun and tungsten beads coated with lipophilic fluorescent dyes such as DiI and DiO [48] yields fine and stable labeling of the membranes and reveals all of the structural details of neuronal dendrites and axons such as boutons and all the various types of dendritic spines. Fig. 9 shows a fragment of spiny dendrite visualized using DiOlistics. Using this technique, extremely fine spine necks which may not fill with traditional intracellular injections (red arrows, Fig. 3B) are clearly visible (red arrows, Fig. 9B), allowing accurate quantification of the relative dimensions of spine heads and necks.

4. Conclusion

The approaches described in this article will contribute new information about subtle morphologic alterations in neuropathologic studies of a variety of neurologic, psychiatric, and developmental disorders in which neurons are known to undergo considerable dendritic attrition and spine pathology [26,27]. Such methodologies are also useful in the field of normal brain aging in which dendritic alterations as well as spine loss are known to occur in primates in the absence of neuronal loss [20,21]. Importantly, the software developed for this project does not require supercomputing power and memory to align, integrate, and visualize the large datasets required for high-resolution reconstruction of entire neurons, but can be implemented on a standard desktop workstation. With the growing recognition that fine dendritic structure and spine morphology are crucial determinants of neuronal dynamics and synaptic plasticity, the ability to resolve detail at both local and global scales is increasingly important for studies relating neuronal function to structure. By reconstructing entire neurons automatically, and on a microscopic scale, the tools described in this report provide the basis for an entirely new level of morphometric analysis and realistic biophysical compartment modeling in multiple neuron types and on spatial scales spanning several orders of magnitude.

Acknowledgments

We thank Huiling Duan, Charles Park, William Janssen, Daniil Rolshud, and Tony Flores for assistance with cell loading and imaging. Special thanks are due John Kichury for ongoing advice concerning high-resolution 3D graphics. This work was supported by NIH Grants DC04632, MH58911, MH060734, RR16754 the Howard Hughes Medical Institute, and Mount Sinai School of Medicine. Microscopy was performed at the MSSM-Microscopy Shared Resource Facility, supported, in part, with funding from NIH-NCI shared resources grant (CA095823), NSF Major Reserch Instrumentation grant (DBI-9724504) and with a Howard Hughes Medical Institute Biomedical Research Support Program award to Mount Sinai School of Medicine.

References

- [1] C. Koch, I. Segev, *Nat. Neurosci.* 3 (2000) 1171–1177.
- [2] P. Vetter, A. Roth, M. Häusser, *J. Neurophysiol.* 85 (2001) 926–937.
- [3] T. Euler, W. Denk, *Curr. Opin. Neurobiol.* 11 (2001) 415–422.
- [4] G. Stuart, N. Spruston, M. Häusser (Eds.), *Dendrites*, Oxford Univ. Press, Oxford, 1999.
- [5] B.I. Henry, P.R. Hof, P. Rothnie, S.L. Wearne, in: M.M. Novak (Ed.), *Emergent Nature: Patterns, Growth and Scaling in the Sciences*, World Scientific, Singapore, 2002, pp. 65–75.
- [6] Z.F. Mainen, T.J. Sejnowski, *Nature* 382 (1996) 363–366.
- [7] J. Duijnhouwer, M.W.H. Remme, A. van Ooyen, J. van Pelt, *Neurocomputing* 38–40 (2001) 183–189.
- [8] J.L. Krichmar, S.J. Nasuto, R. Scorcioni, S.D. Washington, G.A. Ascoli, *Brain Res.* 941 (2002) 11–28.
- [9] P.C. Bressloff, S. Coombes, *Int. J. Mod. Phys. B* 11 (1997) 2343–2393.
- [10] P.C. Bressloff, B. De Souza, *Physica D* 115 (1998) 124–144.
- [11] S. Franceschetti, E. Guatteo, F. Panzica, G. Sancini, E. Wanke, G. Avanzini, *Brain Res.* 696 (1995) 127–139.
- [12] B.W. Connors, M.J. Gutnick, *Trends Neurosci.* 13 (1990) 99–104.
- [13] A. Mason, A. Larkman, *J. Neurosci.* 10 (1990) 1415–1428.
- [14] C.R. Yang, J.K. Seamans, N. Gorelova, *J. Neurosci.* 16 (1996) 1904–1921.
- [15] A. Surkis, C.S. Peskin, D. Tranchina, C.S. Leonard, *J. Neurophysiol.* 80 (1998) 2593–2607.
- [16] K. Stratford, A. Mason, A. Larkman, G. Major, J. Jack, in: R. Durbin, C. Miall, G. Mitchison (Eds.), *The Computing Neuron*, 80, Addison–Wesley, London, 1989, pp. 296–321.
- [17] S.M. Baer, J. Rinzel, *J. Neurophysiol.* 65 (1991) 874–890.
- [18] W.R. Holmes, *Brain Res.* 478 (1989) 127–137.
- [19] C.J. Wilson, *J. Electron Microsc. Tech.* 10 (1988) 293–313.
- [20] H. Duan, S.L. Wearne, J.H. Morrison, P.R. Hof, *Neuroscience* 114 (2002) 349.
- [21] H. Duan, S.L. Wearne, J.H. Morrison, P.R. Hof, *Cereb. Cortex*, in press.
- [22] I. Ferrer, F. Gulotta, *Acta Neuropathol. Berlin* 79 (1990) 680–685.
- [23] S.A. Irwin, B. Patel, M. Idupulapati, J.B. Harris, R.A. Crisostomo, B.P. Larsen, F. Kooy, P.J. Willems, P. Cras, P.B. Kozlowski, R.A. Swain, I.J. Weiler, W.T. Greenough, *Am. J. Med. Genet.* 98 (2001) 161–167.
- [24] J.W. Swann, S. Al-Noori, M. Jiang, C.L. Lee, *Hippocampus* 10 (2000) 617–625.
- [25] E.A. Nimchinsky, A.M. Oberlander, K. Svoboda, *J. Neurosci.* 21 (2001) 5139–5146.
- [26] J.C. Fiala, J. Spacek, K.M. Harris, *Brain Res. Rev.* 39 (1) (2002) 29–54.
- [27] E.A. Nimchinsky, B.L. Sabatini, K. Svoboda, *Annu. Rev. Physiol.* 64 (2002) 313–353.
- [28] W.G. Young, E.A. Nimchinsky, P.R. Hof, J.H. Morrison, F.E. Bloom, *NeuroZoom Software User Guide and Reference Books*, YBM Inc., San Diego, 1997.
- [29] J.J. Capowski, in: R.R. Mize (Ed.), *The Microcomputer in Cell and Neurobiology Research*, Elsevier, Amsterdam, 1985, pp. 85–109.
- [30] G. Stuart, J. Schiller, B. Sakmann, *J. Physiol.* 505 (1997) 617–632.
- [31] J.C. Magee, D.A. Johnston, *Science* 275 (1997) 209–213.
- [32] M. Matsuzaki, G.C.R. Ellis-Davies, T. Nemoto, Y. Miyashita, M. Iino, H. Kasai, *Nat. Neurosci.* 4 (2001) 1086–1092.
- [33] G.J. Stuart, M. Häusser, *Nat. Neurosci.* 4 (2001) 63–70.
- [34] R. Yuste, T. Bonhoeffer, *Annu. Rev. Neurosci.* 24 (2001) 1071–1089.
- [35] A.D. de Lima, T. Voigt, J.H. Morrison, *J. Comp. Neurol.* 296 (1990) 159–172.
- [36] E.A. Nimchinsky, P.R. Hof, W.G. Young, J.H. Morrison, *J. Comp. Neurol.* 374 (1996) 136–160.
- [37] D.A. Agard, Y. Hiraoka, P.J. Shaw, J.W. Sedat, *Methods Cell Biol.* 30 (1989) 353–378.
- [38] P.J. Shaw, D.J. Rawlins, *J. Microsc.* 163 (1991) 151–165.
- [39] I.T. Young, *Methods Cell Biol.* 30 (1989) 2–47.
- [40] W.E. Lorensen, H.E. Cline, *Computer Graphics* 21 (1987) 163–169.
- [41] C. Montani, R. Scateni, R.A. Scopigno, *Visual Computer* 10 (1994) 353–355.
- [42] A. Rodriguez, K.T. Kelliher, D. Ehlenberger, S.C. Henderson, M. Einstein, D. Rolshud, H. Duan, J.H. Morrison, S.L. Wearne, P.R. Hof, *Soc. Neurosci. Abstr.* 27 (2001) 162.
- [43] T. Akenine-Möller, E. Haines, *Real-Time Rendering*, second ed., A.K. Peters, Natick, MA, 2002.
- [44] M. Woo, J. Neuder, T. Davis, D. Shreiner, *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2*, third ed., Addison–Wesley, Reading, MA, 1999.
- [45] C. Erikson, *Polygonal Simplification: An Overview*. Technical Report TR96-016, UNC Chapel Hill Computer Science Department, 1996.
- [46] P.S. Heckbert, M. Garland, *Survey of Polygonal Surface Simplification Algorithms*, Technical Report, Carnegie–Mellon University Computer Science Department, 1997.
- [47] M. Garland, P.S. Heckbert, *IEEE Visualization* 98 (1998) 263–269.
- [48] W.-B. Gan, J. Grutzendler, W.T. Wong, R.O. Wong, J.W. Lichtman, *Neuron* 27 (2000) 219–225.