

# VolRoverN: Enhancing Surface and Volumetric Reconstruction for Realistic Dynamical Simulation of Cellular and Subcellular Function

John Edwards · Eric Daniel · Justin Kinney ·  
Tom Bartol · Terrence Sejnowski · Daniel Johnston ·  
Kristen Harris · Chandrajit Bajaj

Published online: 8 October 2013  
© Springer Science+Business Media New York 2013

**Abstract** Establishing meaningful relationships between cellular structure and function requires accurate morphological reconstructions. In particular, there is an unmet need for high quality surface reconstructions to model subcellular and synaptic interactions among neurons and glia at nanometer resolution. We address this need with VolRoverN, a software package that produces accurate, efficient, and automated 3D surface reconstructions from stacked 2D contour tracings. While many techniques and tools have been developed in the past for 3D visualization of cellular structure, the reconstructions from VolRoverN meet specific quality criteria that are important for dynamical simulations. These criteria include manifoldness, watertightness, lack of self- and object-object-intersections, and

geometric accuracy. These enhanced surface reconstructions are readily extensible to any cell type and are used here on spiny dendrites with complex morphology and axons from mature rat hippocampal area CA1. Both spatially realistic surface reconstructions and reduced skeletonizations are produced and formatted by VolRoverN for easy input into analysis software packages for neurophysiological simulations at multiple spatial and temporal scales ranging from ion electro-diffusion to electrical cable models.

**Keywords** Electron microscopy · Serial sections · 3-D reconstruction · Neuropil · Skeletonization · Reduced model · Electrophysiology

---

**Electronic supplementary material** The online version of this article (doi:10.1007/s12021-013-9205-2) contains supplementary material, which is available to authorized users.

---

J. Edwards · E. Daniel · C. Bajaj (✉)  
Department of Computer Science, ICES, The University of Texas,  
Austin, TX, USA  
e-mail: bajaj@ices.utexas.edu

J. Kinney  
Massachusetts Institute of Technology, Cambridge, MA, USA

T. Bartol · T. Sejnowski  
Howard Hughes Medical Institute, Chevy Chase, MD, USA

T. Bartol · T. Sejnowski  
Salk Institute for Biological Studies, La Jolla, CA, USA

T. Sejnowski  
University of California at San Diego, San Diego, CA, USA

D. Johnston · K. Harris  
Center for Learning and Memory, The University of Texas,  
Austin, TX, USA

## Introduction

Brains are richly structured at the cellular and subcellular level as evidenced by the diversity in form of synapses, the compartmentalization of synaptic spines on dendrites, the intricate branching of dendrites and axons, and the complex inter-digitation of glial processes (Sorra and Harris 2000; Andersen et al. 2006). Clinical findings reveal dramatic disruption in the structure and subcellular composition under a variety of neuropathies (Fiala et al. 2002; Ramírez and Couve 2011; Liu et al. 2011; De Rubeis et al. 2012; van Spronsen and Hoogenraad 2010; Kuwajima et al. 2012). Recent advances in imaging are beginning to provide access to an unprecedented amount of structural data from serial section electron microscopy (EM) at nanometer resolution (Denk and Horstmann 2004; Hayworth et al. 2006; Knott et al. 2008; Bock et al. 2011; Kuwajima et al. 2013; Kleinfeld et al. 2011; Briggman and Bock 2012; Helmstaedter and Mitra 2012). A number of software packages have been developed to support three-dimensional

reconstruction from EM images such as RECONSTRUCT™ (Fiala 2005; Lu et al. 2009), TrakEM2 (Cardona et al. 2012), ilastik (Sommer et al. 2011), NeRV (Jurrus et al. 2012), NeuroTrace (Jeong et al. 2010), KNOSSOS (Helmstaedter et al. 2011; Briggman et al. 2011); however, their surface representations were developed primarily for rapid visualization and are insufficient to serve as a framework for dynamical simulations.

Any algorithm for reconstruction of brain geometry from serial sections must confront the challenge posed by structures that are smaller than section thickness (~45 nm) (Harris et al. 2006). Objects within the thickness of the section can be obscured by overlapping structures in the projected EM image. Consequently, ambiguous geometries arise in the reconstruction of fine structure that is under-sampled by the image data and incorrectly represented by extracted contours, frequently yielding 3D objects that are non-physiological, e.g. with aberrant holes in the surface or erroneous connections between cells.

VolRoverN is a new software package that accepts as input the contour tracings from existing software tools, and automatically generates reconstructions that are physiologically plausible and formatted for easy input into other software tools for simulation of neuronal or other cellular dynamics. VolRoverN makes implementations of published algorithms available to practitioners in an intuitive, comprehensive interface, easing the task of model generation. We describe the functionality of VolRoverN, including accurate 3D surface reconstructions from manual contour

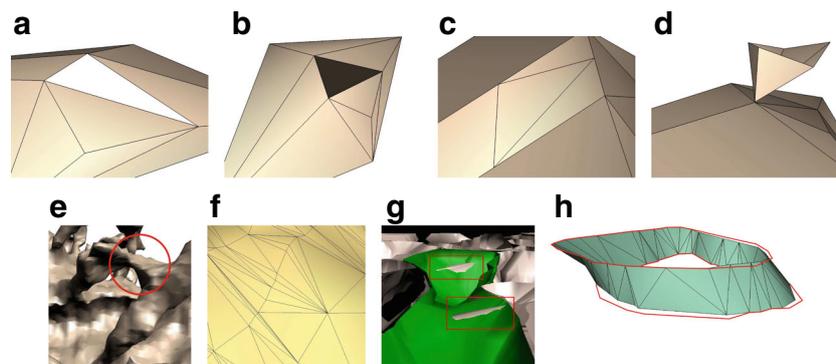
tracings and production of derivative skeletonizations from these reconstructions. We enumerate common errors in surface reconstruction and demonstrate VolRoverN's ability to produce error-free, quality reconstructions.

## Functionality

VolRoverN is freely downloadable at <http://cvcweb.ices.utexas.edu>. It is currently available on the Mac OS X platform, and we anticipate release for Linux and Windows platforms. With the VolRoverN download is a sample dataset with contours and images of 8 axons and 2 dendrites in the CA1 region of the hippocampus. All images in this paper were produced using this dataset. A shared data repository will also be available where users of VolRoverN can share images, traces, 3D meshes, and simulation files for NEURON and MCell.

VolRoverN accepts RECONSTRUCT™ and TrakEM2 contour tracings as input. In the case of TrakEM2, the tracings are pixel-based and are automatically converted to polygonal representation by VolRoverN. Aligned and segmented images can also be imported into VolRoverN for visualization purposes.

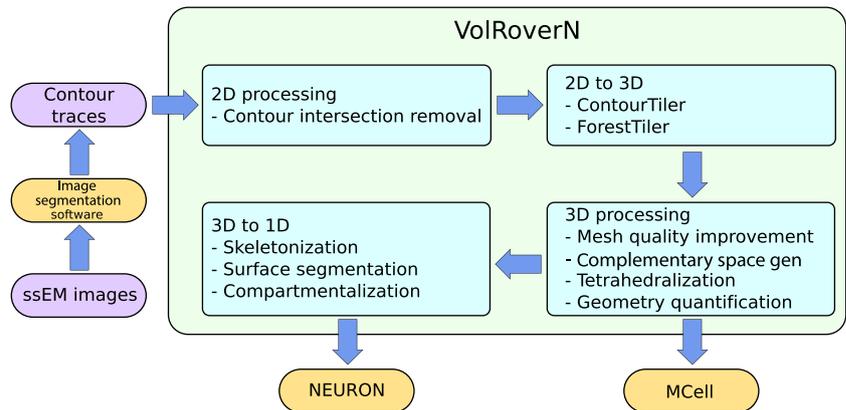
The software first fits a triangulated surface to contours such that the contours are exactly interpolated and the surface meets important quality criteria. We list and show examples of violations of these criteria in Fig. 1. Properties of quality reconstructions include water-tightness,



**Fig. 1** Requirements for quality surface models. The illustrations demonstrate errors commonly encountered in morphological reconstructions of neurons. **a** Surface models should be water-tight, or free of holes. **b** A normal vector is associated with each facet in a mesh representation, and all facets should be consistently oriented. **c** Vertices should not be coincident with edges of other facets. **d** A surface is manifold if an arbitrarily small piece of the surface is a topological disk. In this example the point at which the two surfaces meet is not a topological disk. **e** In this example, two spine heads are erroneously joined during the reconstruction process. Meshes should be topologically consistent with the

physical specimen they are approximating. **f** In a mesh representation, the facets (most commonly triangles) should be of good quality. In the case of a triangle, one definition of quality is that the ratios of edge lengths are close to one, or the triangle is close to equilateral. **g** A very common error in existing neuropil reconstruction methods is object-object intersections. Also common are triangle intersections within the same object. **h** Surface reconstructions should interpolate, or pass exactly through, the original contour tracings (red). The figure shows an example of error in interpolation. Both VolRoverN and marching cubes are error-free in that they exactly interpolate the contours

**Fig. 2** A high-level look at VolRoverN’s functionality. There are four main phases: 2D processing, 2D to 3D reconstruction, 3D processing, and 3D to 1D reduction



manifoldness, lack of intersections, quality (close to equilateral) triangles, and geometric accuracy. With the surface mesh in place the user can make geometric queries, such as surface area and volume of a spine head. Further, VolRoverN provides tools to create derivative models, including 1D cable models. The various models can be saved in standard file formats, including Wavefront obj, OFF, ele/node, MDL (MCell), and HOC (NEURON).

VolRoverN shares a code base with the related software package VolRover 2.0 (Zhang et al. 2012) which performs image processing, reconstruction and visualization of single particle and tomographic cryo-EM and includes 3DEM molecular ultra-structure identification and quasi-atomic model-based refinement. As such, VolRoverN and VolRover 2.0 have similar look and feel, but the tools included in VolRoverN are appropriate primarily for neuronal modeling.

VolRoverN has 4 steps in producing models suitable for analysis (Fig. 2). (1) Process 2D input. (2) Fit a 3D triangulated surface to the contours. (3) Process the 3D

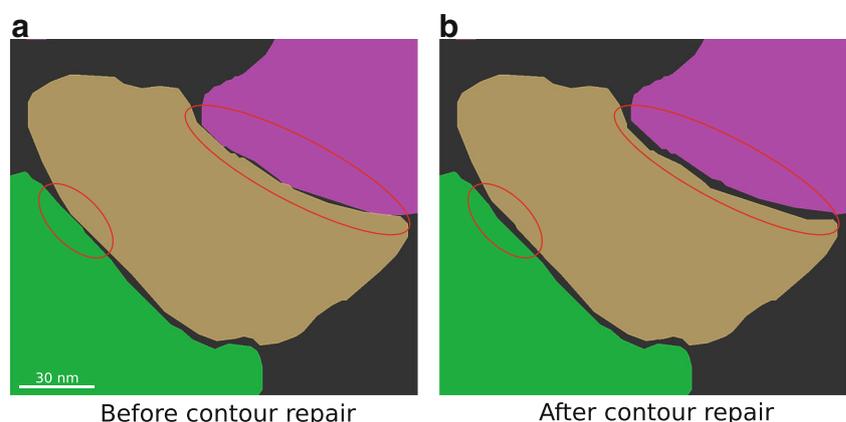
surface meshes, which includes improvement of the mesh. (4) Reduce the mesh to a 1D cable model. We now discuss each of these steps.

## 2D Processing

### Contour Intersection Removal

Contours are given as closed, simple polygons. Contours may be nested, as is the case with surface dimples and organelles. Each contour has a label that corresponds to a unique neuronal process, for example, “axon16” or “dendrite3”. Manually-traced contours can accidentally intersect each other and these overlaps must be removed. VolRoverN has an automatic intersection removal algorithm (Edwards and Bajaj 2011) (Fig. 3) that not only ensures that contours don’t intersect each other, but has an optional feature of guaranteeing a minimum separation distance (Kinney et al. 2013). The 2D intersection removal algorithm is purely geometric and does not take the EM images into consideration.

**Fig. 3** 2D curation. Because components are usually traced independently of each other, intersection errors can occur. **a** A number of intersections and close approaches can be seen between contours. **b** The intersections have been removed and a user-specified contour spacing is enforced



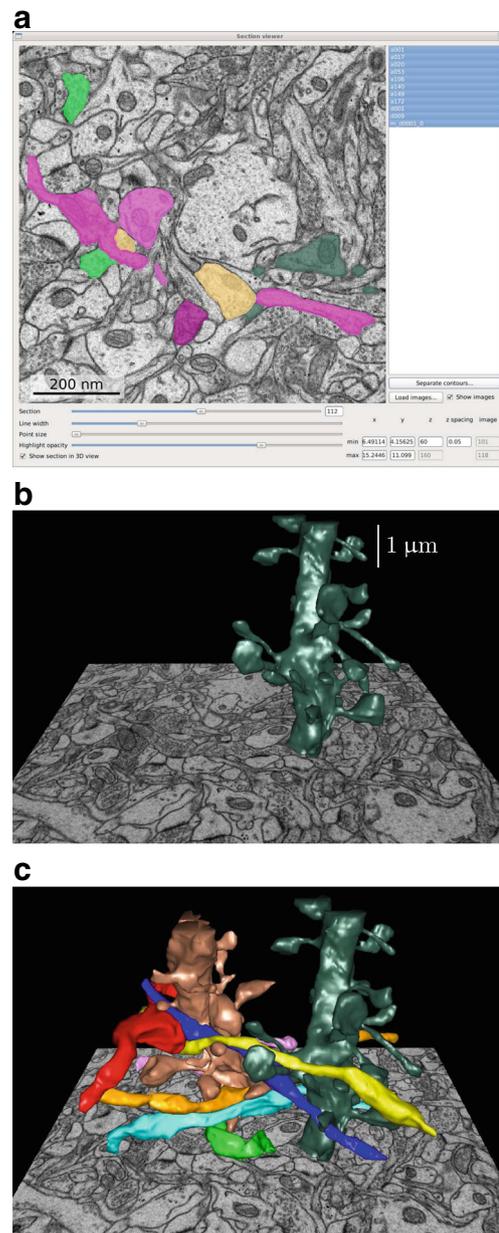
## 2D to 3D

*ContourTiler*

VolRoverN has a 2D image and contour display called the Section Viewer (Fig. 4a) that enables navigation through sections while inspecting contours. The Section Viewer and 3D Viewer are linked: imagery and semi-transparent contours can be visualized in the 3D view alongside surface meshes and volumes (Fig. 4b, c). VolRoverN includes a tool called ContourTiler (Bajaj et al. 1996; Edwards and Bajaj 2011) that fits a triangulated surface to a set of 2D polygonal contours derived from EM images. We call these triangles “tiles”. Three attributes of neuronal data make reconstruction non-trivial: (1) ssEM data is highly anisotropic. In-plane pixel resolution is usually about 2–5 nm, whereas spacing between sections is usually 45–70 nm. (2) Neurons are tortuous, with frequent twists and branching. (3) Neuronal processes are tightly packed. These three attributes make reconstruction of intersection-free surface meshes especially difficult for such complex morphologies. Other cell types, such as glia, can have similarly complex surfaces.

ContourTiler matches adjacent contours with identical labels and fits a surface between them. Matching is commonly referred to as the correspondence problem, i.e., whether two contours with identical labels in adjacent sections should be connected topologically. The correspondence problem is difficult and many methods have been used to solve it, including user consensus (Helmstaedter et al. 2011) and machine learning (Chklovskii et al. 2010). Our algorithm uses a simple heuristic, that of matching contours from adjacent sections together if their projections onto the plane intersect, which results in sensitivity to image registration quality. Our heuristic can also result in errors when small contours that are oriented obliquely to the image plane have grey boundaries that are difficult to trace, which may result in corresponding contours that don't overlap. Technology development in high-throughput microscopy promises to reduce the thickness of tissue sections (REF), which will lead to asymptotically better performance of our simple heuristic. Nevertheless, we anticipate incorporating more sophisticated correspondence predicates in the future.

Our tiling algorithm requires that the contours have no intersections, which is taken care of in our 2D contour intersection removal step. In addition, the contours are assumed to be registered, or aligned. Registration is typically done using the EM images (Saalfeld et al. 2012) and the resulting 2D transformations are applied to the contours. An important property of our reconstruction approach is that the reconstructed surface exactly interpolates the input contours, so no error in the original section planes is introduced.

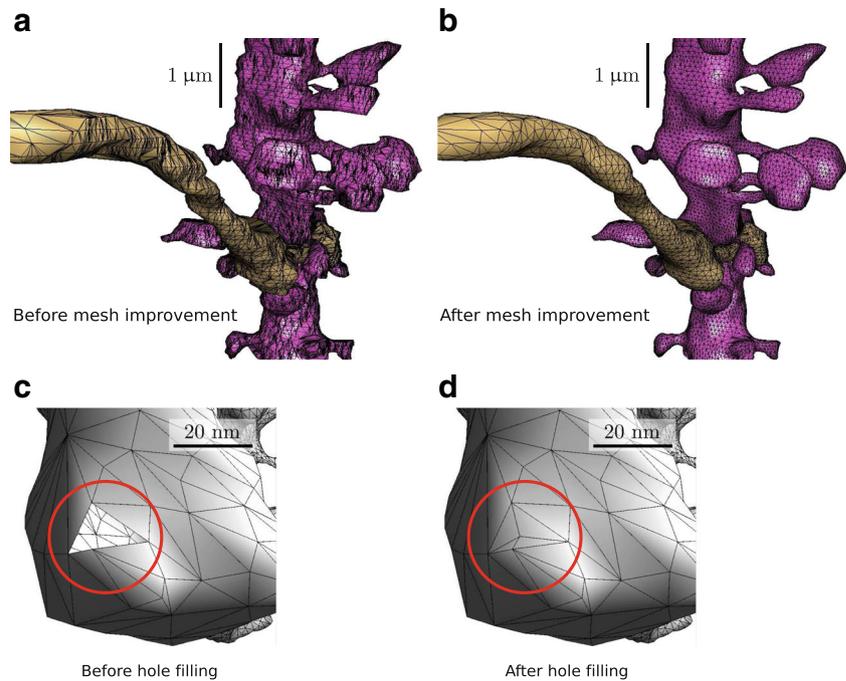


**Fig. 4** Reconstruction from 2D contours. **a** Input to VolRoverN is a set of 2D polygonal traces, or contours, derived from EM images. **b** Software embedded in VolRoverN called ContourTiler fits a triangulated surface to each set of contours to produce a 3D surface model. **c** Multiple components are combined using ForestTiler such that they are free of intersections

*ForestTiler*

ContourTiler produces a surface for each individual object. These surfaces are then combined together into a single geometry file that contains all of the objects. When the section spacing is very large, as is the case with anisotropic data, combining object meshes yields many

**Fig. 5** Mesh improvement. **a** The original reconstructed triangulation of a dendrite and axon. The horizontal bands in the surface mesh reflect the separation between EM images of thin tissue sections. **b** The reconstruction after decimation and smoothing. The final triangulation has fewer than half the triangles as the original and the triangles have far better aspect ratio. **c** Repair utilities in VolRoverN include manifold correction and hole filling. The figure shows a hole outlined in red. **d** After hole filling. The before/after ratio of total mesh surface area in this example was 32.7/32.8, for a total hole surface area of 0.3 %



object-object intersections. Most solutions to the intersection problem are primarily used with reconstruction algorithms that tend not to preserve correct topology in tightly-packed data (Turk and O'Brien 1999). We have implemented an algorithm called ForestTiler (Edwards and Bajaj 2011) that removes intersections in a way that preserves the interpolation property of the original reconstruction. In addition, similar to 2D contour curation, the user can specify a minimum separation distance  $\delta$ , which corrects the reconstruction of unknown regions (between sections) to more closely match empirically determined extracellular spacings (Kinney et al. 2013). ForestTiler does not take membrane junctions into account, but a small value for distance  $\delta$  may be specified to preserve very close spacing.

### 3D Processing

#### Mesh Quality Improvement

VolRoverN includes a suite of tools to produce meshes with good quality triangles, i.e. triangles that are close to equilateral. The first tool is decimation, which uses the QSlim algorithm (Garland 2004) to reduce the number of triangles. QSlim is an edge-collapse algorithm that is popular because of its speed and robustness. We then use the geometric-flow mesh improvement algorithm of Zhang et al. (2005, 2009) that produces a surface mesh with triangles of good aspect ratio (Fig. 5a, b). The mesh improvement algorithm can be iteratively applied for increasing triangle quality. Quality improved meshes are not guaranteed to meet the

contour interpolation property. VolRoverN also has mesh repair utilities (Fig. 5c, d) to repair errors such as holes, non-manifoldness and self-intersections. Non-manifoldness and self-intersections are repaired by automatic removal of offending triangles. Resulting gaps are then closed using our hole-filling tool, which uses an ear-clipping algorithm (O'Rourke 1994). Our mesh repair utilities are general and can be applied to problematic meshes from other reconstruction software packages. Simplification, improvement, and mesh fixing are intended to be applied as follows:

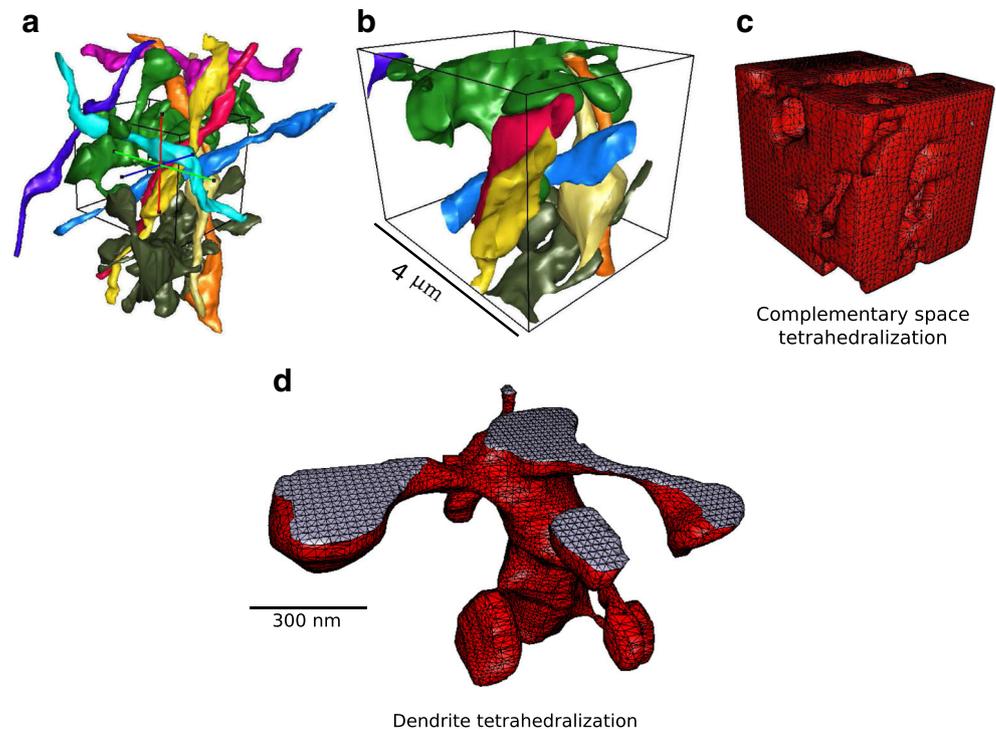
1. Simplification
2. Improvement
3. Repair
4. Improvement

Repair is generally more effective if the model has already been simplified and improved. It is often best to finish with the improvement step as repair can introduce poor quality triangles.

#### Complementary Space and Tetrahedralization

A feature of VolRoverN is automatic complementary space (CS) generation outside of (and complementary to) watertight surface meshes and inside a bounded region of interest. The user is able to define a bounding box and the CS tool constructs a closed polyhedron with faces consisting of portions of the surface meshes and the bounding box (Fig. 6). The CS polyhedron converges to a model of extracellular space (ECS) when the input surface meshes are a complete

**Fig. 6** VolRoverN utilities. **a** A bounding box is placed around the surface meshes in an area of interest. **b** The surfaces are clipped at the bounding box. **c** VolRoverN's ECS and tetrahedralization tools create a volumetric model of complementary space. **d** Tetrahedralization of a dendrite using VolRoverN's tetrahedralization tool



description of cells within a bounding box. Models of ECS have been used in reaction-diffusion simulations (Kinney 2009; Kinney et al. 2013).

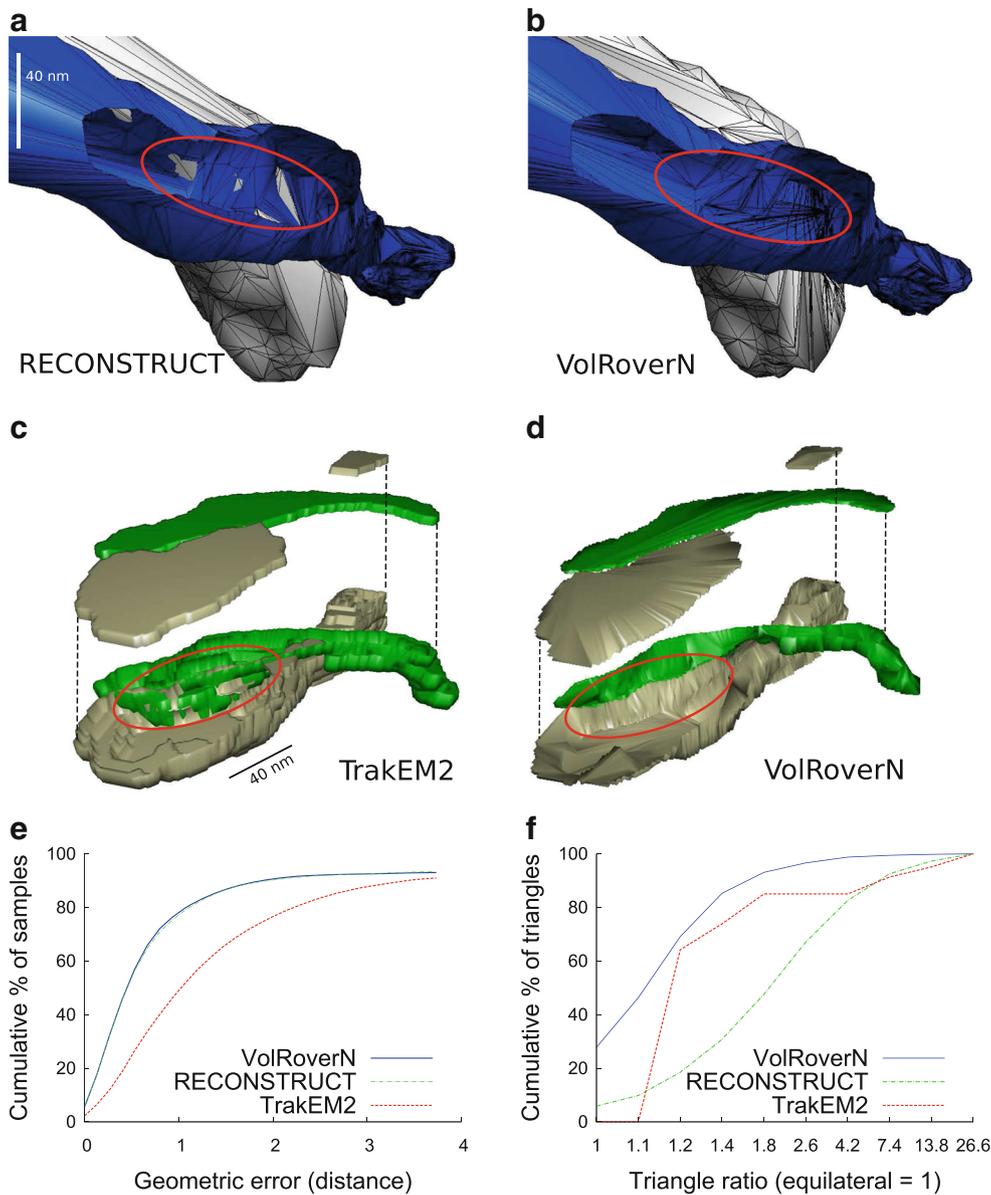
VolRoverN also has the capability of tetrahedralizing surface meshes (Bajaj et al. 1999; Zhang and Bajaj 2006; Zhang et al. 2010) (Fig. 6d). This uses an adaptive subdivision meshing algorithm contained in a library from our Level Set Boundary-Interior-Exterior (LBIE) software package (CVC 2013). Tetrahedra are exported in RAW and ele/node formats for ease of import into simulation packages such as STEPS (Hepburn et al. 2012).

### Comparison to Other Surface Reconstructions

Standard algorithms such as marching cubes, which is used in most popular contouring software packages (Cardona et al. 2012; Sommer et al. 2011; Jurrus et al. 2012), yield large numbers of intersections between objects when the objects are tightly packed (Fig. 7). Additionally, marching cubes produces blocky reconstructions with poor geometric accuracy due to its lack of interpolation between contours. This is especially evident with neuronal EM images that are highly anisotropic. Our algorithm resolves both problems, producing surface meshes that are intersection-free and linearly interpolated between contours.

Here we compare ContourTiler with the implementation of marching cubes found in TrakEM2 (ImageJ)

(Schmid et al. 2010; Cardona et al. 2012) and the Boissonnat algorithm implemented in the RECONSTRUCT software (Boissonnat and Geiger 1992). We compared object-object intersections and geometric accuracy, as well as water-tightness and manifoldness. Other important surface mesh qualities (oriented normals, regularity, topological correctness, and contour interpolation) are met by all three algorithms. We reconstructed the 8 axons and 2 dendrites in the sample dataset distributed with VolRoverN using both RECONSTRUCT and VolRoverN and compared the results by quantifying the most common errors of those described in Fig. 1. Figure 7a–d show reconstruction of two axons in close proximity (a001 and a020 from the sample dataset) using the three reconstruction methods. VolRoverN meshes are free of intersections between multiple objects, in contrast to surfaces produced by the marching cubes and the Boissonnat algorithms. The Boissonnat and marching cubes representations yield a large number of intersections, whereas the VolRoverN surfaces are entirely free of intersections because VolRoverN meshes are guaranteed to have a user-specified minimum spacing between objects. Meshes produced by VolRoverN interpolate, or pass exactly through, the input contours. To quantify surface error in regions between contours we compared surfaces produced by VolRoverN and the other two algorithms to a  $C^1$ -continuous surface fitted to the contours. The distribution of errors is reported in Fig. 7e and shows that VolRoverN's reconstructions are geometrically very similar to the RECONSTRUCT



**Fig. 7** Comparisons with RECONSTRUCT™ and TrakEM2. **a–b** Comparison between RECONSTRUCT™ (Boissonnat) and VolRoverN surfaces using axons a001 and a020 from the sample dataset. Part of a020 is cut out to see the interior intersections. The RECONSTRUCT surfaces yield a large number of intersections between objects. Output from ForestTiler is intersection-free. **c–d** Comparison between TrakEM2 and VolRoverN surfaces. A portion of two axons are reconstructed with ImageJ’s marching cubes implementation (resample=1, no smoothing) and the top is lifted to reveal the interior. Marching cubes yields numerous intersections between objects. Output from ForestTiler is intersection-free. **e** Geometric error measured as distance from a reconstruction to a  $C^1$ -continuous approximating surface. This is a cumulative plot of percentage of

sample points within a given error. 100,000 sample points were taken. VolRoverN and RECONSTRUCT have far less geometric error than TrakEM2. To create the  $C^1$ -continuous surface  $S_{C^1}$ , we randomly choose 4 adjacent, non-bifurcating contours (called  $c_1$ ,  $c_2$ ,  $c_3$ , and  $c_4$ ) and fit cubic B-splines to each of them using a least-squares fit. We then join the contours together with interpolating cubic curves, forming a patch that is  $C^1$ -continuous everywhere between  $c_1$  and  $c_2$ . The data used in this test are from axon a001 between slices 115 and 118. **f** Comparison of quality of triangles between the three reconstruction methods. We define triangle ratio as  $r_c/2r_i$  where  $r_c$  is the circumradius and  $r_i$  is the inradius of a triangle. The ideal triangle ratio is 1. The plot is a cumulative percentage of triangles below a given ratio

implementation of the Boissonnat algorithm, both of which are closer to a smooth approximation than marching cubes.

Triangle quality is another important measure of how successful a simulation will likely be in terms of error

convergence (Shewchuk 2002). Figure 7f illustrates how VolRoverN outperforms both other methods in terms of triangle aspect ratio (ratio of the circumradius to twice the inradius).

## Using VolRoverN Output in Simulations

### 3D to 1D Skeletonization and Surface Segmentation

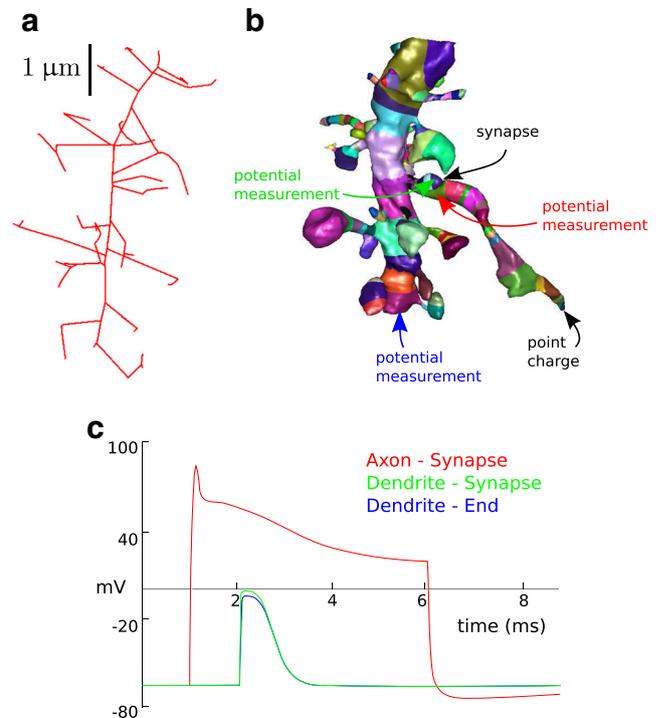
Cable model simulation requires 1D skeleton models of neurons. Cable models are typically created with neurite tracing software, but VolRoverN utilizes surface meshes to generate 1D models automatically. VolRoverN decomposes the mesh into cylindrical chunks using an algorithm that first finds a skeleton of the surface mesh (Fig. 8a) using an iterative Laplace contraction algorithm (Au et al. 2008). The skeletonization algorithm induces a surface segmentation as a side effect. The segmentation is not perfect, so VolRoverN performs operations to repair and smooth the segmentation, resulting in regions such as those shown in Fig. 8b. Each of these regions is approximated with a 1D segment with length and diameter (and thereby volume and surface area) properties. These segments are typically conceptualized as generalized cylinders. Our decomposition algorithm preserves volume, that is, the sum of all cylindrical region volumes equals the volume of the polyhedron. VolRoverN reports surface area and volume of regions of a reconstructed object. After surface segmentation, the user can discover geometric measurements of segments (e.g. Fig. 8b) by clicking on the region.

### NEURON

VolRoverN's skeletonization tools are used to automatically reduce the surface representation to a multicompartmental cable model and simulate ion channel-driven dynamics of membrane voltage in NEURON (Carnevale and Hines 2006). NEURON is a simulation software that implements multi-compartment models of electrical signaling based on cable theory (Fig. 8c). VolRoverN outputs to a NEURON HOC file, which contains length and diameter properties for each region as well as connection properties defining their topological connectedness. The output HOC file also contains a skeleton simulation function. As reduction to 1D includes a coarsening of resolution, we expect NEURON simulation results to be similar regardless of reconstruction algorithm used.

### MCell

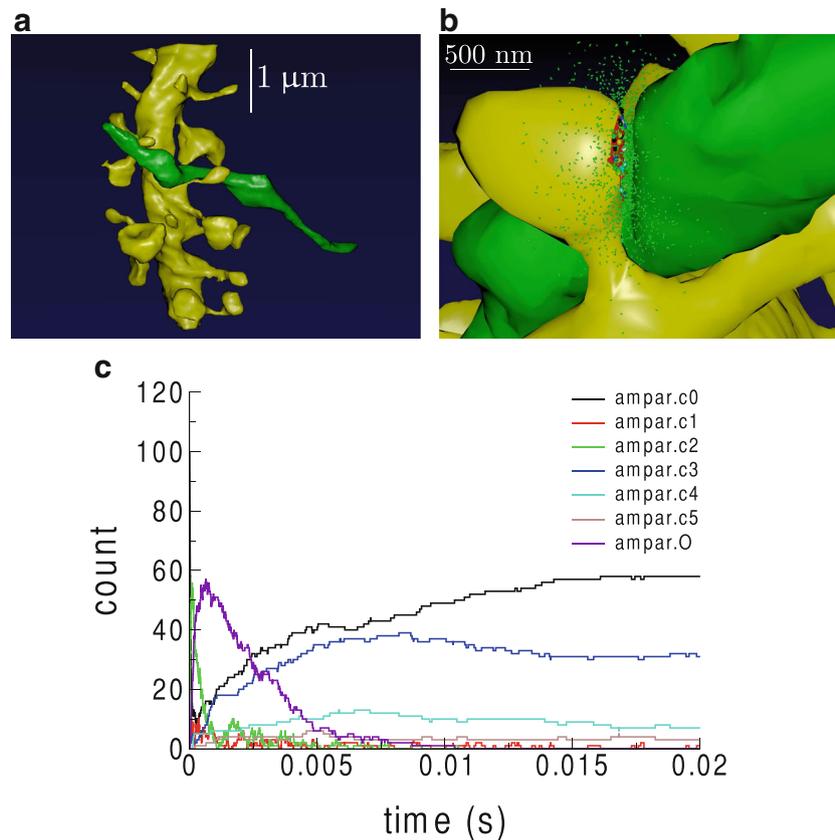
VolRoverN also enables MCell simulations. MCell (Stiles et al. 2001; Kerr et al. 2008) is a software package that simulates multi-ion species reaction-diffusion using Monte Carlo algorithms over geometrically complex domains. VolRoverN's ForestTiler and mesh improvement tools were used to generate surface triangulations of an individual axon and dendrite in the VolRoverN sample (Fig. 9a). The MCell export tool in VolRoverN writes a given surface mesh to an



**Fig. 8** Multi-compartment model generation. Our surface segmentation first skeletonizes the mesh (a), which induces a segmentation (b). Each segment is in a different color in the figure. After correction, the segmentation can be used to produce surface area/volume statistics of different regions as well as labeling different regions for ion diffusion studies. This graphic shows a simple cable model simulation. The compartmentalized versions of the axon and dendrite are input to NEURON. A synapse with a threshold and delay is added between the dendrite and axon and a point charge is placed at the end of the axon. Three potential measurements are made over time. Arrow colors correspond the potential measurements reported in the NEURON simulation graph in figure (c)

MCell MDL file. As noted, these meshes are required to be water-tight, manifold, and free of self-intersections which can be repaired, if necessary, using VolRoverN's mesh repair utilities. A complete physiological simulation study was set up with the CellBlender software package (<http://www.mcell.org>) using the MCell MDL geometry files. The geometric analysis tools in CellBlender confirmed that the imported MDL meshes were of computational quality for use in simulations. CellBlender was then used to generate and run an MCell simulation of glutamatergic synaptic transmission at a synapse between the axon and dendrite (Fig. 9b). Figure 9c shows the time course of activation of synaptic receptors by diffusing neurotransmitter molecules released at the synapse.

Direct comparison of MCell simulations using VolRoverN, Boissonnat, and marching cubes models has not been informative because reaction-diffusion simulations such as MCell require geometric consistency. Meshes with errors are rejected outright with an error message. As shown



**Fig. 9** MCell reaction/diffusion simulation of synaptic transmission from generated model. Generated meshes of axon (green) and dendrite (yellow) were imported into CellBlender to create an MCell simulation from the meshes. Images were rendered using CellBlender. **a** The axon and dendrite. **b** Visualization of synaptic transmission 100 microseconds after release of 2000 molecules of the neurotransmitter glutamate (small green ellipsoids). 10 NMDA receptors (NMDAR) and 100 AMPA receptors (AMPA) were placed at the synaptic contact area between the axon and dendrite (small red patch of membrane

on the dendrite). Color indicates state of activation of the receptors. At 100 microseconds, the glutamate has started to bind and activate some receptors and has started to spill out of the synaptic cleft space into the surrounding volume. **c** Time course of activation of AMPARs. AMPAR can be in 7 states: c0 (unbound state), c1 (one glutamate bound), c2 (two glutamate bound), c3 (one glutamate bound, desensitized state 1), c4 (two glutamate bound desensitized state 2), c5 (two glutamate bound, desensitized state 3), and O (two glutamate bound, ion channel open)

in Fig. 7a and c, Boissonnat and marching cubes models have large numbers of intersections between objects, causing attempted simulations to fail.

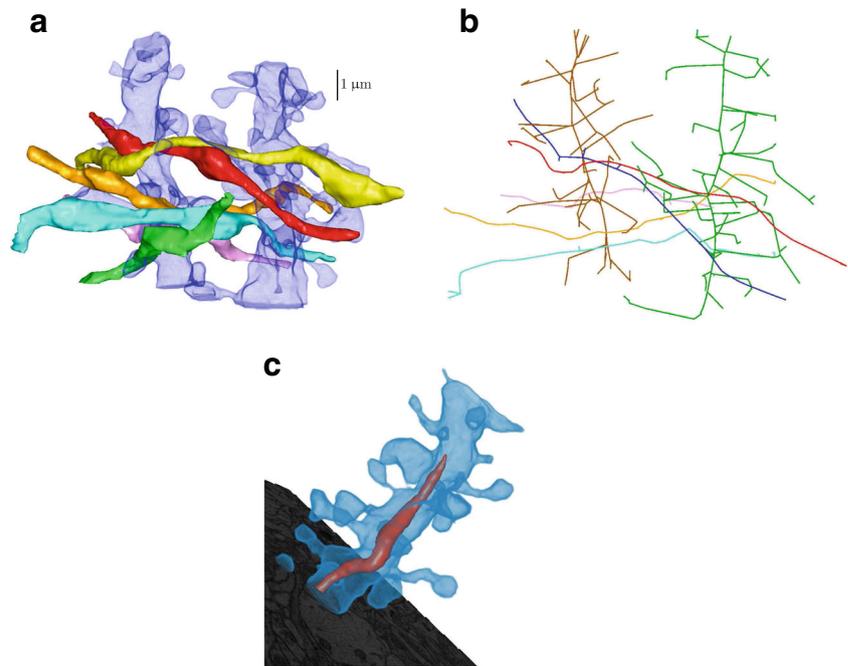
### Volume Rendering and Isocontour Visualization

VolRoverN's volume rendering capabilities and isocontour visualization tools enable interactive visual exploration of topology and geometry. Volume renderings include the capacity to illustrate numerous objects of varying color and transparencies (Fig. 10a) which can also be illustrated as NEURON-ready skeletons (Fig. 10b) and include intracellular organelles (Fig. 10c).

The signed distance function (SDF) in VolRoverN is a tool that produces a volume of scalar values (a volumetric scalar field) representing the distance from a sur-

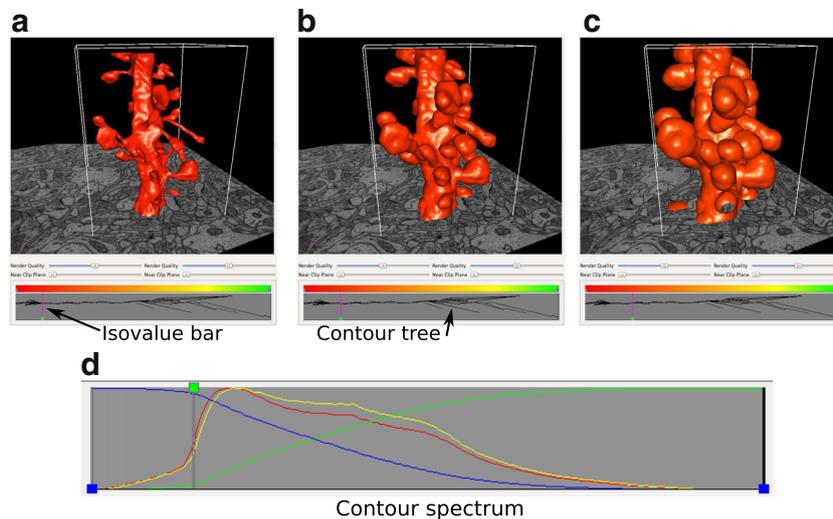
face. SDF, together with VolRoverN's volume rendering capabilities (Fig. 10) and isocontour visualization tools, enables interactive visual exploration of topology and geometry of isosurfaces (also called level sets) at various isovalues (level-set values) of the magnitude of the SDF. VolRoverN volume renders scalar fields (stored in HDF5 format) and geometries together and seamlessly. The transfer function tool supports both color and transparency ramps across the spectrum of level-set values in a volume and is used to color and achieve see-through translucency. The fast isocontour visualization (Bajaj et al. 1996) capability within VolRoverN enables interactive visual exploration of volumetric scalar fields through rendering of isosurfaces generated for distinct isovalues. Fast isocontour visualization when applied to SDF of input surfaces of complicated topology and geometry, provides for a quick visual surface exploration of the topological and geometric complexity

**Fig. 10** Volume rendering with geometry rendering. **a** Axons rendered with semi-transparent dendrites. **b** A view of a skeletonization of all axons and dendrites in the sample dataset. Skeletons can be saved in OFF and raw file formats. **c** A dendrite is rendered with semi-transparent volume rendering to reveal mitochondria. ForestTiler naturally supports nested components



(Fig. 11). SDF fields of surfaces produced by ForestTiler can be examined for changes in their level-set topology and geometry. Fast isocontour visualization also provides a means to explore visually the results of function fields

produced in simulations. A supporting topological visualization tool is the contour tree (Carr et al. 2000; Zhang et al. 2006) where branching reveals the splitting of level sets across the entire range of level-set values (Fig. 11a–c).



**Fig. 11** Iso-surfaces of a signed distance scalar field generated from a triangulated surface model of a dendrite, and at different isovalues. Iso-surfaces are computed from surface geometries and are useful in characterizing object shape. The contour tree at bottom shows the topological branching structure of the isosurface. The vertical line in the contour tree shows the isovalue of the surface relative to the tree. **a** is close to the true surface, as at that isovalue the contour tree is a confluence of branches into one. **b** and **c** are isosurfaces at progressively larger absolute distance values. By convention, negative distance

values indicate points outside the object, while positive values are for points inside. **d** The contour spectrum tool's signature properties plot window. Four attribute curves are shown: surface area (*red*); min volume (*green*); max volume (*blue*); gradient weighted surface area (*yellow*). The *green* isovalue node is at an isovalue for which surface area and gradient weighted surface area curves are close to their respective maximum values, which usually occurs near the zero isovalue. The x axis represents distance from the original surface. Each curve is normalized with respect to the y axis

The contour spectrum (Bajaj et al. 1997) plot gives additional insight, showing signature curves representing the variation of isosurface properties such as area, volume and gradient (normal) magnitude across isovalues (Fig. 11d). The spectrum plot helps quickly locate isovalues where the isosurface properties become critical (e.g. maximum, minimum, etc.).

### Scalability and Performance

The algorithms used in VolRoverN are scalable. Our reconstruction method linearly interpolates between sections, so only two sections need be stored in memory at one time, thus the memory requirements remain static with increasing stack size. Memory also remains static with increasing image resolution since the reconstruction algorithm uses exclusively geometric contours and not the original images. However, memory requirements do increase with larger numbers of contours per image (such as would be the case with larger image footprints), but memory goes up proportionally to the number of contours. The fact that the VolRoverN reconstruction algorithm operates only on pairs of sections makes it an excellent candidate for parallelization, a feature for future development.

ForestTiler and associated mesh improvement tools in VolRoverN are efficient. We tested reconstruction time on the sample dataset, which consists of portions of 8 axons and 2 dendrites with a combined  $129.32 \mu\text{m}^2$  surface area,  $8.44 \mu\text{m}^3$  volume and 204,906 triangles. Reconstruction took 6 min and 40 sec on a Linux Kubuntu workstation with Intel Xeon 3.20 GHz quad core CPU and 4 GB of memory. Decimation, triangle improvement and mesh fixing tools took an additional 56 sec.

### Discussion

VolRoverN plays a complementary role in the set of neuronal morphology software. It provides tools to enhance geometric understanding of 2D tracings and offers an alternative method of skeletonizing neurites. VolRoverN also fills a critical gap, in that it produces meshes that are manifold, geometrically accurate, water-tight, and free of intersections. Before now, producing such reconstructions required a large amount of manual work, but VolRoverN's powerful tools greatly reduce the amount of time and domain knowledge required to prepare reconstructions for geometric analysis. Reconstructions serve as substrate for dynamical simulation of cellular activity.

VolRoverN accepts geometric contours as input. Thus, success in quality reconstruction is at least partly dependent

on the quality of the contours produced using other software tools. This is largely mitigated by visual proofing tools in tracing software (Fiala 2005; Lu et al. 2009; Cardona et al. 2012). However, if surface reconstruction from VolRoverN reveals errors in the contour tracings then the user can revert back to the original tracing software for contour repair at well-identified locations.

Representations of the cell surface enable simulations using 3D boundary element methods (BEM) such as combined Monte Carlo simulation of particle diffusion and kinetic state-based modeling of protein dynamics at the microsecond time scale by MCell (Kinney 2009; Kinney et al. 2013). 3D finite element method (FEM) simulations (Bajaj et al. 2010) of electro-diffusion with multi-species continuum concentrations are enabled by decomposing a reconstruction into a collection of small volumes. Also, by approximating neuronal geometry as a collection of cylindrical compartments each aligned to segments of the geometry (Lindsay et al. 2004), and modeling ionic conductance in each compartment with coupled differential equations, one arrives at the traditional cable simulation of electrical signals in the brain at the millisecond scale, as supported by simulation software such as NEURON (Carnevale and Hines 2006). Deriving these geometric representations of brain structure from a single source (surface meshes) facilitates future multi-scale simulations via coupled MCell, FEM and NEURON models.

*Third-Party Libraries* VolRoverN uses several third-party software libraries. The reconstruction tools use CGAL (2013) and QSlim (Garland 2004). Rendering is done using OpenGL and the interface uses the cross-platform GUI package QT.

*Data Sources* Data used in the figures is from hippocampal area CA1 of a postnatal day 77, perfusion-fixed, male rat (Harris and Stevens 1989; Mishchenko et al. 2010). All images are ssEM with 2 nm in-plane and 45 nm out-of-plane resolution. Image size is  $4\text{K} \times 4\text{K}$  pixels. The dataset is available as a sample with download of VolRoverN. All rendered figures in this paper were produced using VolRoverN except plots (prepared by gnuplot) and the MCell simulation which used CellBlender (Fig. 9a, b).

### Information Sharing Statement

VolRoverN binaries, a tutorial, and a sample dataset are freely downloadable at <http://cvcweb.ices.utexas.edu>. Source code is freely available at the same link for non-profit institutions. We anticipate the future addition of a

public data repository on the same page for images, traces, 3D meshes, and simulation files.

**Acknowledgments** We thank the anonymous reviewers for their many constructive comments that greatly improved the manuscript. Jose Rivera, Josef Spacek, Deborah Watson, Michael Chirillo and John Mendenhall assisted in various phases of this work. This work of CB, JE, ED was funded in part by NIH contracts R01-EB00487, R01-GM074258, and a grant from the UT-Portugal colab project. Work by TJS and TMB, was supported by NSF grant EMI9822, NIH grants MH079076 and P41-GM103712, that of DJ by NIH grants, MH048432 and MH094838 and of KH by NIH grants NS074644, NS21184, and the Texas Emerging Technologies Fund.

## References

- Andersen, P., Morris, R., Amaral, D., Bliss, T., O'Keefe, J. (2006). *The hippocampus book*. USA: Oxford University.
- Au, O.K.C., Tai, C.L., Chu, H.K., Cohen-Or, D., Lee, T.Y. (2008). Skeleton extraction by mesh contraction. *ACM Transactions on Graphics*, 27(44), 1–10.
- Bajaj, C., Bettadapura, R., Lei, N., Mollere, A., Peng, C., et al. (2010). Constructing A-spline weight functions for stable WEB-spline finite element methods. In *Proceedings of the 14th ACM symposium on solid and physical modeling* (pp. 153–158). ACM.
- Bajaj, C., Coyle, E., Lin, K. (1996). Arbitrary topology shape reconstruction from planar cross sections. *Graphical Models and Image Processing*, 58, 524–543.
- Bajaj, C., Coyle, E., Lin, K. (1999). Tetrahedral meshes from planar cross sections. In *Computer methods in applied mechanics and engineering* (pp. 31–52).
- Bajaj, C., Pascucci, V., Schikore, D. (1996). Fast isocontouring for improved interactivity. In *Proceedings of the 1996 symposium on volume visualization* (pp. 39–46). IEEE.
- Bajaj, C., Pascucci, V., Schikore, D. (1997). The contour spectrum. In *Proceedings of the 8th conference on visualization'97* (pp. 167–173). IEEE Computer Society Press.
- Bock, D.D., Lee, W.C.A., Kerlin, A.M., Andermann, M.L., Hood, G., et al. (2011). Network anatomy and in vivo physiology of visual cortical neurons. *Nature*, 471, 177–182.
- Boissonnat, J., & Geiger, B. (1992). Three-dimensional reconstruction of complex shapes based on the Delaunay triangulation. In *Proceedings of SPIE* (Vol. 964, pp. 964–975).
- Briggman, K.L., & Bock, D.D. (2012). Volume electron microscopy for neuronal circuit reconstruction. *Current Opinion in Neurobiology*, 22, 154–161.
- Briggman, K.L., Helmstaedter, M., Denk, W. (2011). Wiring specificity in the direction-selectivity circuit of the retina. *Nature*, 471, 183–188.
- Cardona, A., Saalfeld, S., Schindelin, J., Arganda-Carreras, I., Preibisch, S., et al. (2012). TrakEM2 software for neural circuit reconstruction. *PLoS One*, 7, e38011.
- Carnevale, N., & Hines, M. (2006). *The NEURON book*. Cambridge: Cambridge University.
- Carr, H., Snoeyink, J., Axen, U. (2000). Computing contour trees in all dimensions. In *Proceedings of the eleventh annual ACM-SIAM symposium on discrete algorithms* (pp. 918–926). Society for Industrial and Applied Mathematics.
- CGAL (2013). *Computational geometry algorithms library*. <http://www.cgal.org>.
- Chklovskii, D.B., Vitaladevuni, S., Scheffer, L.K. (2010). Semi-automated reconstruction of neural circuits using electron microscopy. *Current Opinion in Neurobiology*, 20, 667–675.
- CVC (2013). LBIE: level set boundary interior and exterior mesher. [http://cvcweb.ices.utexas.edu/cvcwp/?page\\_id=103](http://cvcweb.ices.utexas.edu/cvcwp/?page_id=103).
- De Rubeis, S., Fernández, E., Buzzi, A., Di Marino, D., Bagni, C. (2012). Molecular and cellular aspects of mental retardation in the fragile x syndrome: from gene mutation/s to spine dysmorphogenesis. In *Synaptic plasticity* (pp. 517–551).
- Denk, W., & Horstmann, H. (2004). Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure. *PLoS Biology*, 2, e329.
- Edwards, J., & Bajaj, C. (2011). Topologically correct reconstruction of tortuous contour forests. *Computer-Aided Design*, 43, 1296–1306.
- Fiala, J. (2005). Reconstruct: a free editor for serial section microscopy. *Journal of Microscopy*, 218, 52–61.
- Fiala, J., Spacek, J., Harris, K. (2002). Dendritic spine pathology: cause or consequence of neurological disorders? *Brain Research Reviews*, 39, 29–54.
- Garland, M. (2004). Qslim. <http://mgarland.org/software/qslim.html>.
- Harris, K., & Stevens, J. (1989). Dendritic spines of CA 1 pyramidal cells in the rat hippocampus: serial electron microscopy with reference to their biophysical characteristics. *The Journal of Neuroscience*, 9, 2982.
- Harris, K.M., Perry, E., Bourne, J., Feinberg, M., Ostroff, L., et al. (2006). Uniform serial sectioning for transmission electron microscopy. *The Journal of Neuroscience*, 26, 12101–12103.
- Hayworth, K., Kasthuri, N., Schalek, R., Lichtman, J. (2006). Automating the collection of ultrathin serial sections for large volume tem reconstructions. *Microscopy and Microanalysis*, 12, 86–87.
- Helmstaedter, M., & Mitra, P.P. (2012). Computational methods and challenges for large-scale circuit mapping. *Current Opinion in Neurobiology*, 22, 162–169.
- Helmstaedter, M., Briggman, K., Denk, W. (2011). High-accuracy neurite reconstruction for high-throughput neuroanatomy. *Nature Neuroscience*, 14, 1081–1088.
- Hepburn, I., Chen, W., Wils, S., De Schutter, E. (2012). STEPS: efficient simulation of stochastic reaction-diffusion models in realistic morphologies. *BMC Systems Biology*, 6, 1752–0509.
- Jeong, W., Beyer, J., Hadwiger, M., Blue, R., Law, C., et al. (2010). Ssecret and neurotrace: interactive visualization and analysis tools for large-scale neuroscience data sets. *IEEE Computer Graphics and Applications*, 30, 58.
- Jurrus, E., Watanabe, S., Giuly, R.J., Paiva, A.R., Ellisman, M.H., et al. (2012). Semi-automated neuron boundary detection and nonbranching process segmentation in electron microscopy images. *Neuroinformatics*, 11(1), 1–25.
- Kerr, R., Bartol, T., Kaminsky, B., Dittrich, M., Chang, J., et al. (2008). Fast Monte Carlo simulation methods for biological reaction-diffusion systems in solution and on surfaces. *SIAM Journal on Scientific Computing*, 30, 3126.
- Kinney, J. (2009). *Investigation of neurotransmitter diffusion in three-dimensional reconstructions of hippocampal neuropil*. Ph.D. thesis, University of California, San Diego.
- Kinney, J.P., Spacek, J., Bartol, T.M., Bajaj, C.L., Harris, K.M., et al. (2013). Extracellular sheets and tunnels modulate glutamate diffusion in hippocampal neuropil. *Journal of Comparative Neurology*, 521, 448–464.
- Kleinfeld, D., Bhariok, A., Blinder, P., Bock, D., Briggman, K.L., et al. (2011). Large-scale automated histology in the pursuit of connectomes. *The Journal of Neuroscience*, 31, 16125–16138.
- Knott, G., Marchman, H., Wall, D., Lich, B. (2008). Serial section scanning electron microscopy of adult brain tissue using focused ion beam milling. *The Journal of Neuroscience*, 28, 2959–2964.
- Kuwajima, M., Spacek, J., Harris, K.M. (2012). Beyond counts and shapes: studying pathology of dendritic spines in the context of the

- surrounding neuropil through serial section electron microscopy. *Neuroscience*, 251, 75–89.
- Kuwajima, M., Mendenhall, J.M., Lindsey, L.F., Harris, K.M. (2013). Automated transmission-mode scanning electron microscopy (tsem) for large volume analysis at nanoscale resolution. *PLoS One*, 8, e59573.
- Lindsay, K., Rosenberg, J., Tucker, G. (2004). From Maxwell's equations to the cable equation and beyond. *Progress in Biophysics and Molecular Biology*, 85, 71–116.
- Liu, M., Duggan, J., Salt, T.E., Cordeiro, M.F. (2011). Dendritic changes in visual pathways in glaucoma and other neurodegenerative conditions. *Experimental Eye Research*, 92, 244–250.
- Lu, J., Fiala, J., Lichtman, J. (2009). Semi-automated reconstruction of neural processes from large numbers of fluorescence images. *PLoS One*, 4, 5655.
- Mishchenko, Y., Hu, T., Spacek, J., Mendenhall, J., Harris, K., et al. (2010). Ultrastructural analysis of hippocampal neuropil from the connectomics perspective. *Neuron*, 67, 1009–1020.
- O'Rourke, J. (1994). *Computational geometry in C*. Cambridge: Cambridge University.
- Ramírez, O.A., & Couve, A. (2011). The endoplasmic reticulum and protein trafficking in dendrites and axons. *Trends in Cell Biology*, 21, 219–227.
- Saalfeld, S., Fetter, R., Cardona, A., Tomancak, P. (2012). Elastic volume reconstruction from series of ultra-thin microscopy sections. *Nature Methods*, 9, 717–720.
- Schmid, B., Schindelin, J., Cardona, A., Longair, M., Heisenberg, M. (2010). A high-level 3d visualization api for java and imagej. *BMC Bioinformatics*, 11, 274.
- Shewchuk, J. (2002). *What is a good linear finite element? Interpolation, conditioning, anisotropy, and quality measures* (preprint). University of California at Berkeley.
- Sommer, C., Straehle, C., Koethe, U., Hamprecht, F.A. (2011). ilastik: interactive learning and segmentation toolkit. In *8th IEEE international symposium on biomedical imaging (ISBI 2011)* (pp. 230–233).
- Sorra, K.E., & Harris, K.M. (2000). Overview on the structure, composition, function, development, and plasticity of hippocampal dendritic spines. *Hippocampus*, 10, 501–511.
- Stiles, J.R., Bartol, T.M., et al. (2001). Monte carlo methods for simulating realistic synaptic microphysiology using mcell. In: *Computational neuroscience: realistic modeling for experimentalists*, (pp. 87–128). Boca Raton: CRC.
- Turk, G., & O'Brien, J. (1999). Shape transformation using variational implicit functions. In *SIGGRAPH'99* (pp. 335–342).
- van Spronsen, M., & Hoogenraad, C.C. (2010). Synapse pathology in psychiatric and neurologic disease. *Current Neurology and Neuroscience Reports*, 10, 207–214.
- Zhang, Y., & Bajaj, C. (2006). Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. *Computer Methods in Applied Mechanics and Engineering*, 195, 942–960.
- Zhang, X., Bajaj, C.L., Kwon, B., Dolinsky, T.J., Nielsen, J.E., et al. (2006). Application of new multiresolution methods for the comparison of biomolecular electrostatic properties in the absence of global structural similarity. *Multiscale Modeling & Simulation*, 5, 1196–1213.
- Zhang, Y., Bajaj, C., Sohn, B. (2005). 3D finite element meshing from imaging data. *The Special Issue on Unstructured Mesh Generation, Computer Methods in Applied Mechanics and Engineering (CMAME)*, 194, 5083–5106.
- Zhang, Y., Bajaj, C., Xu, G. (2009). Surface smoothing and quality improvement of quadrilateral/hexahedral meshes with geometric flow. *Communications in Numerical Methods in Engineering*, 25, 1–18.
- Zhang, Q., Bettadapura, R., Bajaj, C. (2012). Macromolecular structure modeling from 3D EM using VolRover 2.0. *Biopolymers*, 97, 709–731.
- Zhang, Y., Hughes, T.J., Bajaj, C.L. (2010). An automatic 3d mesh generation method for domains with multiple materials. *Computer Methods in Applied Mechanics and Engineering (CMAME)*, 199, 405–415.