Extensible visualization and analysis for multidimensional images using Vaa3D

Hanchuan Peng[1,2], Alessandro Bria[3,4], Zhi Zhou[1], Giulio Iannello[3] & Fuhui Long[2,5]

[1]Allen Institute for Brain Sciences, Seattle, Washington, USA. [2]Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, Virginia, USA. [3]Integrated Research Centre, University Campus Bio-Medico of Rome, Rome, Italy. [4]Department of Electrical and Information Engineering, University of Cassino and Southern Lazio, Cassino, Italy. [5]BioImage, LLC, Bellevue, Washington, USA. Correspondence should be addressed to H.P. (hanchuanp@alleninstitute.org).

Published online 2 January 2014; doi:10.1038/nprot.2014.011

Open-Source 3D Visualization-Assisted Analysis (Vaa3D) is a software platform for the visualization and analysis of large-scale multidimensional images. In this protocol we describe how to use several popular features of Vaa3D, including (i) multidimensional image visualization, (ii) 3D image object generation and quantitative measurement, (iii) 3D image comparison, fusion and management, (iv) visualization of heterogeneous images and respective surface objects and (v) extension of Vaa3D functions using its plug-in interface. We also briefly demonstrate how to integrate these functions for complicated applications of microscopic image visualization and quantitative analysis using three exemplar pipelines, including an automated pipeline for image filtering, segmentation and surface generation; an automated pipeline for 3D image stitching; and an automated pipeline for neuron morphology reconstruction, quantification and comparison. Once a user is familiar with Vaa3D, visualization usually runs in real time and analysis takes less than a few minutes for a simple data set.

INTRODUCTION

Multidimensional microscopic images are indispensable data resources in current biology[1]. A number of bioimage-oriented software packages, such as the free tools Voxx[2], OME[3], ImageJ[4], Icy[5], ilastik[6], CellProfiler[7], CellOrganizer[8], CellExplorer[9], FARSIGHT[10], Bisque[11], BrainExplorer[12] and BrainAligner[13], as well as many commercial software suites, such as Amira (VSG), Imaris (Bitplane), ImagePro (MediaCybernetics) and Neurolucida (MBF Bioscience), have been developed to tackle a variety of visualization and analysis problems for multidimensional images. Reviews of recent software packages in this category, especially the open-source tools, as well as their case studies, can be found in many papers[14–18].

For a 3D or multidimensional image data set, obviously, one may choose to visualize and explore it using a 2D cross-sectional display, one section at a time. This is compatible with the current planar computer-display techniques. It is also used in many tools, such as ImageJ[4], to view many biological and biomedical 3D images in the commonly defined *XYZ* spatial coordinate system. Unfortunately, the use of only 2D cross-sectional views, or their variants such as a tri-view display that simultaneously shows *XY*, *YZ* and *ZX* cross-sectional views, or a combined display with an arbitrary-angle cutting plane[1], is often insufficient to observe complex 3D structures and the relationship among multiple objects (e.g., cells) in a 3D or higher-dimensional image. Hence, it is desirable to have efficient tools to explore faithfully the 3D and multidimensional image data and the correspondingly 'reconstructed' objects in these image data.

Although some conventional 2D visualization–based image analysis packages, such as ImageJ and its newer versions (e.g., Fiji[19]), now incorporate 3D image visualization modules, a number of sophisticated 3D visualization tools have already been developed in the last two decades. The commercial packages such as Amira (VSG) and Imaris (Bitplane) are able to render a 3D image and at the same time analyze the image content. Although these commercial tools are versatile, their use is sometimes limited owing to high expense of licenses, instability when visualizing large image volumes, difficulty to learn and use, and so on. Voxx[2] is a free but closed-source tool that is able to render 3D images and image sequences. However, it does not provide functions to analyze images or display surface objects overlaid on the rendered voxel data. The Open Source Visualization Toolkit (VTK, Kitware) is a toolkit developed mainly for rendering surface meshes but not multidimensional images.

Vaa3D system

The open-source Vaa3D system[20] was specifically designed for bridging such gaps, particularly for the tasks of visualization, interaction, analysis and management of large, multidimensional (including 3D, 4D and 5D and so on) microscopic images. To cope with the need for extension in specific applications, Vaa3D can be enhanced by flexible plug-in programs and working modes. This enables Vaa3D to be used in new domains such as automated analysis of very complex images, microscopic image acquisition, microsurgery of live animals and cell cultures and virtual reality.

The overall design principle of Vaa3D is to provide a simple and cross-platform (Mac, Linux and Windows) graphical user interface (GUI) and at the same time to enhance a user's ability to visualize and quantitatively analyze the large 3D and high-dimensional image stacks used in biology and medical studies. As the analysis of image content is a major component of the system, Vaa3D is able to generate, visualize and quantitatively analyze various surface representations of bioimage objects, such as irregularly shaped meshes for brain compartments, digitally reconstructed neurons and their networks, landmarks of salient features in cell populations, point clouds for the distribution of cell locations and interaction relationships, user-defined lines and curves in 3D space for measuring biological entities, to name a few.

As a software platform for large-scale bioimage visualization and analysis, Vaa3D was specially developed not to replicate or emphasize existing features of previous tools (e.g., ImageJ and Amira) (**Table 1**). For instance, although Vaa3D has the ability to

TABLE 1 | Comparison of extensible software platforms of bioimage visualization and analysis.

Platform software	Cost	Visualization type	3D visualization speed	3D image processing and analysis	3D annotation of image/ surface objects	Direct 3D content exploration	Very large (terabytes) image data and database access
Vaa3D	Open source, free	Built-in visualization for 3D/4D/5D images and heterogeneous 3D surface objects	Fast	Yes	Built-in	Yes (3D pinpointing, 3D curve drawing, 3D region-of-interest generation and other direct 3D exploration techniques)	Yes (e.g., Vaa3D-Atlas Manager for thousands of 3D GAL4 imageries; Vaa3D-TeraFly for whole-brain light-sheet microscopy data sets; Janelia FlyWorkstation)
ImageJ/Fiji	Open source, free	Built-in 2D display; needs an additional plug-in for 3D image visualization; no strong surface visualization	Moderate	Yes	Need plug-ins	No	Yes (e.g., TrakEM for electron microscopy and array tomography data sets)
ITK	Open source, free	NA	NA	Yes	No	No	NA
VTK	Open source, free	Mostly surface rendering, with some 3D image visualization capability	Fast	NA	NA	No	NA
Amira	Commercially available	Multidimensional image and surface visualization	Fast	Yes	NA	NA	NA
Imaris	Commercially available	Multidimensional image and surface visualization	Fast	Yes	Yes for simple surface primitives	Simple 3D curve drawing	NA
MATLAB	Commercially available	Simple 3D surface rendering	Slow	Yes	NA	No	NA

All these platforms can run on multiple operating systems. 'NA' means either a system is not designed for the specific purpose, or well-known examples of this system for a specific purpose were not available when this article was written.

display 2D sections of an image stack nicely, Vaa3D is particularly good at rendering multidimensional (3D, 4D and 5D) image data efficiently and precisely. It also generates multidimensional surface objects that can be overlaid and blended with the volumetric rendering easily. These features, in combination with many others (**Table 1**) featured in this article and also in Vaa3D's resource websites (**Table 2**), make this system unique in offering comprehensive and quantitative solutions to bioimage-related applications. In addition, Vaa3D can be combined with valuable features of other tools, such as conversion of movie formats and specialized cell segmentation and tracking plug-ins using ImageJ, conversion of image file formats using Bioformats (http://loci.wisc.edu/software/bio-formats) and easy scripting using MATLAB.

Examples of Vaa3D functions

Although Vaa3D has been frequently used to tackle challenging neuroscience applications[13,21], Vaa3D itself is a general tool suitable for a wide range of bioimage informatics applications, including the interactive exploration of images with PCs and the pipelined and scaled-up analysis with supercomputers. As Vaa3D is versatile and has a few hundred modules, it is impractical to describe all Vaa3D functions within this article. Therefore, the central goal here is to provide a guided tour of several frequently used functions of Vaa3D. We demonstrate Vaa3D's capability for visualizing and analyzing multidimensional images, as well as the surface objects extracted from these images (Steps 1A–1F). We also show briefly how to extend Vaa3D by using its plug-in

TABLE 2 | Vaa3D resource links.

Resource	Website
Main website	http://vaa3d.org/
Documentation	http://code.google.com/p/vaa3d/
Help/discussion forum	http://www.nitrc.org/forum/forum.php?forum_id=1553
Bug tracking and request new features	http://www.nitrc.org/tracker/?group_id=379

interface (Step 1G) plus three example pipelines (Steps 1H–1J) of more complicated tasks.

We demonstrate three example pipelines (Steps 1H–1J), including automated filtering, segmentation and surface generation for 3D cellular images, automated 3D image stitching, and automated reconstruction and analysis of 3D neuron morphology. These pipelines consist of one or more Vaa3D plug-in extensions and built-in modules. They demonstrate how to combine various Vaa3D's features for a range of applications.

Image analysis for cell biology. In Step 1H, we consider a typical use case of image analysis for cell biology (i.e., quantitative profiling of cell properties). We show how to automatically segment fluorescently labeled cells from a 3D image stack, count the total number of cells and measure the surface (e.g., cell membrane) area of a cell. This is done by first filtering out the noise in the 3D cellular image, followed by adaptive thresholding and watershed segmentation to identify all image objects that correspond to cells. Thereafter, we generate surface meshes for all the segmented cells and measure their total number and the surface areas of individual cells.

Stitching of terabytes of image tiles. In Step 1I, we demonstrate an increasingly encountered use case of *in toto* microscopic imaging studies, in which an enormous number of images of a large specimen are acquired in tiled scan. The image tiles need to be stitched to form a panoramic view. Although Vaa3D has a default 3D image-stitching plug-in[22] that has been successfully used for this problem, here we describe TeraStitcher[23], which is another Vaa3D plug-in that is specifically designed to stitch terabytes of volumetric images of an entire brain or similar massive image data sets. TeraStitcher has been used in managing images with a high number of slices, i.e., with large extension along the *Z* dimension, as those produced by light sheet microscopy techniques[24]. In this use case, the tiles composing the volumetric image are partially overlapped, arranged according to a bi-dimensional matrix along the *XY* dimensions and their relative nominal positions. TeraStitcher is able to stitch teravoxel-sized images even on computers with relatively limited memory and processing resources.

Reconstructing, quantifying and comparing neurons in 3D. In Step 1J, we show how to reconstruct and quantify the 3D morphology of a neuron, where the neuron is fluorescently labeled and the image is acquired using a laser scanning microscope. Reconstruction of 3D neuron morphology is a marked yet very challenging bioimage analysis problem, which is of remarkable importance to the reverse engineering of a brain's structure and functions. Here we describe how to use a Vaa3D plug-in implementing the all-path-pruning method[25] to automatically reconstruct the 3D morphology of a neuron. This is followed by the use of another Vaa3D plug-in that measures morphological features of a neuron. We also show how to calculate the differences between multiple neuron reconstructions. This pipeline is applicable for neuroscience studies in which the neuron morphology needs to be quantified, as well as for similar biology applications in which tree- or fiber-like structures (e.g., blood vessels, bronchial tree and microtubules) need to be extracted and quantified.

MATERIALS

EQUIPMENT

Suitable data set

- Data should be image files (e.g., in TIFF format), surface files (e.g., in SWC format) or files linked together with Vaa3D's linker file format if multiple files are required to be opened together. To run these demonstrations, download the test data from http://www.vaa3d.org/ under the 'Test Data' tab

▲ CRITICAL Many times the test data may be replaced by your own data sets of similar nature. See the TROUBLESHOOTING section.

EQUIPMENT SETUP

Computer In practice, any Mac, Linux or Windows computer running OpenGL 2.1 and above should be able to run the full set of features of Vaa3D. Steps 1A–1F have been tested on: an Intel Core i7 2.7 GHz with 4 GB of random

Box 1 | Setting up Vaa3D ● TIMING 5–20 min

1. Go to http://vaa3d.org/.
2. Click on the 'Download' tab.
3. From the 'Current Release' section, choose the appropriate program corresponding to your operating system (Mac, Linux or Windows) and download it to your local computer. Unzip and install the program.
▲ CRITICAL STEP The Vaa3D program is often compressed as a ZIP file and you need to unzip it before use. For Linux and Windows, unzip the program into a new folder before use. For Mac, unzip the program to generate an installer program for standardized installation. Double-click the installer program to launch a GUI of installation, and enter the user password to complete the installation, which will put the Vaa3D program under the /Applications/vaa3d folder on the local computer.
4. Run the Vaa3D program to launch the GUI. For Windows, double-click the program 'vaa3d_msvc.exe'. For Linux, on a command-line console, enter the folder that contains the unzipped program and type './start_vaa3d.sh' command to run. For Mac, double-click the program '/Applications/vaa3d/vaa3d64.app' (without displaying the log information) or on the command line console run the command '/Applications/vaa3d/vaa3d64.app/Contents/MacOS/vaa3d64' to start the GUI with the running log information displayed.
? TROUBLESHOOTING

access memory (RAM) and NVIDIA GeForce GT 650M running Mac OS X 10.7 64 bit; an Intel Xeon 3.40 GHz with 8 GB RAM running Linux Redhat 5.3 64bit and NVIDIA GTX 280; and an Intel Xeon 3.33 GHz with 3 GB RAM running Windows 7 Professional 64bit. Steps 1G–1J have also been tested on an Intel Core i7 2.7 GHz with 4 GB of RAM and NVIDIA GeForce GT 650M running Mac OS X 10.7 64 bit; an Intel Xeon 3.40 GHz with 8 GB of RAM running Linux Redhat 5.3 64 bit and NVIDIA GTX 280; these options are different on the Windows system, and thus interested readers should check the Vaa3D documentation website and additional resources (**Table 2**) for further information.

Vaa3D download Vaa3D can be downloaded from http://vaa3d.org/ and run locally as a stand-alone application on Mac, Linux and Windows machines (**Box 1**). Otherwise, Vaa3D can be built from source code and run by going to http://code.google.com/p/vaa3d/wiki/BuildVaa3D and following the appropriate guide corresponding to your operating system (Mac, Linux or Windows).

PROCEDURE

Step-by-step case examples

1| On the basis of the information described in the INTRODUCTION and in the table below, choose from options A–J:

Option	Function covered
A	Visualize a 3D multicolor image
B	Visualize a 5D image data set such as a series of multicolor 3-D images
C	Generate 3D markers and measure image content quantitatively using landmarks
D	Generate, quantify, measure and annotate 3D surface meshes
E	Compare, fuse and manage many 3D images
F	Visualize heterogeneous images and respective surface objects
G	Extend Vaa3D functions with its plug-in interface
H	Profile cells quantitatively in 3D on the basis of automated image filtering, segmentation and mesh generation
I	Stitch terabytes of 3D images automatically
J	Reconstruct and quantify 3D neuron morphology automatically

(A) Visualize a 3D multicolor image ● TIMING 2–5 min

(i) From the Vaa3D Test Data web page (http://www.vaa3d.org/), click on '3D image stack of late embryonic fruit fly nervous system' and download the respective image data in TIFF format.

(ii) Open the image stack by either dragging and dropping the file into the main window of Vaa3D or clicking the 'File' main menu, followed by clicking the 'Open image/stack in a new window' menu. This will open a tri-view display for the image stack in **Figure 1a**.

? TROUBLESHOOTING

(iii) Use various control buttons, scroll bars, check boxes and other tools on the right side of the tri-view window to see various regions of an image in different display modes; for example, you can use different color blending method, or use dual-looking glasses to see locally amplified image regions (**Fig. 1b**).

(iv) Click the 'See in 3D' button at the bottom right of the tri-view window. Then choose either a global 3D viewer window (**Fig. 1c**) or a local 3D viewer window (**Fig. 1d**).

▲ CRITICAL STEP The local 3D viewer window is produced either on the basis of the currently defined region of interest (ROI) in a tri-view or adaptively defined ROI from a 3D viewer window; see Step 1A(viii–x) for more information.

? TROUBLESHOOTING

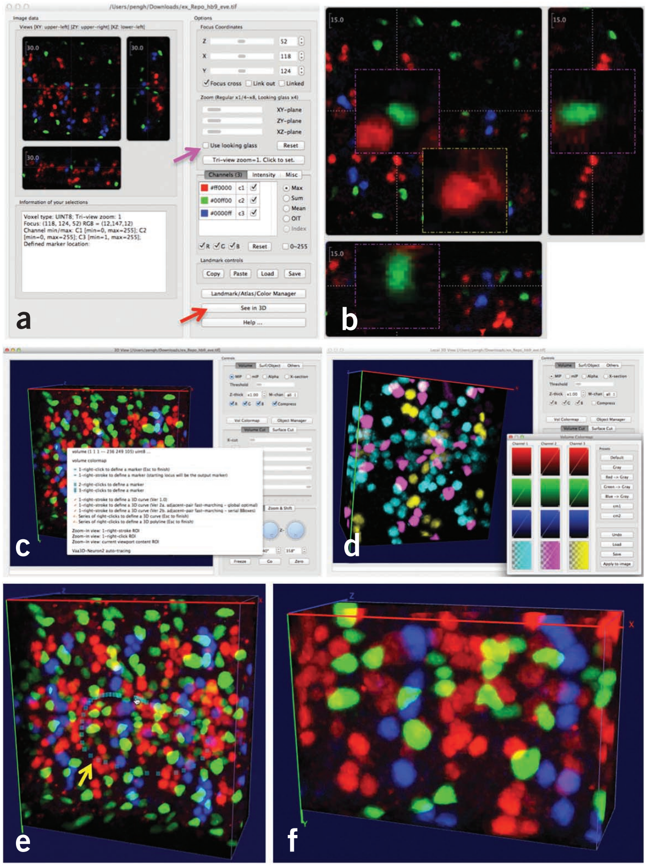
(v) In a 3D viewer window, press and hold the left mouse button to rotate the 3D-rendered image freely.

(vi) In a 3D viewer window, use the mouse scroll wheel to zoom in or zoom out on the 3D-rendered image freely.

(vii) In a 3D viewer window, hold the 'Shift' key, and then press and hold the left mouse button to shift around the 3D-rendered image freely.

(viii) In a 3D viewer window, use various control buttons on the right side to visualize a subportion of the image, map color channels in a customized way or switch between maximum-intensity projection, alpha-blended rendering and several other rendering modes (**Figs. 1c** and **2**).

(ix) In a 3D viewer window, move the mouse to the area where the image is displayed and right-click to activate a pop-up menu that contains a series of operations defined for the image content (**Fig. 1c**).



Information of your selections

Voxel type: UINT8; Tri-view zoom: 1
Focus: (118, 124, 52) RGB = (12,147,12)
Channel min/max: C1 [min=0, max=255]; C2 [min=0, max=255]; C3 [min=1, max=255];
Defined marker location:

Figure 1 | Screenshots of visualization of a single 3D fluorescently labeled cellular image stack of the late-embryonic-stage nervous system of *Drosophila*. (**a**) The tri-view of the image along with the right-side control pane. Magenta arrow, the looking-glass checkbox. Red arrow, the 'See in 3D' button. (**b**) The dual looking-glass display mode of the tri-view, in which two local regions (the one with the purple border is for baseline comparison and the one with the yellow border is for a moving ROI observation) are displayed in locally zoomed-in windows. (**c**) The global 3D viewer of the image, along with the right-side control pane and an image content–specific pop-up menu. (**d**) A local 3D viewer of the image, along with a color-map tool that can be used to adjust how each data channel of the image is visualized using different color-mapping schemes. The color-map tool can be activated by pressing the 'Vol Colormap' button on the right-side control pane of the 3D viewer. (**e**) Defining a 3D ROI using one mouse stroke. Yellow arrow and dotted cyan outline show the ROI. (**f**) The local 3D viewer of the ROI defined in **e**.

(x) Select the menu item 'Zoom-in view: 1-right-stroke ROI' or 'Zoom-in view: 1-right-click ROI' to define a ROI in 3D space directly using one mouse stroke (**Fig. 1e**) or click, and thus open a new local 3D viewer window for that ROI (**Fig. 1d**,**f**).
▲ CRITICAL STEP The local 3D viewer and the 'global'-D viewer can have different colored maps so as to facilitate comprehensive simultaneous visualization of different image regions in different ways.

(xi) If desired, generate the local 3D viewer of a different ROI from either the initial global 3D viewer or an existing local 3D viewer, by repeating Step 1A(ix,x).

(xii) If desired, toggle on and off a 3D animation of the 3D rendering by pressing 'Ctrl/Cmd+A'. This can be done in any 3D viewer.

(B) Visualize a 5D image data set such as a series of multicolor 3D images ● TIMING 2–5 min

(i) From the Vaa3D Test Data web page, click on '5D (x-y-z-c-t) image series data visualization' and download the respective zipped image data set; unzip all these files into one file folder.

(ii) Click the main window menu of Vaa3D and then click the 'File' main menu; follow this by clicking the 'Import' menu.
▲ CRITICAL STEP It is important to use the 'Import' function rather than the normal file-opening method that is designed to open one single file.

(iii) Click the 'Import general image series to an image stack' menu to open a tool to import files.

(iv) Enter the folder in which the time series of images or image stacks has been stored.

(v) Select any image in that folder; then click 'Open' and an 'Image Sequence' dialog will be launched.

(vi) From the 'Image Sequence' dialog, select the option 'Pack images in 'Channel' dimension' and click 'OK' to load the entire time series to Vaa3D and open a tri-view window for the time-series data.
▲ CRITICAL STEP 'Pack images in 'Channel' dimension' is the second option on the drop-down list when it is clicked.
? TROUBLESHOOTING

(vii) Once the tri-view window is open, use the methods in Step 1A(iv) to open a 3D viewer and see the data in 3D (**Fig. 3**).

(viii) In the 3D viewer, for time-series data, there will be an additional scrollbar for individual time points just underneath the 3D displayed images. Drag this indicator back and forth to navigate data that correspond to different time points (**Fig. 3**).

(ix) In the 3D viewer, if desired, follow Step 1A(ix–xii) to generate and visualize local 3D views of the time series (**Fig. 3**).
▲ CRITICAL STEP This is a useful way to visualize the full resolution data of an image region data rapidly, even across a number of time points.

(C) Generate 3D markers and measure image content quantitatively using landmarks ● TIMING ~3–10 min

(i) In a tri-view window of an image, right-click to change the 3D focus location of the current observation, indicated by the cross-line locations in the *XY*, *YZ* and *ZX* cross-sectional planes.

(ii) Press the 'm' key to activate a marker-property dialog, and thus to specify a 3D marker at the focus location.

(iii) Assign a name and a comment to the marker using the marker-property dialog if desired.

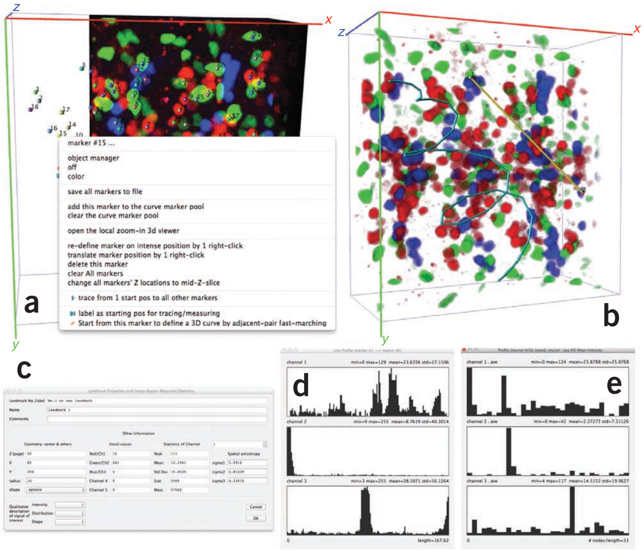

a
b
c
d
e
x
y
z
marker #15 ...
object manager
off
color
save all markers to file
add this marker to the curve marker pool
clear the curve marker pool
open the local zoom-in 3d viewer
re-define marker on intense position by 1 right-click
translate marker position by 1 right-click
delete this marker
clear All markers
change all markers' Z locations to mid-Z-slice
trace from 1 start pos to all other markers
label as starting pos for tracing/measuring
Start from this marker to define a 3D curve by adjacent-pair fast-marching
channel 1
channel 2
channel 3

Figure 2 | Screenshots showing the creation of 3D markers, 3D line segments and 3D curves for an image and quantitative profiling of image content based on these objects. (**a**) Continuous definition of a number of 3D markers via direct 3D pinpointing of visible image objects (e.g., cells). A marker-specific pop-up menu is also shown. (**b**) A 3D line segment (yellow) and a 3D curve (cyan) that cross many image sections. (**c**) The marker-property dialog, which presents options for defining the size and type of local observation and for defining a measuring window around each marker, as well as for selecting the data channel to be used for quantitative real-time profiling of the image content. (**d**) Quantitative profiling of image voxel intensity along a 3D line segment in **b**. (**e**) Quantitative profiling of image intensity along a 3D curve in **b**.

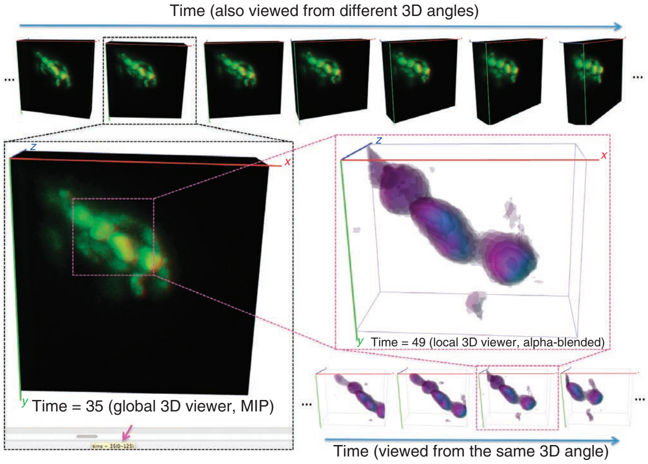

Time (also viewed from different 3D angles)
Time = 35 (global 3D viewer, MIP)
Time = 49 (local 3D viewer, alpha-blended)
Time (viewed from the same 3D angle)
x
y
z
...

Figure 3 | Visualization of a 5D image data set of the *C. elegans* nervous system. In this data set, there are a number of time points, each of which is a multicolor 3D image stack. Top, the time series can be visualized continuously and also in various 3D perspectives. Zoomed-in at left is the global 3D viewer of one particular time point, indicated by the magenta arrow at the bottom of the 3D viewer of this time series; the image is displayed in the MIP mode. Zoomed-in at right is a local 3D viewer of the ROI defined in the global 3D viewer using one mouse stroke; the local image region is displayed with a different color mapping (similar to **Fig. 1d**) and also shown in the alpha-blended rendering mode. Bottom right, the local 3D viewer is also able to display the local image region continuously in time, thus facilitating seamless visualization of a large 5D data set in full spatial and temporal resolution.

(iv) Adjust the radius and shape fields in the marker-property dialog, as well as the value in the 'Statistics of Channel' field, to calculate the basic statistics of a local 3D region centered at the marker location. These statistics include peak intensity, mean intensity, s.d., size, mass and local anisotropy features (**Fig. 2c**).

(v) In a 3D image viewer, move the mouse to the area where the image is displayed, right-click to activate a pop-up menu that contains a series of operations specifically defined for the image content, and then select '1-right-click to define a marker (Esc to finish)'.

(vi) Right-click continuously to define one or more markers in the 3D space, directly at the locations of any visible 3D image objects (**Fig. 2a**).

▲ CRITICAL STEP The cursor needs to be moved to the location of an observable image object before the right mouse button is clicked. Press the 'Esc' key on the keyboard to finish.

? TROUBLESHOOTING

(vii) Move the cursor on top of any marker of interest, and then right-click to activate a marker-specific pop-up menu that contains a series of marker-related functions (**Fig. 2a**).

(viii) Click the first menu item on the marker-specific pop-up menu to access the marker property dialog mentioned in Step 1C(iv) without going back to the tri-view window.

(ix) When at least two markers have been defined, re-do Step 1C(vii) for one of the markers, and then click the menu item 'label as starting pos for tracing/measuring'.

(x) Move to the second marker of interest, re-do Step 1C(vii) and then click the menu item 'line profile from the starting pos'. This operation will result in a straight line being drawn and displayed on top of the 3D display image (**Fig. 2b**), as well as in a line profile window that shows the quantitative intensity profiles of every data channel of the image along the 3D straight line segment defined by the two specified markers (landmarks) (**Fig. 2d**).

(xi) In a 3D viewer, move the mouse to the area where the image is displayed and right-click to activate the image-specific pop-up menu, and then select the 'Series of right-clicks to define a 3D curve (Esc to finish)' menu item.

(xii) Right-click repeatedly to define a series of 3D anchor landmarks, and then press the 'Esc' key to finish. This will generate a 3D curve that is overlaid on top of the image (**Fig. 2b**).

? TROUBLESHOOTING

(xiii) Move the mouse cursor on top of the 3D curve and right-click to activate a curve-specific pop-up menu.

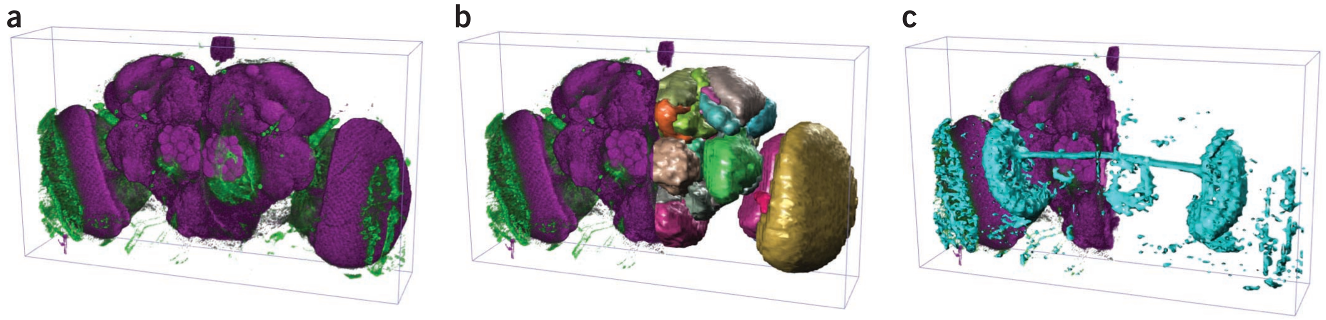


Figure 4 | Screenshots showing the creation and visualization of irregular 3D surface meshes for image content of different data channels. (**a**) A 3D viewer display of an ato-GAL4–labeled image of an adult *Drosophila* brain. Purple, nc82 labeling of a neuropil. Green, the ato-GAL4–labeled neuron population. (**b**) Generation and visualization of different brain compartments based on a pre-generated label field file from the neuropil data channel. The hybrid view shows half of the brain in image voxels and half of the brain with surface meshes of many brain compartments (indicated by different colors). This hybrid view is generated via combining the 'Volume Cut' and 'Surface Cut' functions provided on the right-side control pane of a 3D viewer. The surface mesh is generated using the 'Label field surface' option of the surface-creating tool. (**c**) Generation and visualization of the neuron population in the ato-GAL4 data channel. The surface mesh (cyan) is generated using the 'Range surface' option in the surface-creating tool. The surface mesh is a digital model of the voxel data of the neuron population displayed in green color in **a** and thus facilitates quantitative profiling of the respective image area.

(xiv) Click the menu item 'neuron-segment intensity profile' to display windows showing the image voxel mean intensity and total-mass profiles along the 3D curve (**Fig. 2e**).

(D) Generate, quantify, measure and annotate 3D surface meshes ● TIMING ~3–10 min

(i) From the Vaa3D Test Data web page, click on '3D fruit fly brain compartments and surface-mesh generation example' and download the corresponding zipped image data set; unzip it into a folder that contains two files. Open the file 'fly_brain_ato_pattern.tif' using Vaa3D and then generate the 3D viewer for this file (**Fig. 4a**).

(ii) Drag the file 'fly_brain_label_field.tif' and drop it into the 3D viewer window of the image 'fly_brain_ato_pattern.tif'. This will open a surface-creating tool.

(iii) Select the 'Label field surface' option and click 'OK'.

(iv) Select the 'Marching Cubes' option and click 'OK'.

(v) Select the mash density option to be '100', and click 'OK'.

▲ CRITICAL STEP If this value is changed to a larger value, e.g., 200, the mesh will be generated in a more detailed manner, which will take longer to process.

(vi) Click 'No' for the dialog that asks for importing a label field name file.

(vii) Wait until the surface meshes of different brain compartments have been generated (**Fig. 4b**).

(viii) Move the cursor of the mouse on top of an arbitrary mesh for a compartment, indicated by a different color, and then click the right mouse button. A pop-up menu specific to the surface mesh will appear.

(ix) Click the first menu item in the pop-up menu; a surface annotation dialog will appear. This annotation dialog displays the total area of the current selected surface.

(x) Use the same dialog to annotate this surface item, including the name and any comments. Then click 'OK'.

(xi) Repeat Step 1D(viii) and then for the surface-specific pop-up menu, click the 'save all label-field surface objects to file' and save the generated surface-mesh objects to one file 'fly_brain_label_field.vaa3ds', which will be used in a subsequent example (i.e., Step 1F).

(xii) On the right-side control pane of the 3D viewer, click the 'Surf/Object' tab and then click the 'Load/Save Surf >>' button.

(xiii) Click the 'Create Surface from Associated Image Channel 2' option, which will activate the surface-creating tool similarly to the result of Step 1D(ii). In this demo, the tool will generate the surface object for the ato-GAL4 neuron population in the data.

(xiv) Select 'Range surface' and click 'OK'.

(xv) Enter '80' and '255' to specify the lower and upper bounds of the image voxel intensity range within which a surface mesh will be generated.

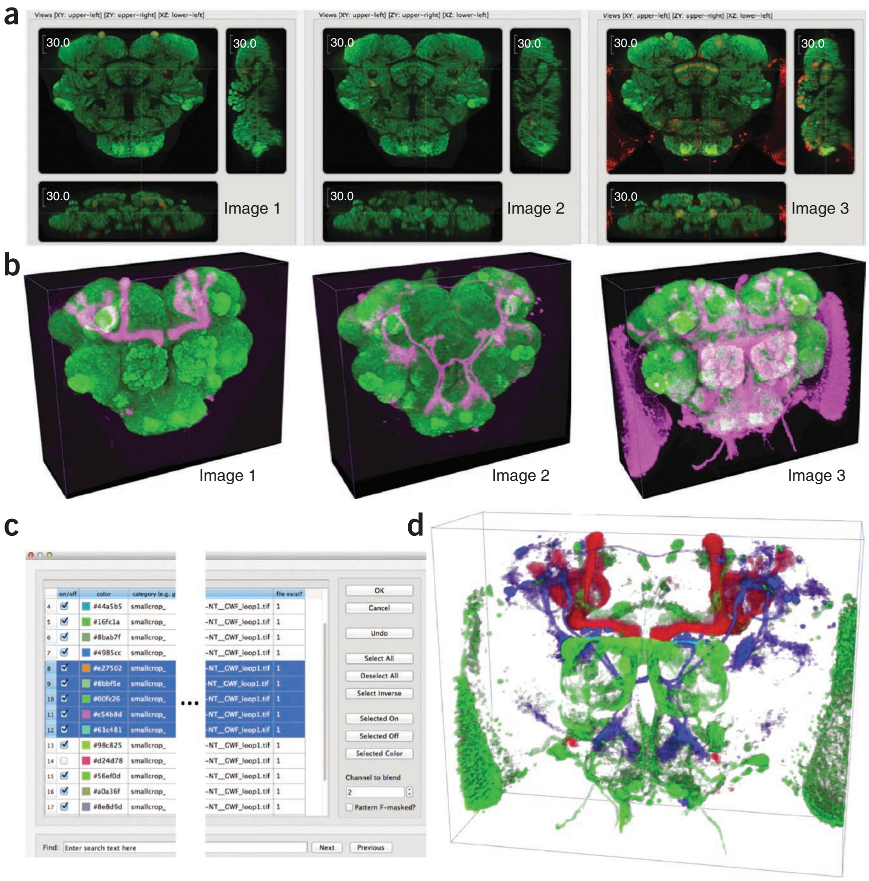
▲ CRITICAL STEP These two values can be changed between 0 and 255; the wider the range, the larger coverage of the surface mesh will have for the image. This also requires a longer time to generate the mesh.

(xvi) Select several options similar to those in Step 1D(iv–vi).

(xvii) Wait until a new surface mesh is generated, which will replace the old mesh generated in Step 1D(vii) (**Fig. 4c**).

(xviii) Move the mouse cursor on top of the surface mesh, right-click to pop up a surface mesh–specific menu and choose the first menu item. The same annotation dialog of Step 1D(ix) will appear, but this time it will display the total surface area of the image region that contains the neuron population.

Figure 5 | Screenshots of colocalization and fusion of many 3D image stacks. (**a**) Synchronization of tri-views of multiple image stacks of adult *Drosophila* brains that contain different GAL4-labeled neuron populations. Green, neuropil. Red, neuron populations. (**b**) Synchronization of the 3D viewers of images in **a**. Green: neuropil. Purple: neuron populations. (**c**) The Atlas Manager tool can also load, display and assign various color-mapping schemes to different images at the same time. (**d**) The 3D colocalized, fused and visualized neuron populations of **a** and **b**, using the Atlas Manager in **c**. Different colors indicate different neuron populations.



	on/off	color	category (e.g. g	…	file exist?
4	☑	■ #44a5b5	smallcrop_	-NT__CWF_loop1.tif	1
5	☑	■ #16fc1a	smallcrop_	-NT__CWF_loop1.tif	1
6	☑	■ #8bab7f	smallcrop_	-NT__CWF_loop1.tif	1
7	☑	■ #4985cc	smallcrop_	-NT__CWF_loop1.tif	1
8	☑	■ #e27502	smallcrop_	-NT__CWF_loop1.tif	1
9	☑	■ #8bbf5e	smallcrop_	-NT__CWF_loop1.tif	1
10	☑	■ #00fc26	smallcrop_	-NT__CWF_loop1.tif	1
11	☑	■ #c54b8d	smallcrop_	-NT__CWF_loop1.tif	1
12	☑	■ #61c481	smallcrop_	-NT__CWF_loop1.tif	1
13	☑	■ #98c825	smallcrop_	-NT__CWF_loop1.tif	1
14	□	■ #d24d78	smallcrop_	-NT__CWF_loop1.tif	1
15	☑	■ #56ef0d	smallcrop_	-NT__CWF_loop1.tif	1
16	☑	■ #a0a36f	smallcrop_	-NT__CWF_loop1.tif	1
17	☑	■ #8e8d9d	smallcrop_	-NT__CWF_loop1.tif	1

(E) Compare, fuse and manage many 3D images ● TIMING ~3–10 min

(i) From the Vaa3D Test Data web page, click on 'Image fusion and atlas viewer example: 20 fly brains' and download the respective zipped image data set; unzip all files to a file folder. Each of the images contains an nc82 neuropil label and a GAL4-labeled neuron population. These images have been pre-registered using BrainAligner[13].

(ii) Open any two or more images in Vaa3D; for each image, one tri-view window is launched.

(iii) Select a tri-view window arbitrarily and click the 'Link out' checkbox on the right side of the control pane.

(iv) For each of the remaining tri-view windows, click the 'Linked' checkbox on the right side of the control pane.

(v) Go back to the first tri-view window and adjust the focus point of the observation position; all the 'linked' tri-views will be synchronized. In this way, the image contents at the same location across different images can be compared in the tri-view display mode (**Fig. 5a**).

(vi) Open the 3D viewer of one arbitrarily selected image (for simplicity, call it Image_T).

(vii) Press and hold the left mouse button in the 3D viewer of Image_T; move around the mouse and thus rotate the 3D displayed image.

(viii) Open the 3D viewer of another arbitrarily selected image (for simplicity, call it Image_S).

(ix) In Image_T's 3D viewer, click the 'Freeze' button on the right side of the control pane. The rotation angles will be normalized.

(x) In Image_S's 3D viewer, click the 'Zero' button on the right side of the control pane.

(xi) In Image_S's 3D viewer, enter the *X*, *Y* and *Z* rotation angle values such that they are exactly the same as those in Image_T's 3D viewer; this will synchronize the observation angle of Image_S to be exactly the same as that of Image_T, so that these two images can be compared directly in 3D-rendered scenes (**Fig. 5b**).

▲ CRITICAL STEP These three values should always be input in this order: first *X*, then *Y* and then *Z*.

(xii) In the main window menu of Vaa3D, click 'Advanced', then '3D image atlas' and then 'Build an atlas linker file for [registered] images under a folder'.

(xiii) Click 'OK' for the instruction dialog.

(xiv) In the file dialog, select any image file in the folder created in Step 1E(i). Click 'Open' and then click 'Ok' for the file name–filtering dialog.

(xv) This will create a linker file with the .atlas extension in its file name. Click 'Save' to save this linker file. An information box will display that this file has been saved; click 'OK' to close it.

(xvi) Drag and drop this linker file into the main window of Vaa3D. Inside this linker file, the first image will be opened in a tri-view window.

(xvii) Press 'Ctrl/Cmd+A' to launch the Atlas Manager tool (**Fig. 5c**).

(xviii) Click ‘Select All’ on the right-side control pane of the Atlas Manager tool.
(xix) Click ‘Select On’ on the right-side control pane of the Atlas Manager tool.
(xx) Click ‘OK’ on the right-side control pane of the Atlas Manager tool. In this case, all the images inside the linker file will be loaded and fused into one single image using the colors specified in the Atlas Manager tool.
▲ CRITICAL STEP The ‘Channel to blend’ option can be changed to fuse image contents of a different color channel.
(xxi) Wait until all images have been fused and displayed in the tri-view window. The fused data that contain multiple neuron populations in this example can also be visualized in a 3D viewer (**Fig. 5d**).
(xxii) Change the colors of different images in the Atlas Manager using various other options provided in the tool and fuse the images again.
(xxiii) Check some images to be ‘Off’ in the Atlas Manager tool and fuse the remaining images again.

(F) Visualize heterogeneous images and respective surface objects ● TIMING 2–10 min

(i) Follow the instructions in Step 1D to download the sample data set, extract the image file ‘fly_brain_ato_pattern.tif’ and then produce the surface-mesh file ‘fly_brain_label_field.vaa3ds’; once these files have been extracted and generated, locate and reuse them in the following steps.
(ii) Open the image ‘fly_brain_ato_pattern.tif’ in a 3D viewer.
(iii) Drag and drop the file ‘fly_brain_label_field.vaa3ds’ into the 3D viewer.
(iv) From the Vaa3D Test Data web page, click on ‘3D reconstructed neuron 1’ and on ‘3D reconstructed neuron 2’ to download two neuron reconstruction surface files. Drag and drop these two files, one at a time, into the 3D viewer.
(v) From the Vaa3D Test Data web page, click on ‘3D point cloud …’ and download the point cloud surface file. Drag and drop this file into the 3D viewer.
(vi) Observe the three different types of surface objects, including irregular surface meshes, point clouds and tubular/networked structures (neurons in this case), all rendered together with the image in 3D.
(vii) In the 3D viewer, use the right-side control pane of ‘Volume Cut’ and ‘Surface Cut’ to test various ways to visualize the relationships of these image volume and surface data (the hybrid display result will be similar to **Fig. 4b**).
(viii) Click the ‘Object Manager’ button on the right side of the control pane of the 3D viewer. Go to any of the individual tabs of ‘Label Surface’, ‘Neuron/line Structure’ and ‘Point Cloud’ to turn some parts of the component surface objects on or off, and then observe how the heterogeneous data have been visualized.
(ix) In the display area of the 3D viewer window, move the mouse cursor on top of any of the image or surface objects. Click the right mouse button, and thus activate a content-specific pop-up menu there.
(x) If a pop-up menu specific to ‘Label Surface’, ‘Neuron/line Structure’ and ‘Point Cloud’ has been activated, click the menu item ‘Lock scene and adjust this object’; a ‘Basic Geometry of Surface Object’ dialog will be displayed.
(xi) Adjust any of the entries in the ‘Basic Geometry of Surface Object’ dialog, and then observe how the surface objects have been visualized.

(G) Extend Vaa3D functions with its plug-in interface ● TIMING ~2 min–2 h

(i) Check the web page ‘http://code.google.com/p/vaa3d/wiki/BuildVaa3D’ to install the necessary software environment and get the Vaa3D source code.
(ii) Open Vaa3D. Go to the ‘Plug-in’ main window menu and click ‘_Vaa3D_plug-in_creator’.
▲ CRITICAL STEP This tool and the subsequent guide work smoothly on Mac and Linux; for a Windows machine, depending on the actual source code in the plug-in, it may need additional tweaks to make the plug-in buildable from the source code.
(iii) Click ‘Create plug-in’.
(iv) In the plug-in–creating tool, enter the file path for the Vaa3D source code and the path where the new plug-in source code will be saved. Set the plug-in name to ‘test1’, and enter menu item names as required. Then click ‘OK’.
(v) Read the two-step instructions in the information message box.
(vi) Follow the instructions to type ‘qmake’ and ‘make’ to build the plug-in.
? TROUBLESHOOTING
(vii) In Vaa3D, go to the ‘Plug-in’ main window menu and click ‘Re-scan all plug-ins’.
(viii) In Vaa3D, go to the ‘Plug-in’ main window menu, and now a plug-in with the name ‘test1’ will appear in the drop-down menu.
(ix) Run the new plug-in.
(x) Edit the source code under the specified folder to include any other code or modules the new plug-in needs to have. Then rebuild and run the plug-in.
? TROUBLESHOOTING

(H) Profile cells quantitatively in 3D on the basis of automated image filtering, segmentation and mesh generation ● TIMING ~5 min–1 h

(i) Follow Step 1A(i,ii) to download and open the file ‘ex_Repo_hb9_eve.tif’, which is a cellular image of the fruit fly embryonic nervous system. In the tri-view, first check off the ‘c2’ and ‘c3’ channels on the right-side control pane, and

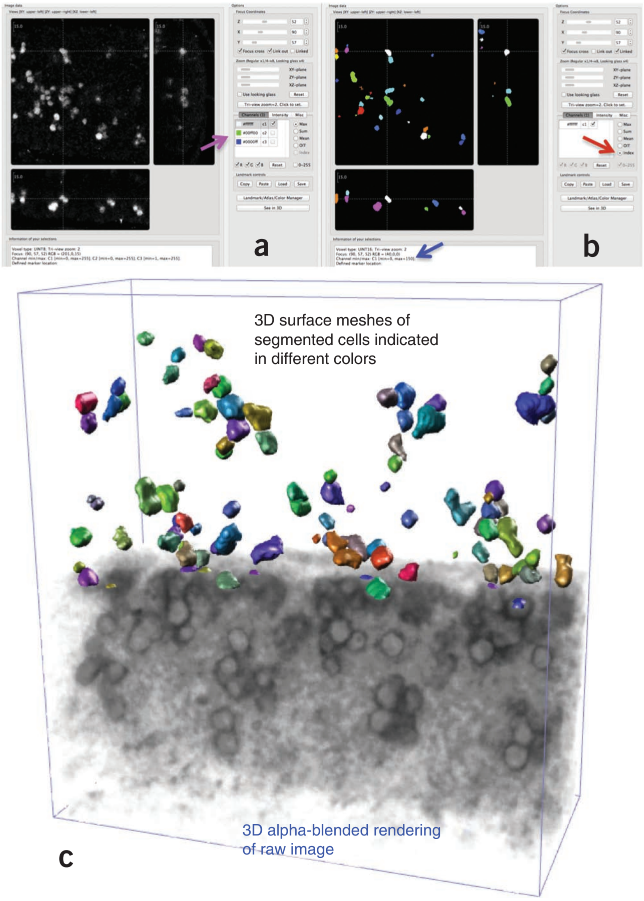


Figure 6 | Screenshots of automated 3D segmentation and quantitative analysis of a 3D cellular image stack. (**a**) The input image stack of a late embryonic stage nervous system of *Drosophila*. The first data channel is displayed in grayscale, whereas the two other data channels are not displayed. Magenta arrow, the checkboxes that control whether or not some data channels are displayed in the tri-view. (**b**) The automated segmentation result in which each segmented cell is labeled in a different color. Red arrow, the 'Index' radio box that allows different image intensity values to be used to indicate different image objects. Blue arrow, the 'max' field in the information box indicates the total number of segmented cells in this case. (**c**) Generation and visualization of the surface mesh representations for the segmented cells. This allows measuring of the surface area of any individual cell. With Vaa3D's functions that combine volume and surface rendering in the same 3D view, it is possible to create a vivid visualization of the cell segmentation process as shown here, to carry out proofreading and editing of the segmentation results, and to perform a quantitative assessment of the distribution and morphologies of cells in the 3D space, all within a few seconds (for this example).

then right-click on the color button of channel 'c1' to activate a pop-up menu and select 'Gray (r=g=b=255)'. This will display only the first channel of this image in grayscale (**Fig. 6a**).

(ii) In Vaa3D, go to the 'Plug-in' main window menu and click 'image_segmentation', then click on 'Cell_Segmentation', and finally click on 'Segment all image objects (adaptive thresholding followed by watershed)'.

(iii) In the parameter-defining dialog, choose the data channel of the image stack. Enter 1 for the following example steps.

(iv) In the parameter-defining dialog, choose the data-smoothing parameters for the median filter and the Gaussian filter. Normally, the default parameters do not need to be changed.

(v) In the parameter-defining dialog, either choose adaptive thresholding or use a global threshold for the entire image or for every *Z*-slice of the image stack. These parameters do not have to be changed from the defaults.

(vi) In the parameter-defining dialog, choose the segmentation method: either the intensity-based watershed algorithm or the shape-based watershed algorithm.

(vii) Click 'OK' when all the parameters have been set.

(viii) Wait ~20–30 s for a new label field image to be generated in the Vaa3D main window. The tri-view window of this new image has the title 'Segmented image'.

(ix) Check the information box right below the cross-sectional display in the tri-view window of the segmented image (**Fig. 6b**); the 'max' value indicates the total number of segmented image objects, which correspond to cells in this example.

(x) Click the 'Index' checkbox on the right-side control pane of the tri-view; each label in the label field is displayed in a different color (**Fig. 6b**).

(xi) Save this segmented image as a new file 'segment_label_field.v3draw'.

(xii) Open a 3D viewer window for the image 'ex_Repo_hb9_eve.tif'.

(xiii) Click the 'Vol Colormap' button on the right-side control pane of the 3D viewer of the image 'ex_Repo_hb9_eve.tif'.

(xiv) Click the 'Red→Gray' button to display only the first color channel in the 3D viewer in grayscale.

▲ CRITICAL STEP 'Red', 'Green' and 'Blue' refer to the first, second and third channels, respectively.

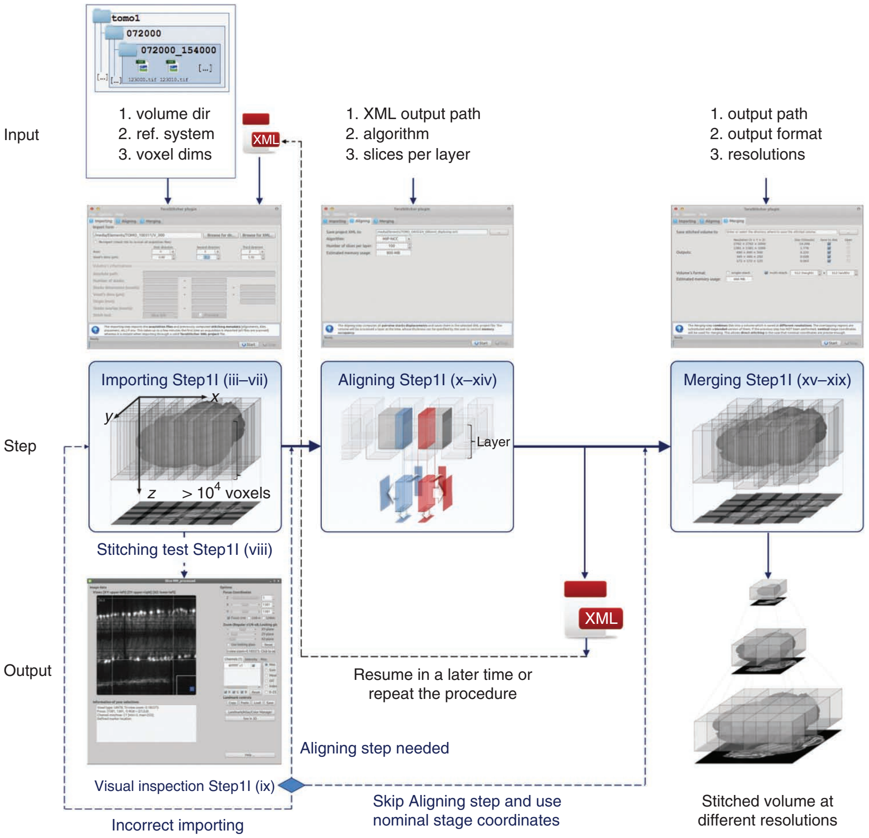
(xv) Drag the saved file 'segment_label_field.v3draw' and drop it into the 3D viewer of the image 'ex_Repo_hb9_eve.tif'. This will invoke the surface mesh–generation tool of Vaa3D.

(xvi) Follow Step 1D(iii–x) to create the surface mesh of the segmented cells, overlay these meshes on top of the grayscale-displayed input image and quantify the surface areas of these objects (**Fig. 6c**).

(I) Stitch terabytes of 3D images automatically ● TIMING ~20 min–2 h

(i) Go to the Vaa3D main window menu 'Plug-in', and click 'image_stitching'→'terastitcher'→'TeraStitcher' to launch the TeraStitcher plug-in. The flowchart of the pipeline is shown in **Figure 7**.

Figure 7 | Pipeline for stitching terabytes of 3D images automatically with the TeraStitcher plug-in. In the central row, the main steps are grouped into three macro steps: 'Importing' Step 1I (iii–vii) for importing the data volume; 'Aligning' Step 1I (x–xiv) for computing tiles' displacements; and 'Merging' Step 1I (xv–xix) for generating the stitched volume and storing it at different resolutions. In the first row, the inputs of the macro steps are listed and the corresponding user interfaces are shown. In the last row, the outputs of the macro steps and other optional steps are depicted with a subfigure.



(ii) Go to https://code.google.com/p/terastitcher/downloads/list to download seven test data sets. Decompress them and put all the files into one folder.
▲ **CRITICAL STEP** It is important to decompress the data to the two-level hierarchical folder structure embedded in the compressed files.

(iii) Select the 'Importing' tab to import the decompressed files.

(iv) Press 'Browse for dir…' and select the root folder of the acquired 3D image that was previously saved in a two-level hierarchical folder structure (for more detailed information about this format, see TeraStitcher online help at https://code.google.com/p/terastitcher/).
? TROUBLESHOOTING

(v) If asked, associate the *x* and *y* axes to the two levels of the folder structure by using the pull-down menus in the 'Axes' row. The *z* axis is already associated with the folder level that contains individual slices. Folders and slices will be scanned using ascending alphanumerical ordering. To use descending alphanumerical ordering along a certain axis, select the axis indicated by the minus sign '–' (e.g., '–*X*' or '–*Y*' or '–*Z*'). For the test data set provided, select '*Y*', '–*X* and '*Z*' for the first, second and third axes, respectively.

(vi) If asked, insert the voxel dimension (in micrometers) along each direction. For the test data set provided, select '0.80', '0.80' and '1.00' for the first, second and third directions, respectively.

(vii) Press the 'Start' button on the bottom bar and wait a few minutes for the tool to scan all the acquisition files. When the scan is finished, the acquisition information is shown in the 'Volume's information' panel.

(viii) In the 'Importing' tab, select the slice to stitch using the spinner in the 'Stitch test' row, and then press the nearby 'Preview' button and wait for the tool to stitch the selected slice using nominal stage coordinates. When stitching is finished, the slice is shown in Vaa3D.

(ix) Inspect the stitched slice along the stack borders highlighted with white.
▲ **CRITICAL STEP** If stack adjacency is not correct, then go to Step 1I(iii), check the 'Re-import' checkbox and repeat Step 1I(iv–viii) inserting a different reference system at Step 1I(v). If misalignments are negligible, go to Step 1I(xv). If you need to align the stacks, go to Step 1I(x).

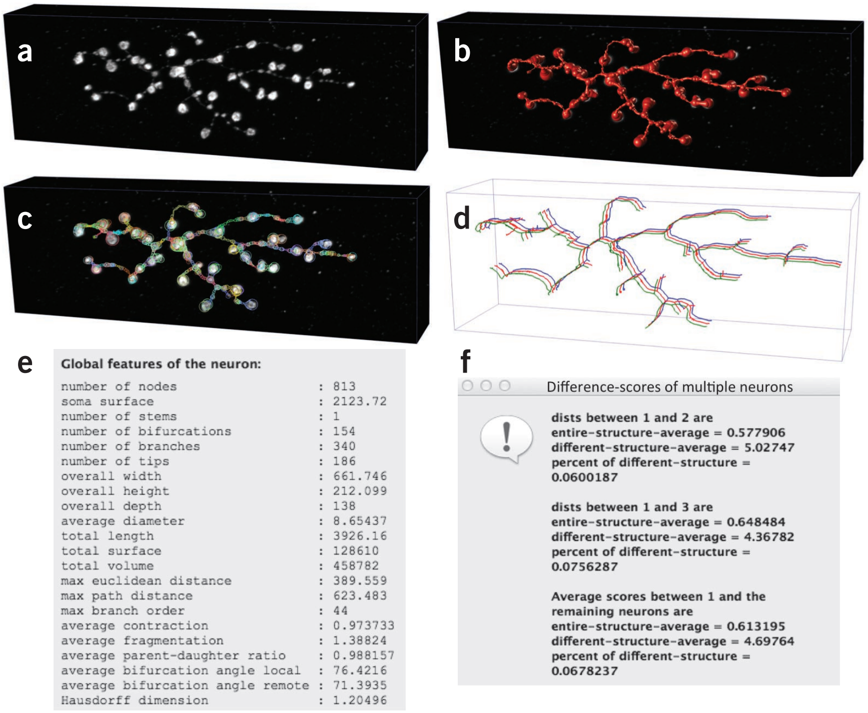
(x) Select the 'Aligning' tab.

(xi) Insert the path where the output XML file is to be saved. By default, an XML named 'xml_displcomp' is saved in the root folder of the acquisition.

(xii) Select the 'MIP-NCC' (maximum intensity projection–normalized cross correlation) algorithm to compute the pairwise displacements of all tiles.

(xiii) Select the number of slices per layer. The whole volume is processed one layer at a time, so as to minimize memory use. By default, this parameter is set to '100'.

(xiv) Press the 'Start' button and wait for the tool to process the whole volume, which may take a few minutes. A progress bar indicating the estimated remaining time is shown during processing.
▲ **CRITICAL STEP** This aligning process may take several hours, depending on the volume size of the images.



```
Global features of the neuron:

number of nodes                     : 813
soma surface                        : 2123.72
number of stems                     : 1
number of bifurcations              : 154
number of branches                  : 340
number of tips                      : 186
overall width                       : 661.746
overall height                      : 212.099
overall depth                       : 138
average diameter                    : 8.65437
total length                        : 3926.16
total surface                       : 128610
total volume                        : 458782
max euclidean distance              : 389.559
max path distance                   : 623.483
max branch order                    : 44
average contraction                 : 0.973733
average fragmentation               : 1.38824
average parent-daughter ratio       : 0.988157
average bifurcation angle local     : 76.4216
average bifurcation angle remote    : 71.3935
Hausdorff dimension                 : 1.20496
```

```
Difference-scores of multiple neurons

dists between 1 and 2 are
entire-structure-average = 0.577906
different-structure-average = 5.02747
percent of different-structure =
0.0600187

dists between 1 and 3 are
entire-structure-average = 0.648484
different-structure-average = 4.36782
percent of different-structure =
0.0756287

Average scores between 1 and the
remaining neurons are
entire-structure-average = 0.613195
different-structure-average = 4.69764
percent of different-structure =
0.0678237
```

Figure 8 | Screenshots of automated reconstruction and analysis of the 3D morphology of neurons. (**a**) The 3D view of a *Drosophila* neuron image stack. (**b**) The 3D reconstruction of morphology (red) overlaid on top of the image stack. This overlay display in 3D is helpful for the proof-reading and editing of the reconstruction. (**c**) The decomposed neuron segments, each in a different color, when the reconstruction is being edited. Each segment can be deleted or broken into smaller pieces for further editing. Missing segments can be added on the basis of the 3D curve generation functions or other neuron reconstruction methods in Vaa3D. (**d**) Three reconstructions (in different colors) that correspond to different reconstruction parameters can be overlaid in the same 3D viewer to allow visual comparison. For better visualization, these three reconstructions are displaced slightly via using the 'Lock scene and adjust this object' tool (Step 1F(x)). (**e**) Twenty-two morphology features were calculated for one neuron reconstruction by using the 'global_neuron_feature' plug-in. (**f**) The difference-scores between a manually selected neuron and the remaining neurons in **d**. These scores can be calculated using the 'Distance of two neurons' tool.

(xv) Select the 'Merging' tab.

(xvi) Insert the path where to the stitched volume is to be saved.

▲ CRITICAL STEP The path should point to an empty folder. If the folder is not empty, a warning message informs the user that an error may occur during the merging process.

(xvii) Select the resolutions to save. By default, five resolutions are saved. Starting from the highest-resolution layer, each pixel in the lower-resolution layers is computed as the mean of the 8-pixel cube of the immediately-higher-resolution layer.

(xviii) Select the organization of stacks by either checking 'single-stack' or 'multi-stack'. In the latter case, the dimensions of individual stacks must be specified.

(xix) Press the 'Start' button and wait for the tool to process the whole volume, which may take a few minutes. A progress bar indicating the estimated remaining time is shown during processing.

▲ CRITICAL STEP This merging process may take several hours, depending on the volume size of the images.

(J) Reconstruct and quantify 3D neuron morphology automatically ● TIMING **~5–15 min**

(i) From the Vaa3D Test Data web page, click on 'Neuron tracing test image 1' and download the respective zipped image data set; unzip the files to a new file folder.

(ii) Open the file 'neuron01.tif' using Vaa3D.

(iii) In Vaa3D, go to the 'Plug-in' main window menu and click 'neuron_tracing', then click on 'Vaa3D_Neuron2', and finally click on 'Vaa3D-Neuron2-APP1'.

(iv) Click 'ok' in the parameter-defining dialog.

(v) Wait for a few seconds to see a message box that indicates the amount of time the automated neuron reconstruction has taken and the location of the saved neuron reconstruction file. Click 'OK' to close this dialog.

▲ CRITICAL STEP The saved neuron reconstruction file will have an extension '_app1.swc' and will be in the same file folder as the input image; in this example, it will have the name 'neuron01.tif_x114_y55_z30_app1.swc'.

(vi) Open a 3D viewer for the image 'neuron01.tif' (**Fig. 8**).

(vii) Drag the 'neuron01.tif_x114_y55_z30_app1.swc' file into the 3D viewer.

(viii) Use 'Cmd/Ctrl+L' to toggle between the line (skeleton) display mode and the surface mesh display mode of the neuron.

(ix) Use various tools on the right-side control pane of the 3D viewer to visualize the reconstructed 3D morphology of the neuron (**Fig. 8b**).

(x) Move the mouse cursor on top of the neuron and right-click to activate a neuron-specific pop-up menu.

(xi) Select the 'Edit this neuron' menu. The neuron structure will be turned into a decomposed structure in which each segment will have a different color (**Fig. 8c**).

(xii) Right-click on the neuron structure again; now the neuron-specific pop-up menu will have additional menu items that allow editing (e.g., deleting or annotating a segment) of the neuron.

(xiii) Press 'Alt+P' to activate a 'Surface/Object Advanced Option' dialog to try additional ways to see the contour of the neuron and other features.

(xiv) Repeat Step 1J(iii) and change the 'Visibility threshold' in the parameter-defining dialog from 30 (default value) to 40 and produce a new reconstruction. Rename it to a different file.

(xv) Repeat Step 1J(iii) and change the 'Visibility threshold' in the parameter-defining dialog from 30 (default value) to 50, and produce a new reconstruction. Rename it to a different file.

(xvi) Drag and drop the three reconstructions that correspond to visibility thresholds 30, 40 and 50, respectively, into the same 3D viewer window, so as to overlay them on top of each other to make a comparison.

(xvii) Click the 'Object Manager' on the right-side control pane of the 3D viewer. Next, click the 'Neuron/line Structure' tab, and finally click on the 'Color' button to change the three neurons' colors and differentiate them (**Fig. 8d**).

(xviii) Move the mouse cursor to any neuron and right-click to activate the neuron-specific pop-up menu.

(xix) Select the menu item 'Distance of two neurons' to calculate the Euclidean distance and other difference scores[20] of the currently selected neuron and each of the remaining neurons (**Fig. 8f**).

? TROUBLESHOOTING

(xx) In Vaa3D, go to the 'Plug-in' main window menu and click 'neuron_utilities', then click on the 'global_neuron_feature' plug-in, and finally click on the 'compute global features' menu item. This will activate a tool to select a reconstructed neuron file, calculate its morphology features and display an information box indicating 22 global morphological quantities of this neuron (**Fig. 8e**). These quantities are designed to be compatible with those computed using the L-measure tool[26].

? TROUBLESHOOTING

Troubleshooting advice can be found in **Table 3**.

TABLE 3 | Troubleshooting table.

Step	Problem	Possible reason	Solution
Box 1	Cannot find Vaa3D program under /Applications/vaa3d	The user does not have appropriate administrator privileges to install a program	Obtain the administrator permission to install a program, and also change the permissions for the /Applications folder so that it is writeable by the user
	On Windows OS cannot find as many Vaa3D plug-in programs as on Mac and Linux OSs	Some functions used by different plug-ins may be dependent on the actual OS	Make changes to the respective source code of the plug-ins to make them buildable under Windows by following http://code.google.com/p/vaa3d/wiki/BuildVaa3D. Many examples can be found in other plug-ins' source code
1A(ii)	Image file is not open	Wrong data type or file format	Try to load files as TIFF, LSM, MRC or Vaa3D's raw file format. Some other formats can be loaded through some plug-ins such as Bioformats
1A(iv)	3D viewer does not display anything, is too slow, or has some disabled buttons in the 3D viewer	The OpenGL version of the graphics card is too old. This may be because a too-old graphics card has been used, or because the inappropriate setting of OpenGL of the graphics card is used, or because there is a bug in the OpenGL driver for the graphics card	In a 3D viewer window, press 'Ctrl+Shift+I' to see the OpenGL version and other information of the graphics card. For some new graphics cards, especially on the Windows system, the OpenGL version (or some graphics features) can or should be adjusted. For old graphics cards, check whether the latest driver is used, or whether there are some known bugs in the card's driver program. You can also download the free tool GLView to test the OpenGL capability
1B(vi)	Cannot load the image sequence	Files in image series might be named incorrectly, or individual files have a file format that is not recognizable by Vaa3D	Make sure individual files have a file format supported by Vaa3D (e.g., TIFF, Vaa3D's raw file format). The files in the image sequence should be named as follows: x_001.tif, x_002.tif, and so on. Check the file names of the sample data set for an example

(continued)

TABLE 3 | Troubleshooting table (continued).

Step	Problem	Possible reason	Solution
1C(vi)	Marker defined at a different location from the expectation	The image may contain too much noise, or an incorrect color channel has been used	Although Vaa3D is able to estimate what is the most probable color channel a user is using for defining a marker, the color channel can be explicitly specified by pressing a number key, '1' (for the first channel), '2' (for the second channel), and so on, while right-clicking or using the option box on the right-side control pane of the 3D viewer. For a noisy image, if a marker cannot be unambiguously defined using one mouse click, the marker can be defined correctly using either two mouse-clicks or the one-mouse-stroke menu items in the image-specific pop-up menu
1C(xii)	Marker defined at a different location from the expectation	The image may contain too much noise, or the color channel has not been specified correctly	Among several possible solutions, one simple way is to first define all markers using the method in Step 1C(vi), and then click the marker-specific menu to add them into a marker pool, and finally define a 3D curve based on the marker pool
1G(vi)	Cannot find 'qmake' or 'make'	The compiling environment and/or Qt development environment have not been set up	Check the Vaa3D GoogleCode site (**Table 2**) and follow instructions provided there
1G(x)	Plug-in cannot run	Wrong setting of 'Debug' or 'Release' version or other differences from the main Vaa3D program; or need additional dependency library files	Try to rebuild Vaa3D and the respective plug-in using the same compiling and Qt development environment, with the same 'Release' or 'Debug' option, and make sure all dependency library files that are needed by the plug-in are in the search paths
1I(iv)	Error: 'The inserted path does not exist'	Spaces in the path	Move the acquisition/XML file to a path without spaces
	Error: 'A problem occurred during scanning of subdirectories'	One or more stacks is missing	Check that all stacks are not empty, and that all the first-level folders contain the same number of second-level folders
	Error: 'Unable to find stacks in the given directory'	Wrong root folder is selected	Check that the path shown in the 'Import form' panel points to the root folder of the acquisition
1J(xix)	The menu item 'Distance of two neurons' does not appear	Fewer than one neuron has been loaded into the same 3D viewer	Many Vaa3D pop-up menus, including the neuron-specific pop-up menu, are designed to be 'smart' and context-sensitive. Thus, make sure two or more neurons are loaded to make this menu item visible

● TIMING

Step 1A, visualize a 3D multicolor image: 2–5 min
Step 1B, visualize a 5D image data set such as a series of multicolor 3D images: 2–5 min
Step 1C, generate 3D markers and measure image content quantitatively using landmarks: ~3–10 min
Step 1D, generate, quantify, measure and annotate 3D surface meshes: ~3–10 min
Step 1E, compare, fuse and manage many 3D images: ~3–10 min
Step 1F, visualize heterogeneous images and respective surface objects: 2–10 min
Step 1G, extend Vaa3D functions with its plug-in interface: ~2 min–2 h
Step 1H, profile cells quantitatively in 3D on the basis of automated image filtering, segmentation and mesh generation: ~5 min–1 h

Step 1I, stitch terabytes of 3D images automatically: ~20 min–2 h
Step 1J, reconstruct and quantify 3D neuron morphology automatically: ~5–15 min
Box 1, setting up Vaa3D: 5–20 min

In most cases, the visualization runs in real time and analysis can be done within minutes when the user is familiar with Vaa3D. When a different test data set is used, the analysis steps may take a longer time, especially for the TeraStitcher plug-in, when a massive amount of image data set is tested. A user may need more time to accomplish some of the tasks, depending on the computer hardware and software configuration, the network speed and the user's experience with Vaa3D.

ANTICIPATED RESULTS

Vaa3D provides a one-stop solution on commonly used PCs for streamlined visualization, exploration and analysis of large-scale image data sets, as exemplified in **Figures 1–8**. In addition, Vaa3D and many of its plug-ins can be pipelined using scripts.

For visualization of multidimensional images (Steps 1A and 1B), Vaa3D allows real-time visualization of the volumetric image content in the 3D space on the basis of MIP and alpha-blended rendering. The rendering frame rate is ~20–50 Hz in most commonly used graphics cards and computers we have tested. Combined with advanced rendering modes, such as asynchronous rendering[20] and especially with local 3D viewers, the system is able to visualize multigigabyte images simultaneously at both high speed and high resolution. This can be extended to visualize terabyte-sized image stacks in real time via the plug-in interface (Step 1G). As each 3D viewer and each looking-glass zoomed-in box in a tri-view window can have a completely different color-mapping scheme from those of the main global 3D viewer and the tri-view, it is convenient to visualize the multidimensional large image data sets in a flexible and comprehensive way. Subvolume visualization can also be achieved in a straightforward way by either generating a 3D ROI–based local 3D viewer or directly restricting the 3D region of display in a 3D viewer. Various rendering methods (e.g., MIP, alpha-blended views, cross-sectional views, animation mode) can be switched from one to the other without any delay.

Quantitative analysis of images relies on the generation of various surface objects (Steps 1C, 1D, 1H and 1J). With Vaa3D, it is efficient to generate 3D objects, including 3D markers, 3D lines and curves, irregular surface meshes or tree-like structures (e.g., neurons). Vaa3D also allows real-time profiling of the image content on the basis of these generated objects.

It is often required to go beyond a single image data set in many applications. Examples include colocalization of patterns across a number of samples, comparison of multiple images of various specimens or image-computing parameter settings and fusion of images from different sources. In Step 1E, Vaa3D allows colocalizing and comparing patterns in many tri-views and 3D viewers. When massive data sets that contain a number of images are considered, Vaa3D is able to manage them by using the Atlas Manager; it can then selectively fuse information for the data channels of interest.

Vaa3D's ability to visualize heterogeneous data sets, including multidimensional images and multiple types of surface objects, all within one single 3D viewer (Step 1F), enables highly effective ways to proofread, edit, annotate and quantify various surface objects and image computing results directly in 3D. This feature is crucial for a range of biological imaging–related applications. Steps 1H–1J demonstrate several applications in different domains.

Vaa3D is extensible, especially through its plug-in interface (Step 1G). On Mac and Linux, it often needs a minimal effort to connect to other existing C/C++-based image computing codebases and thus to allow these programs to take advantage of Vaa3D's visualization and analysis capabilities. On Windows, in most cases, Vaa3D can be extended similarly with the correct program development environment setup. In the remaining cases, some Windows OS–specific functions should be used.

The three demonstrated pipelines (Steps 1H–1J) can be reused for similar applications that are often seen in current biology. Some of the parameters, such as the radius of the median filter, the type of watershed and so on, may need to be fine-tuned for the image data of interest. The extraction and quantification of image objects, including cells and complicated cellular reconstructions, often can be done in 3D within a short amount of time, especially when compared with other 2D-based tools. Importantly, a user may also use a number of other released Vaa3D modules and plug-ins, which we have not described here, together with these three exemplar pipelines for more sophisticated studies. Information on these additional tools can be found in various Vaa3D resources shown in **Table 2**.

ACKNOWLEDGMENTS We thank Z. Ruan, H. Xiao and Y. Wan for developing several modules briefly discussed in this article; L. Qu, Y. Yu, J. Zhou, L. Ibanez, P. Yu, C. Bruns and many other contributors to the Vaa3D project; C. Doe, E. Heckscher, R. Kerr, J. Simpson, P. Chung, G. Rubin, L. Silvestri, L. Sacconi, F.S. Pavone, http://NeuroMorpho.org/, the Janelia FlyLight project and many other collaborators for providing test data for the demonstrations. We thank the Janelia Farm Research Campus of the Howard Hughes Medical Institute and the Allen Institute for Brain Science for support of this work.

AUTHOR CONTRIBUTIONS H.P. conceived and developed this study; A.B. and G.I. developed the TeraStitcher plug-in; F.L. developed the cell segmentation plug-in; H.P. supervised or assisted all plug-in developments; H.P. wrote the manuscript with the contributions from co-authors. Z.Z. helped in testing the options and in editing the manuscript.

COMPETING FINANCIAL INTERESTS The authors declare no competing financial interests.

Reprints and permissions information is available online at http://www.nature.com/reprints/index.html.

1. Long, F., Zhou, J. & Peng, H. Visualization and analysis of 3D microscopic images. *PLoS Comput. Biol.* **8**, e1002519 (2012).
2. Clendenon, J.L., Phillips, C.L., Sandoval, R.M., Fang, S. & Dunn, K.W. Voxx: a PC-based, near real-time volume rendering system for biological microscopy. *Am. J. Physiol. Cell Physiol.* **282**, C213–C218 (2002).
3. Swedlow, J.R., Goldberg, I., Brauner, E. & Sorger, P.K. Informatics and quantitative analysis in biological imaging. *Science* **300**, 100–102 (2003).
4. Abràmoff, M.D., Magalhães, P.J. & Ram, S.J. Image processing with ImageJ. *Biophotonics Internatl.* **11**, 36–42 (2004).
5. de Chaumont, F. *et al.* Icy: an open bioimage informatics platform for extended reproducible research. *Nat. Methods* **9**, 690–696 (2012).
6. Sommer, C., Straehle, C., Kothe, U. & Hamprecht, F.A. Ilastik: Interactive learning and segmentation toolkit. *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on* 230–233 (2011).
7. Carpenter, A.E. *et al.* CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol.* **7**, R100 (2006).
8. Murphy, R.F. Cell organizer: image-derived models of subcellular organization and protein distribution. *Comput. Methods Cell Biol.* **110**, 179 (2012).
9. Long, F., Peng, H., Liu, X., Kim, S.K. & Myers, E. A 3D digital atlas of *C. elegans* and its application to single-cell analyses. *Nat. Methods* **6**, 667–672 (2009).
10. Luisi, J., Narayanaswamy, A., Galbreath, Z. & Roysam, B. The FARSIGHT trace editor: an open source tool for 3-D inspection and efficient pattern analysis aided editing of automated neuronal reconstructions. *Neuroinformatics* **9**, 305–315 (2011).
11. Kvilekval, K., Fedorov, D., Obara, B., Singh, A. & Manjunath, B. Bisque: a platform for bioimage analysis and management. *Bioinformatics* **26**, 544–552 (2010).
12. Lau, C. *et al.* Exploration and visualization of gene expression with neuroanatomy in the adult mouse brain. *BMC Bioinformatics* **9**, 153 (2008).
13. Peng, H. *et al.* BrainAligner: 3D registration atlases of *Drosophila* brains. *Nat. Methods* **8**, 493–498 (2011).
14. Peng, H. Bioimage informatics: a new area of engineering biology. *Bioinformatics* **24**, 1827–1836 (2008).
15. Swedlow, J.R., Goldberg, I.G. & Eliceiri, K.W. Bioimage informatics for experimental biology. *Ann. Rev. Biophys.* **38**, 327 (2009).
16. Shamir, L., Delaney, J.D., Orlov, N., Eckley, D.M. & Goldberg, I.G. Pattern recognition software and techniques for biological image analysis. *PLoS Comput. Biol.* **6**, e1000974 (2010).
17. Danuser, G. Computer vision in cell biology. *Cell* **147**, 973–978 (2011).
18. Eliceiri, K.W. *et al.* Biological imaging software tools. *Nat. Methods* **9**, 697–710 (2012).
19. Schindelin, J. *et al.* Fiji: an open-source platform for biological-image analysis. *Nat. Methods* **9**, 676–682 (2012).
20. Peng, H., Ruan, Z., Long, F., Simpson, J.H. & Myers, E.W. V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. *Nat. Biotechnol.* **28**, 348–353 (2010).
21. Gonzalez-Bellido, P.T., Peng, H., Yang, J., Georgopoulos, A.P. & Olberg, R.M. Eight pairs of descending visual neurons in the dragonfly give wing motor centers accurate population vector of prey direction. *Proc. Natl. Acad. Sci. USA* **110**, 696–701 (2013).
22. Yu, Y. & Peng, H. Automated high speed stitching of large 3D microscopic images. *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'2011)*. 238–241 (2011).
23. Bria, A. & Iannello, G. TeraStitcher-a tool for fast automatic 3D-stitching of teravoxel-sized microscopy images. *BMC Bioinformatics* **13**, 316 (2012).
24. Silvestri, L., Bria, A., Sacconi, L., Iannello, G. & Pavone, F. Confocal light sheet microscopy: micron-scale neuroanatomy of the entire mouse brain. *Optics Express* **20**, 20582–20598 (2012).
25. Peng, H., Long, F. & Myers, G. Automatic 3D neuron tracing using all-path pruning. *Bioinformatics* **27**, i239–i247 (2011).
26. Scorcioni, R., Polavaram, S. & Ascoli, G.A. L-Measure: a web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies. *Nat. Protoc.* **3**, 866–876 (2008).