V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets

Hanchuan Peng, Zongcai Ruan, Fuhui Long, Julie H Simpson & Eugene W Myers

The V3D system provides three-dimensional (3D) visualization of gigabyte-sized microscopy image stacks in real time on current laptops and desktops. V3D streamlines the online analysis, measurement and proofreading of complicated image patterns by combining ergonomic functions for selecting a location in an image directly in 3D space and for displaying biological measurements, such as from fluorescent probes, using the overlaid surface objects. V3D runs on all major computer platforms and can be enhanced by software plug-ins to address specific biological problems. To demonstrate this extensibility, we built a V3D-based application, V3D-Neuron, to reconstruct complex 3D neuronal structures from high-resolution brain images. V3D-Neuron can precisely digitize the morphology of a single neuron in a fruitfly brain in minutes, with about a 17-fold improvement in reliability and tenfold savings in time compared with other neuron reconstruction tools. Using V3D-Neuron, we demonstrate the feasibility of building a 3D digital atlas of neurite tracts in the fruitfly brain.

Quantitative image analysis is rapidly becoming the bottleneck in high-throughput experiments that seek to gain new insights from advanced 3D microscopic digital images collected using multiple fluorescent probes of different colors, and optionally also for a series of time points[1,2]. This task is often challenging owing to the complexity of the multidimensional image objects in terms of shape (e.g., neurons) and texture (e.g., subcellular organelles), the large scale of the image data, which currently is in the range of hundreds of megabytes to several gigabytes per image stack, the low or limited signal-to-noise ratio of the image data, and the inapplicability of many two-dimensional (2D) image visualization and analysis techniques in these higher dimensional situations.

Real-time 3D visualization-assisted analysis can effectively help an investigator produce and proofread biologically meaningful results in these complicated situations. It is thus highly desirable to have high-performance software tools that can simultaneously visualize large multidimensional image data sets, allow the user to interact and annotate them in a computer-assisted manner, and extract quantitative analysis results.

Many existing software packages, such as Amira (Visage Imaging), ImageJ[3], Chimera[4], Neurolucida (MBF Bioscience) and Image Pro (MediaCybernetics), can be classified into this visualization-assisted analysis category when an image under consideration has a relatively small volume or is primarily processed in a slice-by-slice 2D manner. The Visualization Toolkit[5], in conjunction with the Insight segmentation and registration toolkit[6], is also useful for visualization-assisted analysis of 3D biomedical images. However, the performance of these tools is not yet scalable to multidimensional, multi-gigabyte image data sets.

Here we present a visualization-assisted analysis system called V3D designed to fill this niche. V3D has two distinctive features. First, it

contains a visualization engine that is fast enough to render multi-gigabyte, 3D volumetric data in real time on an ordinary computer and to be scalable to very large images. Second, we developed methods to directly pinpoint any XYZ location in an image volume using just one or two mouse clicks while viewing it directly in 3D. The previous alternatives were to do this in a 2D slice or to use a virtual mouse in an expensive stereo-viewing system. These two characteristics in combination make V3D suitable for performing complicated analyses of multidimensional image patterns in a user-efficient way. We demonstrated the strength of V3D by developing a suite of tools for neuroscience, called V3D-Neuron. We applied V3D-Neuron to reconstructing a 3D neuronal structure and produced a 3D digital atlas of stereotypical neurite tracts in a fruitfly's brain.

RESULTS

Real-time 3D visualization of large-scale heterogeneous data

The cross-platform V3D visualization engine renders heterogeneous data including 3D, four-dimensional (4D) and five-dimensional (5D) volumetric image data and a variety of 3D surface objects. These capabilities are demonstrated in **Figure 1a** (and **Supplementary Video 1**), which in a single 3D view shows a multi-channel image stack of a fruitfly brain, a set of surface objects corresponding to brain compartments and several individually reconstructed neurons.

V3D can render image intensity and color in several ways. For the multidimensional intensity data of image stacks, V3D can render a maximum intensity projection, an alpha-blending projection and a cross-sectional view, as well as permitting arbitrary cutting planes through the image stack. For color channels that are often associated with different fluorescent molecules, V3D provides color mapping for improved visualization. V3D supports four categories of 'model' objects: (i) irregular surface meshes (to model objects such as a brain

Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, Virginia, USA. Correspondence should be addressed to H.P. (pengh@janelia.hhmi.org).
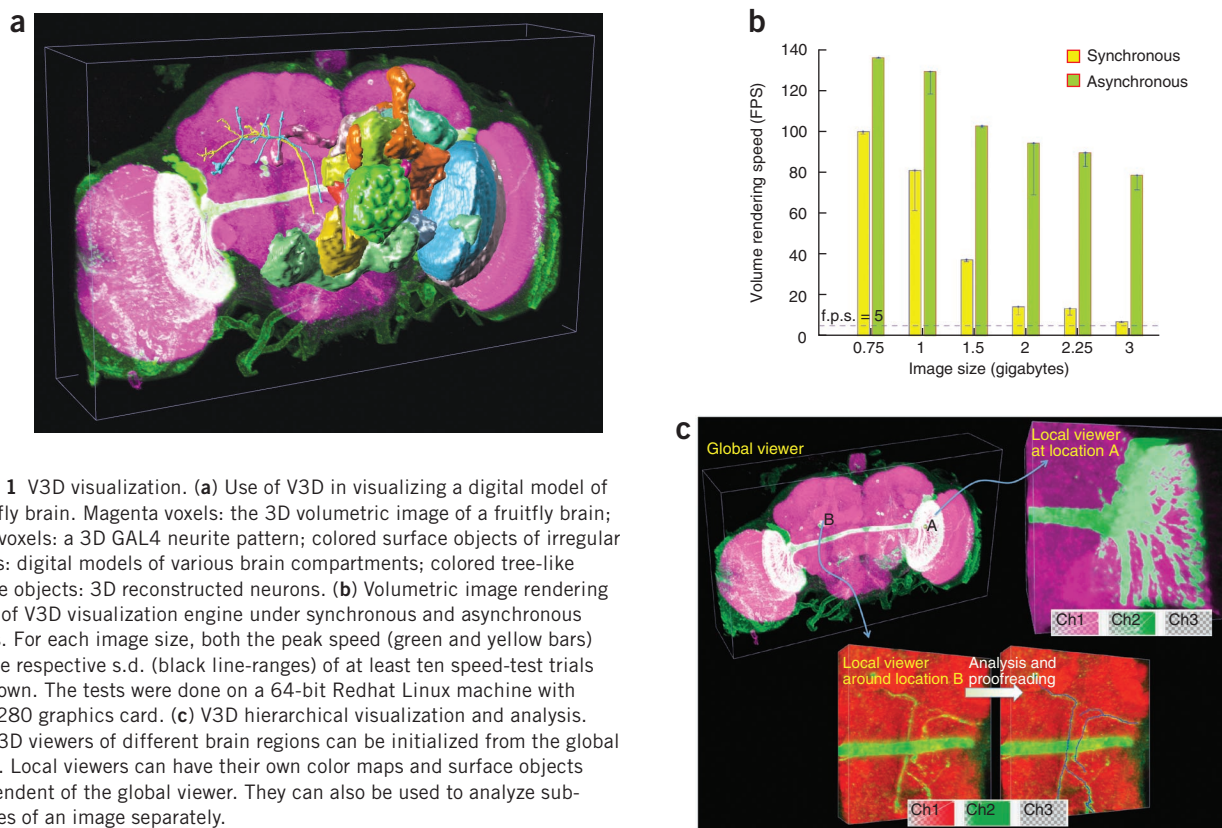
Figure 1 V3D visualization. (**a**) Use of V3D in visualizing a digital model of a fruitfly brain. Magenta voxels: the 3D volumetric image of a fruitfly brain; green voxels: a 3D GAL4 neurite pattern; colored surface objects of irregular shapes: digital models of various brain compartments; colored tree-like surface objects: 3D reconstructed neurons. (**b**) Volumetric image rendering speed of V3D visualization engine under synchronous and asynchronous modes. For each image size, both the peak speed (green and yellow bars) and the respective s.d. (black line-ranges) of at least ten speed-test trials are shown. The tests were done on a 64-bit Redhat Linux machine with a GTX280 graphics card. (**c**) V3D hierarchical visualization and analysis. Local 3D viewers of different brain regions can be initialized from the global viewer. Local viewers can have their own color maps and surface objects independent of the global viewer. They can also be used to analyze sub-volumes of an image separately.

compartment or a nuclear envelope), (ii) spheres (as a simple proxy for globular objects such as cells and nuclei), (iii) fibrous, tubular or network-like structures with complicated branching and connection patterns (to describe the tree-like structure of a neuron, or other relational data or graphs of surface objects), and (iv) markers (to represent various XYZ locations and often used to guide image segmentation, tracing, registration and annotation). Any combination of collections of these objects can be rendered in different colors, styles and in different groupings, on top of an intensity-based multidimensional image. We have used these features to describe a range of digital models of microscopic image patterns, such as a 3D digital nuclear atlas model of *Caenorhabditis elegans*[7] and the fruitfly brain map in **Figure 1a**.

V3D permits interactive rotation, zoom and pan by moving the computer mouse. The speed of the V3D visualization engine is one of the key factors to its usability. It is well known that 3D volumetric image visualization is computationally expensive. To enable real-time 3D rendering of gigabytes of volumetric image data, we first optimized the synchronous rendering of 4D red-green-blue-alpha images using OpenGL 2D or 3D texture mapping (http://www.opengl.org/ and ref. 8), at the full resolution. This resulted in a maximum intensity projection rendering at rates of 15 and 8 frames per second (f.p.s.) in our tests of 2.25- and 3-gigabyte red-green-blue colorimetric images ($1,024 \times 1,024 \times 768$ and $1,024^3$ voxels, respectively) on a 64-bit Linux machine with only 1-gigabyte of graphics card memory (Nvidia GTX280) (**Fig. 1b**). A number of studies show that 5 f.p.s. gives a satisfactory interactive experience[9], indicating that V3D synchronous rendering is sufficiently fast for multi-gigabyte images.

We further designed an asynchronous rendering method performing at almost 80 f.p.s. for a 3-gigabyte image, a tenfold improvement in speed compared to the synchronous rendering (**Fig. 1b**). To do so, we took advantage of the fact that human eyes are not sensitive to the details of a moving object and can distinguish such details only when an object is still[10]. Therefore, in the asynchronous mode, when a user is rotating, or using other ways to interact with a large image, V3D renders a medium-resolution representation of this image. Once such an interaction is over, V3D displays the full resolution. This asynchronous method achieves both high resolution and high speed. It is largely independent of the graphics cards used and is limited by the bandwidth of the peripheral component interconnect express
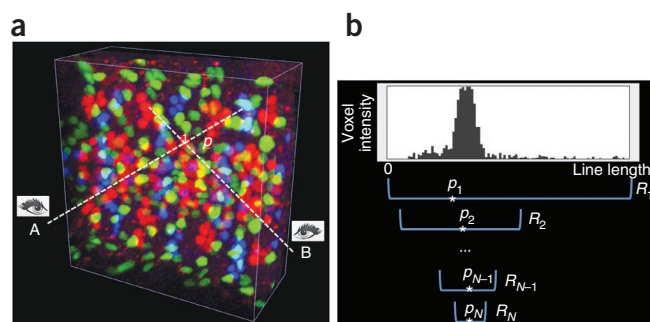

Figure 2 3D pinpointing methods of V3D. (**a**) 3D pinpointing using two mouse-clicks. The color image is a 3D confocal image of neurons, fluorescently tagged for three different transcriptional factors at the same time, in a fruitfly embryo. A and B: non-parallel rays generated at two viewing angles, corresponding to two mouse-clicks; p: the estimated 3D intersection location that is closest to both A and B. (**b**) 3D pinpointing using one mouse-click. p_1 to p_N: the progressively estimated centers of mass; R_1 to R_N: the progressively smaller intervals to estimate p_1 to p_N.
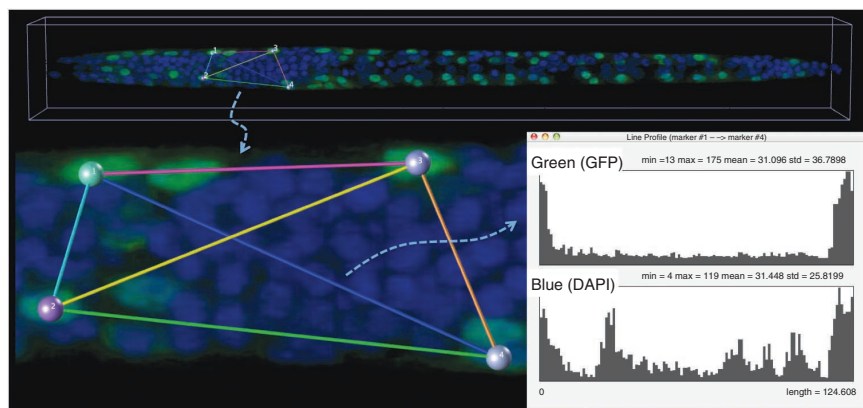
Figure 3 Quantitative measurement of the 3D gene expression level in a *C. elegans* confocal image. Green voxels: myo3: green fluorescent protein (GFP)-tagged body wall muscle cells; blue voxels: DAPI (4,6-diamidino-2-phenylindole)-tagged nuclei for the entire animal; colored spheres: pinpointed markers; colored line-segments: the line-indicator for measuring along different directions and with different starting and ending locations; line profile graph: the channel-by-channel display of the voxel intensity along a line segment.

connection of the underlying computer. We achieved a similar speed improvement on different test machines, such as a Mac Pro with a 512-megabyte graphics card (**Supplementary Fig. 1**).

The only restriction on image size for V3D is that there should be sufficient memory in the underlying hardware. For very large images, for example, 8–16 gigabytes, or for an older machine, the interaction speed becomes an issue. V3D accommodates these situations by displaying two 3D viewers for an image stack: a global viewer and another local viewer of a region of interest dynamically specified by the user (**Fig. 1c** and **Supplementary Video 2a–c**). This combination of dual 3D viewers allows responsive visualization of any detail in a large image stack. Visualization-assisted analysis can be performed in both global and local 3D viewers.

Direct 3D pinpointing of 3D image content
Direct user interaction with the content of a volumetric image requires a user to select one or more points in 3D space, which may be used as 'markers' in subsequent analyses. To complement our fast 3D viewers, we developed two software-based methods to accomplish direct 3D pinpointing within a volumetric image, which has previously only been possible with stereo-viewing hardware or virtual reality equipment.

The first method prompts a user to mouse-click on a point of interest twice while the user is inspecting a volumetric image, each from a different viewing angle (**Fig. 2a** and **Supplementary Video 3a**). Each click defines a ray through the current cursor location orthogonal to the screen. V3D creates a marker at the point in space for which the sum of its Euclidean distance to each ray is minimal (this is to allow for some slight displacement and/or inaccuracy in the user's 2D clicks). This method is independent of the color channels of the image.

The second method pinpoints a 3D marker with just one mouse click (**Fig. 2b** and **Supplementary Video 3b**). Because in this case there is only one ray defined by the click, we estimated the most probable location on the ray that the user intended by looking at the intensity information in a given image color channel. We used the mean-shift algorithm[11] that begins by finding the center of mass (CoM) of the projection ray, and then repeatedly reestimates a CoM using progressively smaller intervals around the proceeding CoM until convergence. This heuristic is very robust in estimating

the desired 3D location in real data sets. When there are multiple color channels, the user specifies one by pressing a number key, e.g. 1, 2, while clicking the mouse.

3D pinpointing is useful for measuring and annotating any spatial location in a multi-dimensional image (**Supplementary Fig. 2**), as well as providing manually selected seeds for more complex visualization-assisted analyses and for manually correcting the output of such analyses. For instance, one can quantitatively profile the voxel intensity along the straight line segment connecting a pair of markers. Thus one could quickly measure the gradient of fluorescently labeled gene expression along an arbitrary direction in an image of *C. elegans* (**Fig. 3**). We also used 3D pinpointing to rapidly count the number of neurons in the arcuate nucleus of a mouse brain (**Supplementary Video 4**), and to examine the relative displacement of neurons in a 5D image series of a moving *C. elegans* that was produced by selective plane illumination microscopy (**Supplementary Fig. 3a–c** and **Supplementary Video 5**).

V3D-Neuron: 3D reconstruction of neurites
V3D provides basic software functions for building specialized applications to address specific biological questions. To demonstrate this, we considered digital reconstruction or tracing of 3D neurite structures, which is one of the essential yet bottleneck steps in understanding brain circuit structure and function[12]. We developed V3D-Neuron to trace the 3D structure of a neuron or a neurite tract from images. V3D-Neuron also immediately displays the tracing results superimposed upon the raw image data, letting one proofread and correct the tracing results interactively.

We reconstructed a neuron (**Fig. 4** and **Supplementary Video 6**) based on automated searching of the optimal 'paths' connecting a set of markers, which are locations pinpointed by a user to indicate where the tracing should begin and end (**Fig. 4a**). Our algorithm finds a smooth tree-like structure in the image voxel domain to connect one marker (root) to all remaining markers with the least 'cost', as defined below. We treated individual image voxels as graph vertexes and defined graph edges between the spatial neighboring voxels. The weight of an edge is defined as the product of the inverse of the average intensity of the two neighboring voxels and their Euclidean distance. The least-cost path between a pair of markers will be the one that goes through the highest intensity voxels with the least overall length. We used Dijkstra's algorithm[13] to find the shortest paths (parent-vertex lists) from every non-root marker to the root. We detected in these lists the vertexes where two or more paths merge. These vertexes were treated as the branching points of the reconstructed neuron tree. Subsequently, we represented the traced neuron using the individual segments that connect markers and branching points (**Fig. 4b**), which can be easily edited in 3D whenever needed. For each segment, V3D-Neuron lets an investigator optionally refine its skeleton using a deformable curve algorithm[14] that fits the path as close as possible to the 'midline' of the image region of this segment. In most cases, this not only leads to a smoother path that better approximates the skeleton of a neuron (**Fig. 4c**) but also more accurately estimates the radius of the traced neuron along this skeleton.
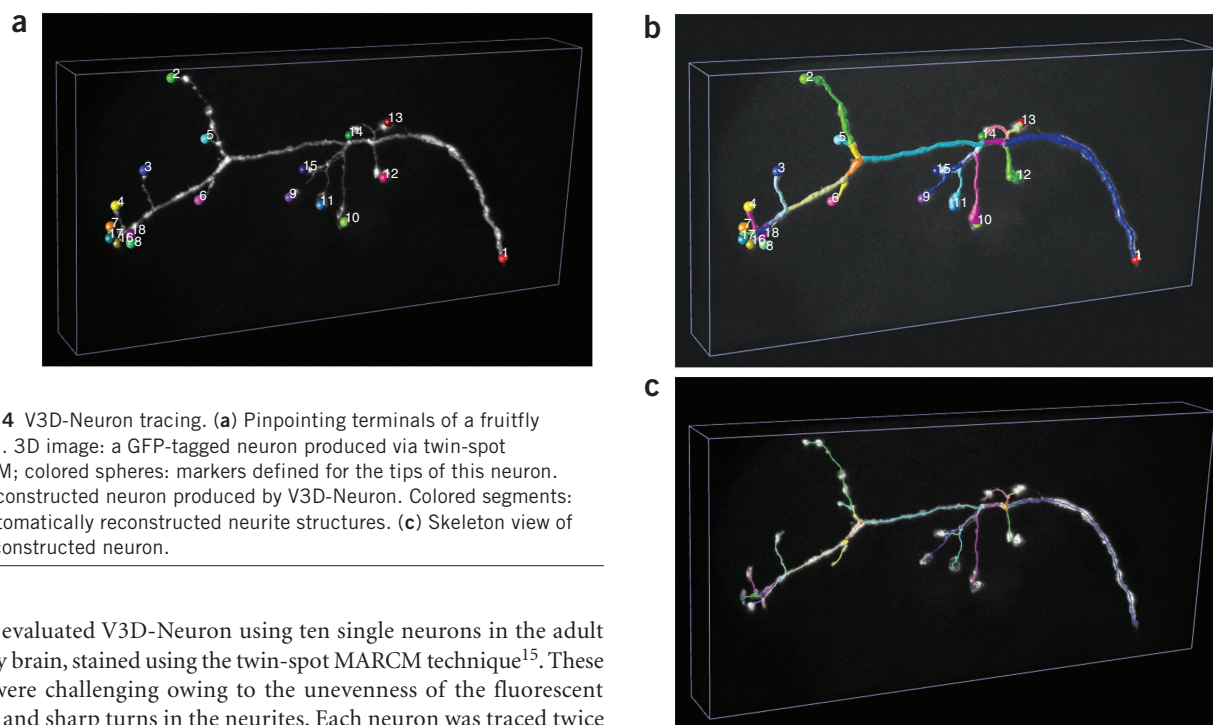
Figure 4 V3D-Neuron tracing. (**a**) Pinpointing terminals of a fruitfly neuron. 3D image: a GFP-tagged neuron produced via twin-spot MARCM; colored spheres: markers defined for the tips of this neuron. (**b**) Reconstructed neuron produced by V3D-Neuron. Colored segments: the automatically reconstructed neurite structures. (**c**) Skeleton view of the reconstructed neuron.

We evaluated V3D-Neuron using ten single neurons in the adult fruitfly brain, stained using the twin-spot MARCM technique[15]. These data were challenging owing to the unevenness of the fluorescent signal and sharp turns in the neurites. Each neuron was traced twice independently. V3D-Neuron allowed us to reconstruct each neuron and correct the potential errors, typically in 3–5 min. Compared to the state-of-the-art 2D manual reconstruction tool of Neurolucida (also with two independent tracings for each neuron), V3D-Neuron avoided the inconsistencies between manual tracings (**Fig. 5a**, e_1–e_4), which were due to human variation in tracing during distinct trials. The difference between V3D-Neuron reconstructions occurred mostly at the

tips of the neurons (**Fig. 5a**, e_5) because of the human variation in pinpointing terminal markers. Quantitatively, V3D-Neuron reconstructions have sub-pixel precision, whereas the manual reconstructions have a remarkably bigger variation for the entire structure (**Fig. 5b**). Taking a closer look at the proportion of neuron skeletons that have visible deviation (that is, >2 voxels) between two trials, we found on average only 1.26% of the total path length of a reconstruction was >2 voxels from its counterpart when traced with V3D-Neuron, whereas the same statistic was 21.8% for Neurolucida (**Fig. 5c**). In addition, producing the reconstruction in V3D-Neuron took an order of magnitude less time, as we spent more than 2 weeks obtaining the Neurolucida results. A comparison between V3D-Neuron and other tools (e.g., Image Pro, see **Supplementary Note**) also showed that V3D-Neuron was superior.
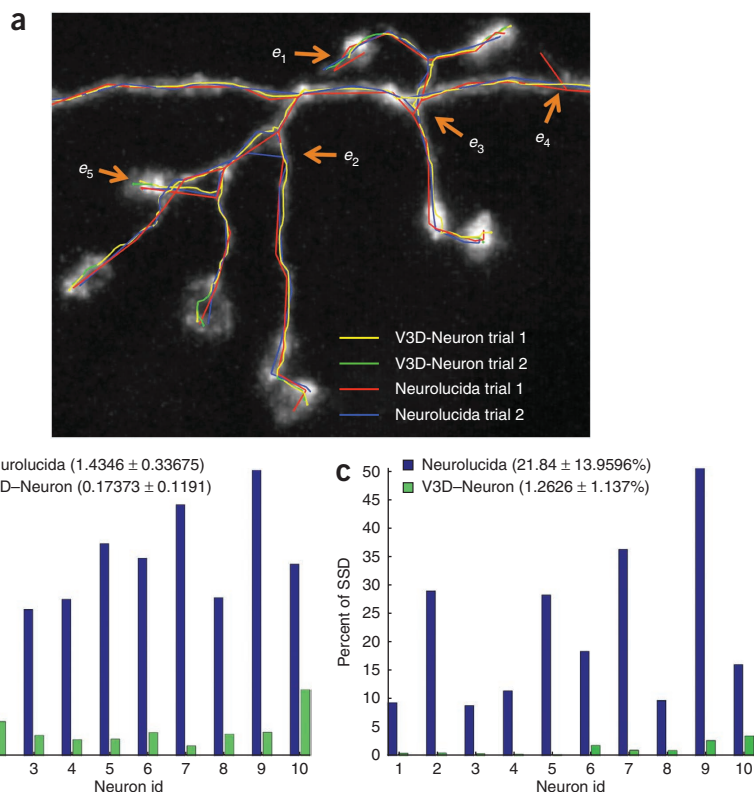

Figure 5 Accuracy of V3D-Neuron reconstructions compared with manual reconstructions. (**a**) Inconsistency of independent trials of reconstructions. e_1, e_2, e_3, e_4: examples of the obvious inconsistent parts in manual reconstructions; e_5: an example of the inconsistent region in V3D-Neuron reconstructions. (**b**) Spatial divergence of reconstructed neurons using different methods, each with two independent runs. Also shown in the legend are the average and the s.d. of the spatial divergence (Online Methods) of all neurons. (**c**) Percent of the neuron structure that noticeably varies in independent reconstructions. Also shown in the legend are the average and the s.d. of this score over all neurons. SSD: substantial spatial distance.

Figure 6 An atlas of stereotyped neurite tracts in a fruitfly brain. (**a**) Statistical models of the 3D reconstructed neurite tracts. Grayscale image: the 'typical' fruitfly brain which was used as the target in 3D alignment; each colored tubular structure: the average of multiple neurite tracts reconstructed from images of the same GAL4 line. The width of each tract equals twice the spatial variation of the respective group of reconstructions. (**b**) Distribution of the spatial variation of all neurite tracts.
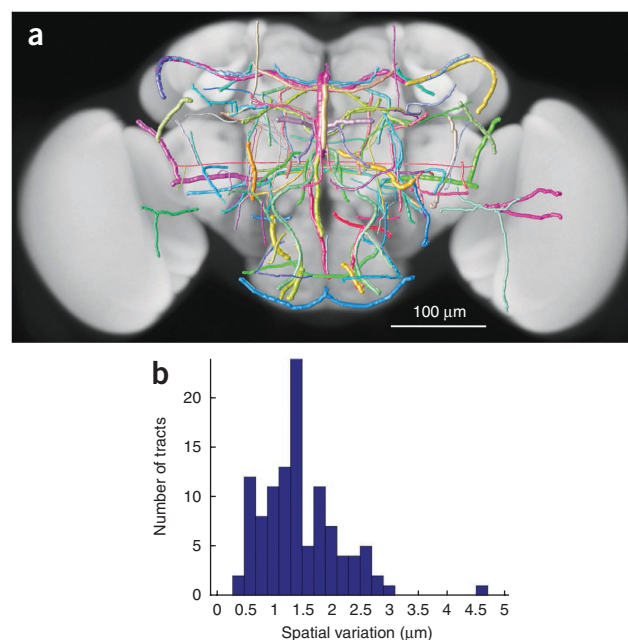

A comprehensive 3D atlas of neurite tracts in a brain is critical for understanding both the anatomy and functions of the brain. To demonstrate the feasibility of building such an atlas with V3D-Neuron, we started with more than 3,000 confocal image stacks of the adult fruitfly brain taken of 500 fruitfly enhancer trap GAL4 lines (unpublished data). First, these images were deformably aligned to a 'typical' brain (**Fig. 6a**) using an NC82 intensity pattern in a separate channel (unpublished data), which serves as the reference for accomplishing the automatic alignment. Then, we used V3D-Neuron to trace 111 representative neurite tracts that project throughout the brain (**Fig. 6a** and **Supplementary Video 7**). We reconstructed two to six instances of each tract from multiple replicate images of the same GAL4 line. We produced a mean tract model for each group of reconstructed instances and the average deviation of members of the group from this mean. The overall shape and relative positions of these tracts are reproducible with deviations in the range from 0.5 to 4.5 µm (**Fig. 6b**). This variance approximates the upper bound on the biological variability, as some portion of it was due to alignment optimization differences. Compared to the typical size of an adult fruitfly brain (590 µm × 340 µm × 120 µm), the deviations are small, indicating that the layout of these neural tracts is highly stereotyped in a fruitfly brain.

DISCUSSION

V3D provides an efficient and ergonomic platform for visualization-based bioinformatics in neuroscience, molecular and cell biology, and general biomedical image computing. Compared to many existing visualization-assisted analysis tools, V3D provides real-time visualization of multi-gigabyte volumetric images and the ability to interact with and quantitatively measure the content of such images directly in 3D. Together, these features facilitate rapid computer-assisted analyses and proofreading of the output of such analyses. V3D can also be used for fully automatic analyses of massive image data sets. The software runs on all major computer platforms (Mac, Linux and Windows). It is further enhanced by various built-in V3D toolkits for better visualization and processing, and is extensible by means of user-developed plug-ins (Online Methods). V3D includes a carefully designed graphical user interface and supports common file formats.

The emerging flood of multidimensional cellular and molecular images poses enormous challenges for the image computing community. Although fully automated high-throughput analyses are desired, these approaches are difficult to perfect and are still in their infancy. In this light, it will be helpful to have real-time visualization-assisted tools like V3D to develop, debug, proofread and correct the output of automated algorithms. Indeed, we have already used V3D in developing a number of automatic image analysis methods in addition to V3D-Neuron, including 3D cell segmentation[7], 3D brain registration and 3D neuron structure comparison, among others. Some of these methods, such as the 3D brain registration algorithm, are being deployed in the high-throughput automatic mapping of the 3D neuroanatomy of a fruitfly (unpublished data).

Image analysis is particularly challenging when the signal-to-noise ratio is low or when only part of an image needs to be analyzed, such as reconstructing one neuron from several that are present in an image. The neurite tracing algorithm in V3D-Neuron addresses this challenge by combining global cues supplied by a user and local cues generated automatically from the image data, allowing the user to guide the software when fully automated analysis is difficult. In contrast, existing neuron reconstruction algorithms[16–20] either have not considered such a combination, or are limited to 2D. More importantly, our method is able to complement these algorithms by enabling them to be invoked within V3D, possibly with some guiding marker information, and then their results proofread and corrected in a faster and more accurate way.

Stereotypy, which often refers to the biological invariability, and the degree to which it occurs in the gross and fine-scale anatomy of fly neurons is a question of biological significance and debate[21,22]. The stereotypy of neurite locations found in this work indicates that it may be fruitful to add more neurites to our preliminary atlas and to use these neurites as templates to retrieve and catalog unknown fruitfly neurite patterns. Such information will be useful in modeling 3D anatomical projections and connectivity diagrams of neurons in a brain and in investigating neuron functions. The approach presented here is applicable to a wide range of organisms and at different imaging scales. V3D and V3D-Neuron also have the potential to allow one to manipulate the image data in real-time during functional imaging.

METHODS

Methods and any associated references are available in the online version of the paper at http://www.nature.com/naturebiotechnology/.

Note: Supplementary information is available on the Nature Biotechnology website.

Madison) for assistance in implementing a V3D plug-in. We also thank G. Rubin, and R. Kerr (Janelia, HHMI) for helpful comments on the manuscript.

1. Peng, H. Bioimage informatics: a new area of engineering biology. *Bioinformatics* **24**, 1827–1836 (2008).
2. Wilt, B.A. *et al.* Advances in light microscopy for neuroscience. *Annu. Rev. Neurosci.* **32**, 435–506 (2009).
3. Abramoff, M.D., Magelhaes, P.J. & Ram, S.J. Image processing with ImageJ. *Biophotonics Int.* **11**, 36–42 (2004).
4. Pettersen, E.F. *et al.* UCSF Chimera—a visualization system for exploratory research and analysis. *J. Comput. Chem.* **25**, 1605–1612 (2004).
5. Schroeder, W., Martin, K. & Lorensen, B. *Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*, 4th edn. (Kitware, Inc, Clifton Park, New York, USA, 2006).
6. Yoo, T.. *Insight into Images: Principles and Practice for Segmentation, Registration, and Image Analysis* (A K Peters, Ltd., Wellesey, Massachusetts, USA, 2004).
7. Long, F., Peng, H., Liu, X., Kim, S. & Myers, E.W. A 3D digital atlas of C. elegans and its application to single-cell analyses. *Nat. Methods* **6**, 667–672 (2009).
8. Wright, R.S. & Lipchak, B.. *OpenGL Superbible*, 3rd edn. (Sams Publishing, Indianapolis, Indiana, 2005).
9. Chen, J.Y.C. & Thropp, J.E. Review of low frame rate effects on human performance. *IEEE Trans. Syst. Man Cybern. A Syst. Hum.* **37**, 1063–1076 (2007).
10. Westheimer, G. Visual acuity. *Annu. Rev. Psychol.* **16**, 359–380 (1965).
11. Fukunaga, K. & Hostetler, L.D. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Inf. Theory* **21**, 32–40 (1975).
12. Roysam, B., Shain, W. & Ascoli, G.A. The central role of neuroinformatics in the national academy of engineering's grandest challenge: reverse engineer the brain. *Neuroinformatics* **7**, 1–5 (2009).
13. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numerische Mathematik* **1**, 269–271 (1959).
14. Peng, H., Long, F., Liu, X., Kim, S. & Myers, E. Straightening *C. elegans* images. *Bioinformatics* **24**, 234–242 (2008).
15. Yu, H., Chen, C., Shi, L., Huang, Y. & Lee, T. Twin-spot MARCM to reveal the developmental origin and identity of neurons. *Nat. Neurosci.* **12**, 947–953 (2009).
16. Al-Kofahi, K. *et al.* Rapid automated three-dimensional tracing of neurons from confocal image stacks. *IEEE Trans. Inf. Technol. Biomed.* **6**, 171–187 (2002).
17. Al-Kofahi, K. *et al.* Median-based robust algorithms for tracing neurons from noisy confocal microscope images. *IEEE Trans. Inf. Technol. Biomed.* **7**, 302–317 (2003).
18. Wearne, S.L. *et al.* New techniques for imaging, digitization and analysis of three-dimensional neural morphology on multiple scales. *Neuroscience* **136**, 661–680 (2005).
19. Zhang, Y. *et al.* Automated neurite extraction using dynamic programming for high-throughput screening of neuron-based assays. *Neuroimage* **35**, 1502–1515 (2007).
20. Meijering, E. *et al.* Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry* **58A**, 167–176 (2004).
21. Jefferis, G.S. *et al.* Comprehensive maps of *Drosophila* higher olfactory centers: spatially segregated fruit and pheromone representation. *Cell* **128**, 1187–1203 (2007).
22. Murthy, M., Fiete, I. & Laurent, G. Testing odor response stereotypy in the *Drosophila* mushroom body. *Neuron* **59**, 1009–1023 (2008).

ONLINE METHODS

V3D design. We developed V3D using C/C++, with GCC compliers. We used Qt libraries (Nokia) for developing the cross-platform user interface. V3D can be compiled as either a 32-bit or a 64-bit program; the latter can load and save multi-gigabyte and bigger image stacks.

The V3D visualization engine optimizes OpenGL calls to maximize the 3D rendering throughput of a computer graphics card. We rendered volumetric data and surface data separately. For volumetric data, we considered a 4D image stack as the basic display object. A 4D image with any number of color channels is always mapped to red-green-blue-alpha channels for rendering. Then the V3D visualization engine produces either 2D or 3D OpenGL textures, whichever the user chooses, to render the colorimetric 3D volume image. V3D visualization engine also detects automatically a graphics card's data compression capability and lets the user control that. V3D allows a user to switch between synchronous and asynchronous modes for volumetric rendering. In the asynchronous mode, V3D renders a medium-size representation while a user is interacting with the currently displayed volume image, and renders the full-size image once such interaction (e.g., rotating, zooming) is over. The medium-size volume, which has $512 \times 512 \times 256$ red-green-blue voxels by default and can be enlarged if the graphics card has enough memory, is good for visual inspection of image details in most practical cases we have tested. For each image, V3D has a default global 3D viewer and a dynamically generated local 3D viewer for any region of interest. Both the global and local viewers in V3D can use either the synchronous or asynchronous rendering methods for visualization-assisted analysis. Because the local 3D viewer often uses a much smaller volume than the entire image, we often reduced the time of an analysis by aggregating the visualization-assisted analysis results in multiple dynamically invoked local 3D viewers. The volumetric rendering of a 5D image was built upon rendering a series of 4D image stacks, each of which corresponds to a time point. For 3D surface data, the V3D visualization engine uses an object list to manage the rendering of irregular surface meshes (for surface objects of any shape, e.g., brain compartments), spherical objects (for point cloud objects, e.g., cells, nuclei), tubular objects with branches (e.g., neurons, or user-defined line segments) and markers (user-defined XYZ locations). These surface objects can be displayed as solid surface, wiring frame, semitransparent surface or contours. For the tubular objects with branches, V3D has an additional skeleton display mode. These different visualization methods together make it easy to observe an image and the respective surface objects. They are useful for a number of tasks, for example, developing new image analysis algorithms, proofreading image analysis results, and annotating the image content or surface objects. The speed-test of V3D for **Figure 1b** was done on a Dell Precision T7400 desktop. For the test of synchronous rendering of **Figure 1b**, we used the compression of the graphics card.

A user can develop plug-in programs for V3D. We designed V3D application interfaces (APIs) for writing plug-ins. A plug-in program can request V3D to supply a list of currently opened images, as well as the markers and the regions of interest a user may have defined for these images. After processing any of these data, a plug-in can display the results in either an existing window or a new image viewer window. In this way, it is convenient to use the V3D visualization engine and 3D interaction of image content to develop new image processing tools.

V3D lets one annotate any XYZ location and measure the statistics of image voxels, including the mean intensity, peak intensity, s.d., volume and anisotropy of the shape of the local image patterns in a local region around this 3D location (**Supplementary Fig. 2**). Because for fluorescent images these statistics usually are associated with biological quantities such as gene expression, V3D makes it possible to quickly generate a data table of measurements of gene expression and protein abundance in a number of 3D regions. V3D also supports other types of quantitative measuring, including profiling voxel intensity along a straight line (**Fig. 3**), and profiling the length and the number of branches of a neuron.

V3D-neuron design and algorithm. To model a neuron (or neurite) digitally, we developed a suite of new 3D tracing, 3D visualization, 3D editing and 3D comparison methods on top of V3D. These functions all together are called V3D-Neuron. The version of V3D-Neuron reported here is 1.0.

We modeled a 3D traced neuron as a graph of 'reconstruction nodes (vertexes)'. Each node is described as a sphere centered at a 3D XYZ location p.

The diameter of the sphere, r, quantifies the width of the neurite fiber at this location. The value r is defined as the diameter of the largest surrounding sphere centered at p but at the same time within this sphere at least 90% (or 99% at a user's choice) of image voxels are brighter than the average intensity of all image voxels. To describe this graph, we used the SWC format[23]. Through a single run of the tracing algorithm, V3D-Neuron produces a tree graph, or a line graph if there is no branching point. Through multiple sequential reconstruction steps, V3D-Neuron is able to produce a forest of these trees, or looped graphs, or any complicated structures of neurites.

V3D-Neuron tracing method integrates both the shortest path algorithm and the deformable curve algorithm. In V3D-Neuron, we also provided several ways for a user to optionally optimize the performance. We allowed using only the set of image voxels whose intensity is greater than the average voxel intensity of the entire image in the shortest path computation. We also allowed considering only six neighboring voxels along X, Y and Z directions of a $3 \times 3 \times 3$ local cube, instead of using all the 26 neighbors, in formulating the shortest path search. These optimizations substantially accelerate our algorithm. We are also developing other methods, such as detecting the 3D markers automatically, to improve the V3D-Neuron tracing. Although we believe the 3D interactive tracing in V3D-Neuron is a merit but not a caveat, V3D-Neuron of course can also reconstruct a neuron in an off-line (non-interactive) manner, similar to all other conventional off-line automatic neuron tracing methods.

V3D-Neuron visualizes in 3D the reconstructed digital model of a neuron right away. We overlaid a reconstruction on top of the raw image in several ways (**Supplementary Video 8**): (1) display the entire model; (2) display a portion of the model at a user's choice; (3) display only the skeleton of the model; (4) display the model semi-transparently; (5) display the contour of the model. These different methods can also be combined in visualizing and proofreading a reconstruction. In this way, it is straightforward to tell whether or not in the reconstruction there is any error, which can be corrected easily using the 3D editing function below.

In the 3D editing mode (**Supplementary Video 9**), V3D-Neuron automatically represents the entire neuron model as the aggregation of segments, each of which is bounded by a pair of tip or branching nodes. V3D-Neuron renders each segment using a different color, lets a user edit in 3D directly its type (axon, dendrite, apical dendrite, soma or a user-defined type), scale its diameter, and allows deleting it if this segment is deemed to be wrong. Each segment can also be broken into smaller pieces, which can be further edited in the same way. V3D-Neuron can scale, rotate, mirror and translate a neuron in 3D. It also allows annotating a neuron using user-supplied information, and undo/redo operations.

V3D-Neuron can also be used to compare the similarity of multiple neurons. It can display many neurons in the same window in 3D, thus allowing visual comparison of their structures (**Supplementary Video 10**). Moreover, it displays the basic information of the morphology of a neuron, including the total length, the number of branches, number of tips and number of segments. It also provides a method to compute the spatial "distance" (SD) of any two neurons. To compute the distance, we first spatially resampled the neuron model so that the distance between adjacent reconstruction nodes is 1 voxel in the 3D space. Then we computed the directed divergence of neuron A from neuron B, or $D_{DIV}(A,B)$, as the average Euclidean distance of all the resampled nodes in A to B. Finally, the undirected spatial distance between A and B, $SD(A,B)$ is defined as the average of $D_{DIV}(A,B)$ and $D_{DIV}(B,A)$. $SD(A,B)$ is a good indicators of how far away of A and B. However, when A and B are fairly close, $SD(A,B)$ does not well quantify the amount of different portions of the two structures. Therefore, we defined another score called substantial spatial distance (SSD), which is the average distance of all resampled nodes that are apart from the other neuron at least 2 voxels. The percentage of resampled nodes that are substantially distal to the counterpart neuron is a robust indicator of how inconsistent of these two neuron reconstructions. Indeed, for the results in **Figure 5**, the SSD scores of V3D-Neuron reconstructions and manual reconstructions are similar, both about 3 voxels. This is because the major SSD parts of V3D-Neuron reconstructions are at the neuronal terminal regions where the variation of human pinpointing is about 3 voxels, similar to that produced using the Neurolucida tool. However, the percentage of SSD nodes of manual reconstructions is much bigger than that of V3D-Neuron reconstructions (**Fig. 5c**). When two neurons are spatially apart from each

other, SD and SSD scores have comparable values. Of course, when an image is quite dark to see, the 3D pinpointing variation of human subjects becomes more pronounced, which may accordingly enlarge the variation of the paths detected by V3D-Neuron.

Data and software. The multi-platform V3D software can be freely downloaded from http://penglab.janelia.org/proj/v3d. Included in **Supplementary Software** is the 64bit Linux version for Redhat and Fedora. A test image stack and an instruction to run V3D are included in **Supplementary Data**.

Additional tutorial movies and test data sets are available at the same web site. The database of stereotyped neurite tracts is provided at http://penglab.janelia.org/proj/flybrainatlas/sdata1_flybrain_neuritetract_model.zip. The entire database can be conveniently visualized using the V3D software itself (**Supplementary Video 10**).

23. Cannon, R.C., Turner, D.A., Pyapali, G.K. & Wheal, W.H. An on-line archive of reconstructed hippocampal neurons. *J. Neurosci. Methods* **84**, 49–54 (1998).