

V3D User Manual

by Hanchuan Peng (pengh@janelia.hhmi.org)
2009-May-04, V3D version v2.031

(Note: This manual has not been carefully examined and thus may contain typos/errors, etc. Also because V3D is been developed rapidly, - you may find some GUI components in newer versions are different from what are discussed below.

This manual does not contain or explain anything about the image analysis modules and plugins of V3D; these additional components are extremely useful. If you have interests in them, you can contact me.)

1. What is V3D?

V3D is a software package for visualizing, measuring, and analyzing 3D image stacks. It also visualizes and generates various surface objects, such as irregularly shaped meshes, digitally reconstructed neurons, landmarks, point clouds, user-defined line segments in 3D space, etc.

The following is a description of V3D at its website:

“A Swiss army knife for visualizing and analyzing large 3D image stacks and various surface data. Not only a fast and handy 3D renderer, but also a container of powerful 3D image analysis (cell segmentation, neuron tracing, brain registration, annotation, etc) and data management modules. With the versatile 3D viz-analysis features and quantitative statistical measurements, V3D provides an excellent desktop solution for various bioimage informatics applications, and also a platform to develop/test/debug new 3D image analysis algorithms which can be deployed for high-throughput processing. V3D streamlines the viz-analysis workflow, - the whole is greater than the sum of its parts!”

The overall design philosophy of V3D is to provide a simplest GUI to maximize the user’s ability to visualize and [quantitatively] analyze the large 3D image stacks used in biology and medical studies. We have tried to make the GUI very simple, a lot of times just intuitive and self-contained. Yet we realize sometimes the image visualization and analysis functions are quite complicated, thus it would still help if we have a documented User Manual. This is why this document comes into being.

1.1 Where to get help?

Four ways to get help.

- Click the “information” icon button at the left-side toolbar of V3D main window, or click the respective help menu-item in the main menu. This will display most (if not all) useful the short-keys, mouse usages, version information, etc.
- Read this user manual carefully.
- Check the V3D website for additional information, example data, or video examples.

- Ask (either email or talk to) Hanchuan Peng, or some other developers in V3D developing team.

1.2 Where is V3D website?

You can always google “Hanchuan Peng” and get the V3D link from the Software page.

1.3 How to cite V3D?

We hope you find V3D useful for your research; if you have used this tool in your work, we ask you to cite V3D as the following:

Peng HC, Ruan ZC, Long FH, (2009) “V3D: A Swiss army knife for visualization and analysis of large 3D biological images”, *HHMI-JFRC Technical Report*, <http://penglab.janelia.org/proj/v3d>.

1.4 What file formats are currently supported in V3D?

- * 3D RGB color image stacks (3D tiff stack, Zeiss LSM, tiff series, MRC image stack (for Electron Microscopy), or Hanchuan'RAW file)
- * 3D irregular shaped surfaces (Wavefront .OBJ, .v3ds)
- * 3D neuron structures or any relational data that can be described as a graph (.swc)
- * 3D point cloud (.apo - simple CSV format)
- * 3D landmarks (*.csv, .marker - simple CSV format)
- * Various linker and atlas files (user defined, see examples below)

1.5 Supported 3D real-time interactive visualization

- * MIP (Maximal intensity projection)
- * Alpha blending
- * Cross-sectional view
- * Surface rendering
- * Volume and surface cut, front cut
- * Color-mapping of voxels and recoloring of surface objects
- * Animation and movie making
- * 3D surface mesh generation
- * 3D neuron tracing
- * 3D landmarking
- * Switching the relative 3D depth (layer) relationships of volumetric data and surfaces
- * tri-view display with colocalization of multiple images, and a number of other functions

1.6 Image analysis functions

- * Quantitative measurements directly in 3D (landmarks and line segments)
- * 3D annotation, landmarking, data management
- * 3D cell segmentation, neuron tracing ^(a)
- * 3D deformable registration ^(a)
- * 3D neuron structure comparison ^(b)

* Multiple image 3D colocalization, blending, and many more!

(a): add-on module

(b): partially add-on module

If you need some of the add-on modules, which are under further development or have been held back due to various reasons, please check with Hanchuan Peng.

1.7 Comparison of V3D and several related software packages



V3D - the whole is greater than the sum of the parts

software	free?	easy to use	viz quality/speed	3D img processing	3D img object annotation	3D interaction of content	large database access	the whole greater than sum of parts
Amira	No	No (Yes to experts)	Ok (great for expert) / speed ok	some	some	Weak	No	☹ (whole smaller than sum)
Matlab	No	Yes (easy to code, diff to fine-tune viz)	Low (can be fined tuned)	Yes (need to learn how to code)	No	No	Do-able	Yes!!
ImageJ	Yes	Yes	low (maybe yes to experts)	Yes (functions limited)	Maybe	No	Do-able	Yes!!
VolRover	Yes	No	--	No	No	No	N/A	N/A
Voxx	Yes	Yes	Ok	No	No	No	No	equal
NeuroLucida	No	No (Yes to experts)	Low	some	Yes (neuron tracing by manual drawing)	No	Maybe	☹
V3D	Yes	Yes	Good	Good	Good	Good	Good	Yes!!

2. V3D GUI Components

V3D has two main GUI components, one is called *tri-view main window*, the other one is called *3D viewer*. In addition, V3D has a number of menus, pop-up dialogs and other information windows.

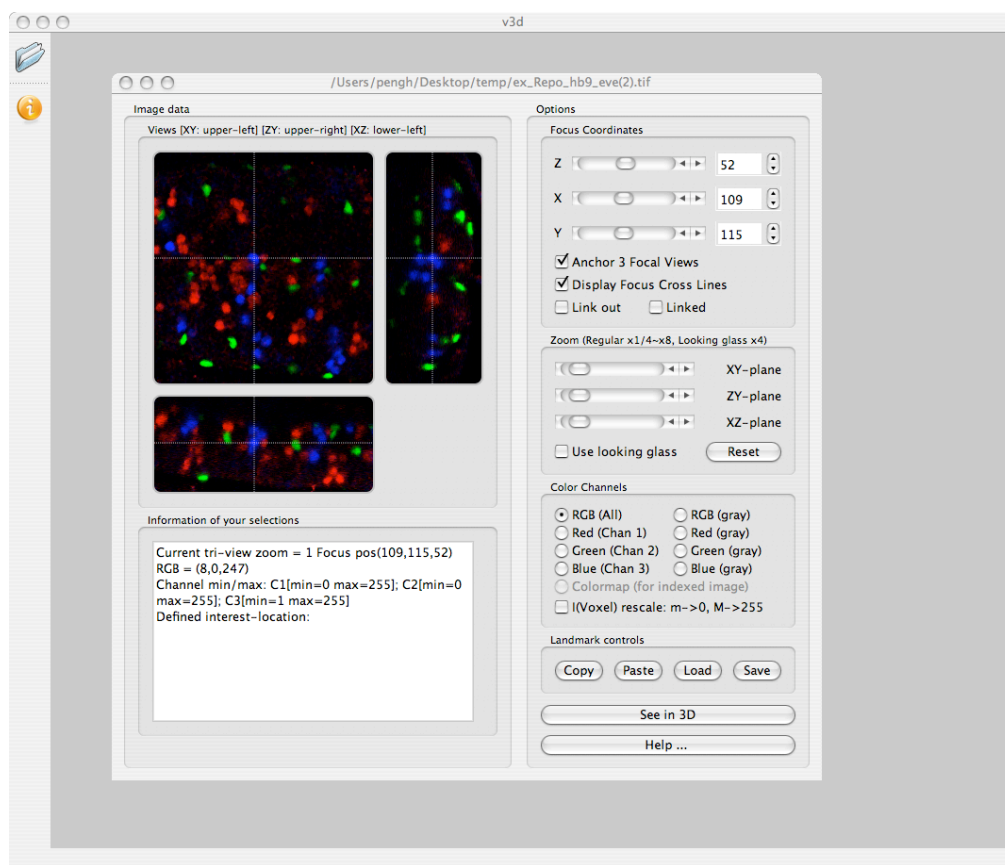
2.1 How to open a file, and brief overview of tri-view main window and 3D viewer

The first step to use V3D is of course to open a file (any of the files discussed in Section 1). Normally this can be done by clicking the “Open file” icon button at the left-side toolbar, or choosing main menu -> “File” -> “Open image/stack in a new window”. This will open either a tri-view main window or a 3D viewer, or both (if V3D is configured to do so). V3D also support **drag and drop file opening**. In most cases, except for some cases of the network drives, an

appropriate image file or surface object file can be directly dragged and dropped into the V3D main window and thus gets opened.

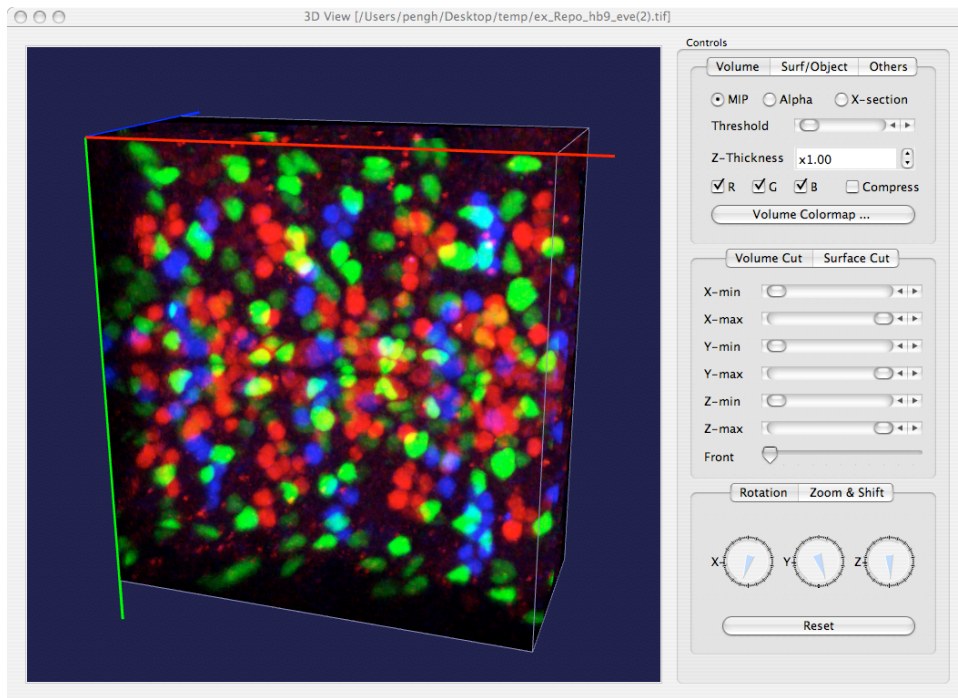
The ***tri-view main window*** is launched when V3D open an image, -- a tri-view window (similar to the figure below) which appears with XY, YZ, and ZX plane anchored together, displaying the 3D information of the stack. There are a number of control buttons at the right side of each opened tri-view window, which lets the user to see image contents at various 3D locations, zoom-in/-out, display one or more color channels, manage the landmarks defined in 3D, or call the 3D viewer.

Below the tri-view display area there is a text area showing the current zoom, focusing location, RGB value of the current focus point (indicated by the cross in the tri-view display), channel color information, and the locations of defined landmarks.



The ***3D viewer*** can be launched (or put to the front if it is occluded by other windows) by pressing the “See in 3D” button in the tri-view display window. It can also be launched by clicking the “3D viewer” menu under the “Visualize” main menu, or simply by pressing “Cmd-V” (for Mac) (“Ctrl-V” for Windows and Linux). A 3D viewer of the image above will look like the following.

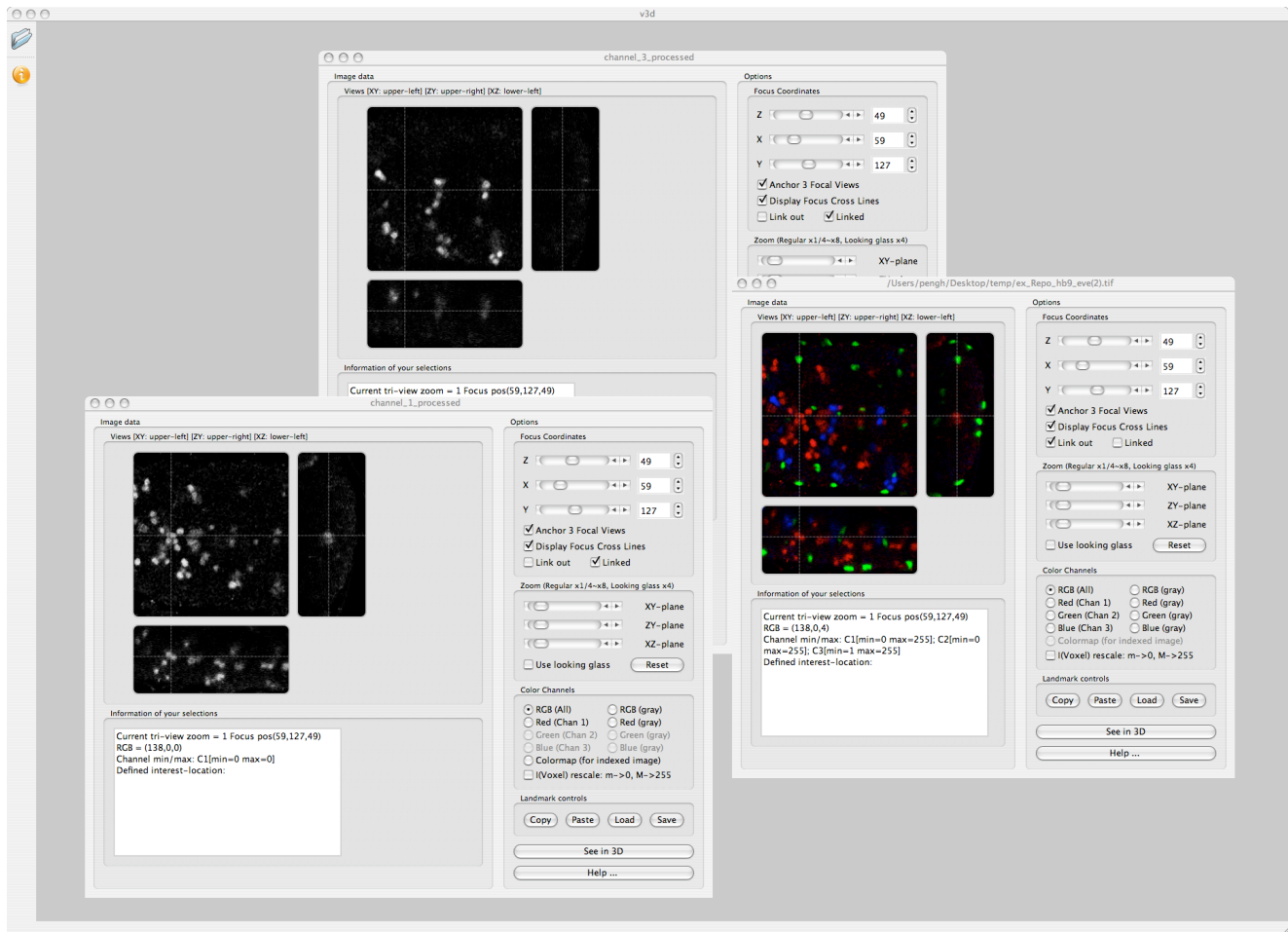
Similar to the tri-view main window, at the right side of the 3D viewer there are a number of control buttons for a user to manage what the rendering modes/effects to be produced. For detailed usages of these functions see blow.



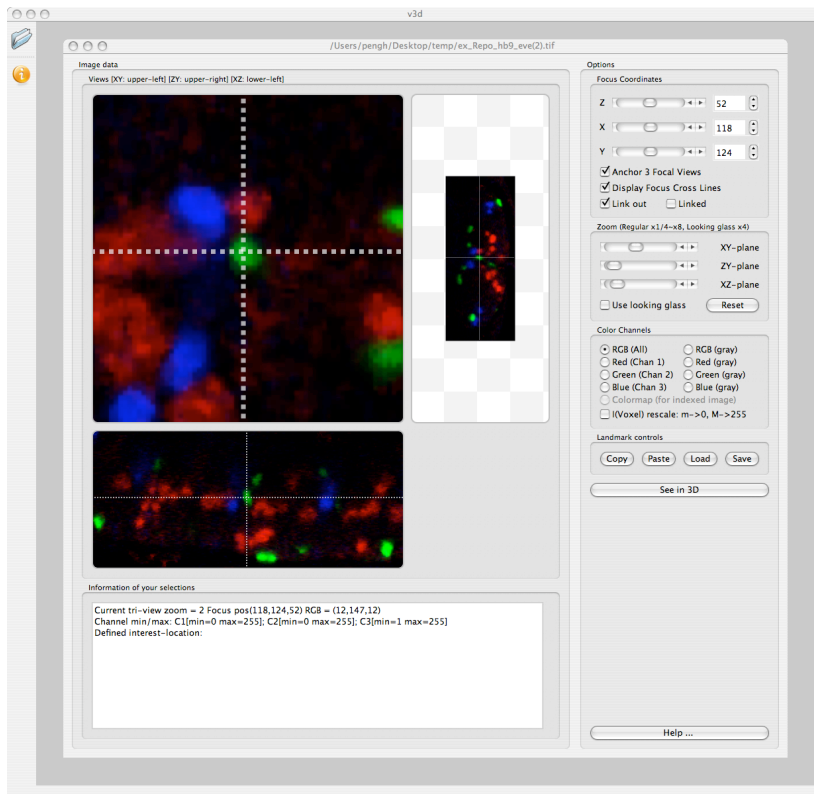
2.2 How to use the Tri-view main window display

The tri-view window is intuitive. The control buttons at the right side let one change the focusing planes, the zoom of the view, and others. One can also change the current displayed plane in a 3D stack by scroll up and down the wheel of the mouse. Depending on the XY, ZY, or XZ plane, scrolling the wheel will change the section of these planes according. Right-clicking the mouse (or the secondary mouse button) will directly change the focusing points to a new location of the mouse pointer, and thus update the displayed image content of two viewer of the tri-view display. Moving the mouse while holding down the primary mouse button (usually the left-mouse button) will pan around the image, if it is displayed in a zoomed view (see details below).

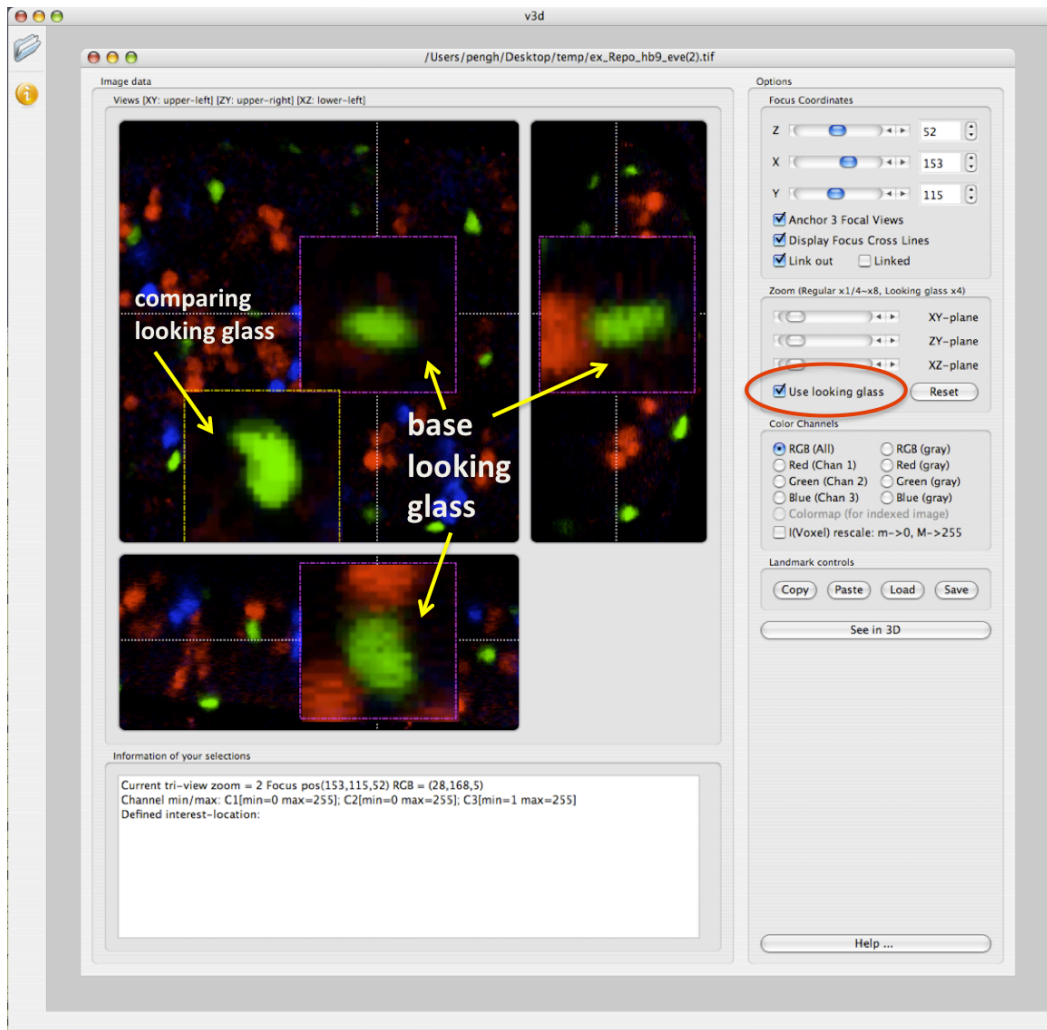
The **“link-out” and “linked” check boxes** can synchronize the focusing planes of multiple images. For example, as shown below, V3D can split color channels, - for the color RGB image two channels are separated (the third channel is not shown). When the “link out” checkbox is “on” for the RGB image, and the “linked” button of the two single channel images are also checked, these three tri-view displays are synchronized. In this way, a user can colocalize the image contents conveniently.



The **zoom controls** are easy to understand, - similar to a Google map, - when one of the XY, YZ, and ZX plane is zoomed in, a user can pan the image around using mouse. Note that V3D also allow the entire tri-view window is zoomed-in or zoomed-out, - which provides extra convenience. This can be done by pressing “1” (zoom-in) or “O” (zoom-out) keys. The current zoom is displayed in the text field below the tri-view. Two short cut keys “1” and “2” are also supported in V3D to go to zoom=1 and 2, respectively. For example, in the following example a user press the “2” key to zoom-in twice (i.e. double) the size of the above color RGB tri-view, and then select the right-hand zoom for XY-plane to be bigger than 1, and ZY-plane zoom smaller than 1. This provides good ways to visualize both global and local information of a 3D image.

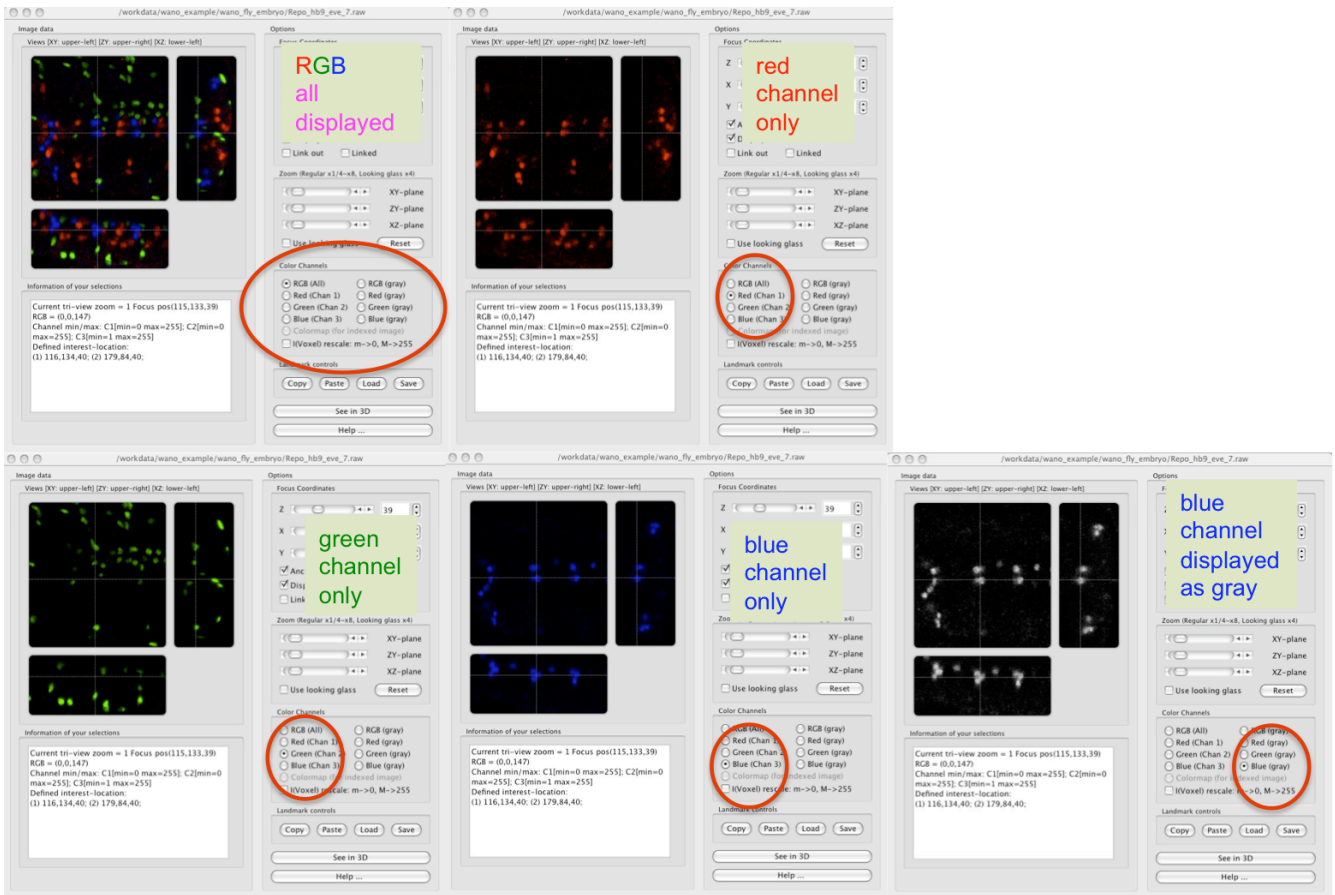


Another good way to zoom-in an image region of interest is to use the looking glass, which can be turned on by checking the respective checkbox right below the zoom controls. V3D designs a **special dual looking glass system: the base-looking-glass** is the three magenta boxes for all the XY, ZY, and XZ planes shown below; **the comparing-looking-glass** is the yellow box that smoothly transit in any of the three planes automatically, and provides a moving comparison of any image region against the base-looking-glass region. A user can change the location of the base-looking-glass by a right-click of the mouse, which change the current focusing point. The comparing-looking-glass moves around while a user moves the mouse anywhere in the image region. A user can also customize the size of the looking glasses by choosing “Adjust preference” in the main window menu.

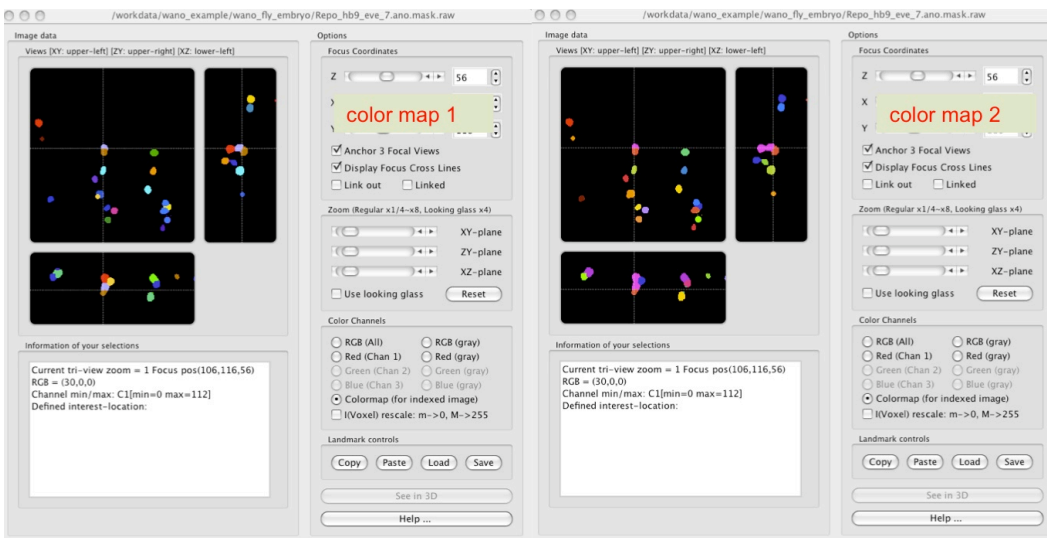


The tri-view main window has several different ways to display the color channels in an image. A user can click the **Color channel controls** to select displaying all the RGB channels together in color, or in grayscale, or one particular channel in color or in gray. The grayscale display is useful many times when one want to see the image content in a clearer way, as in many cases human being's eyes are more sensitive to black/white information than to colors. One example of the color display is shown below.

The minimum and maximum voxel values of color channels are displayed in the text information field of the tri-view. A user can select the control checkbox "I(Voxel) rescale: m->0, M->255" to scale the displayed intensity (not the real image data).



Colormap display is useful to differentiate neighboring pixel values that are slightly different, or to display indexed image. V3D allows one treat any single-channel image as an indexed image, and thus display different pixel values using different colors. There are three built-in color maps in V3D, - they can be switched from one to another by pressing “Shift-C” key. One example of an image displayed in different colors can be seen below. When the user is happy with a particular color map, and would like to see the same color scheme in the 3D viewer, this can be done by choosing menu “image/data” -> “convert indexed/mask image to RGB image”.



One central and powerful feature of V3D is its **3D landmarking and landmark managing capabilities**. There are many ways in V3D to define landmarks, -- a user can conveniently define landmarks in both the tri-view main window and the 3D viewer. Later in the manual there will be summary of landmarking, landmark management, and how to best make use of them. Here is a brief introduction how to define and use landmarks in the tri-view main window. Note that V3D sometimes will call a landmark as “marker”, - they are the exchangeable terms.

A user can press **“m” key (for “marker”) in the tri-view display**, - this will define a landmark at the current mouse focusing point (the intersection of the two cross-lines in the XY, YZ, ZX planes). A pop-up **landmark property window** as shown below will appear, displaying the

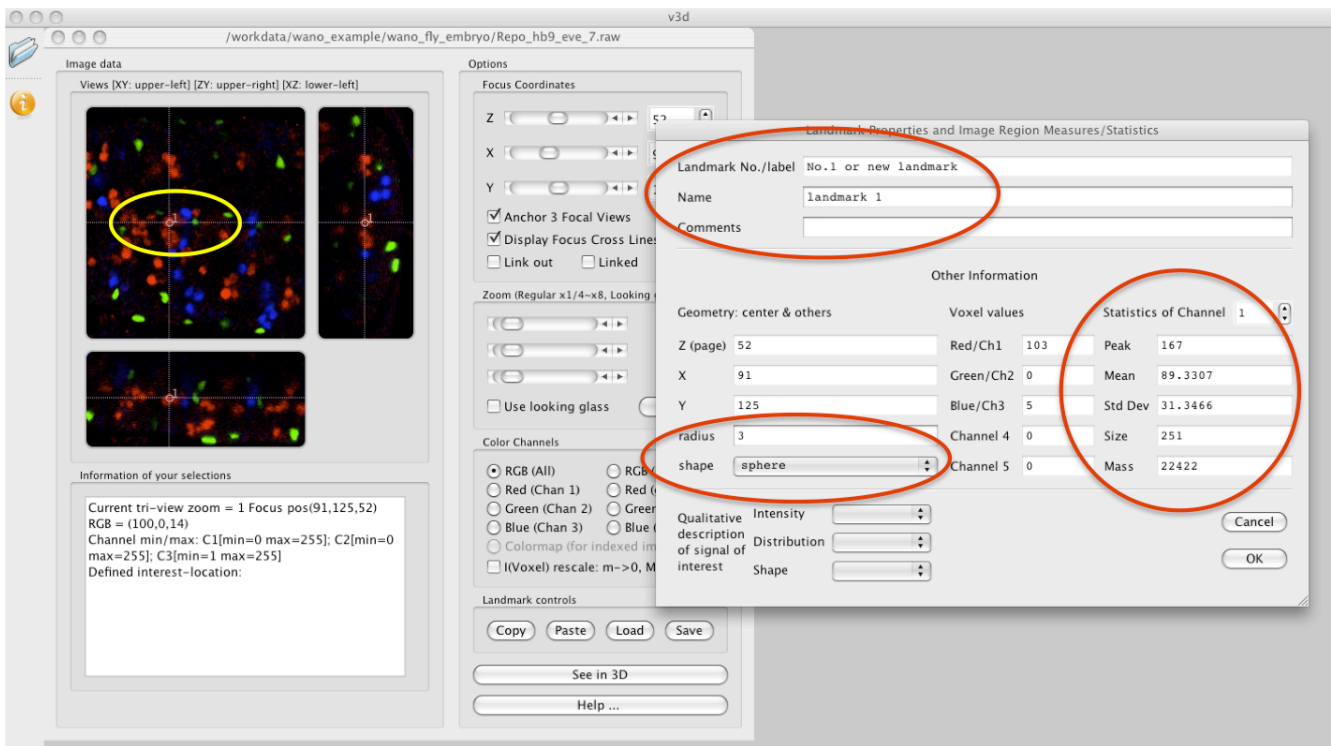
- order of this landmark,
- the default name (can be edited),
- user-defined comments (editable),
- the location (X,Y,Z) of the landmark,
- the radius (changeable) of a small 3D region surrounding the landmark (thus define the size and shape of the landmark),
- the shape of the landmark, can be sphere, cube, circle only in one of the 3D planes, etc),
- Voxel values (RGB, and additional channels if there are any) of the landmark location,
- Voxel/pixel statistics of the landmark region for separate channels, including peak intensity, mean intensity, standard deviation, size of the region (# of pixels), and mass (sum of all voxels).
- Other qualitative descriptor of the small image region associated with this landmark

The **landmark property window** is a useful tool for

- (1) recording user text-annotation of any location-related image content and**
- (2) measuring the pixel statistics of image regions.**

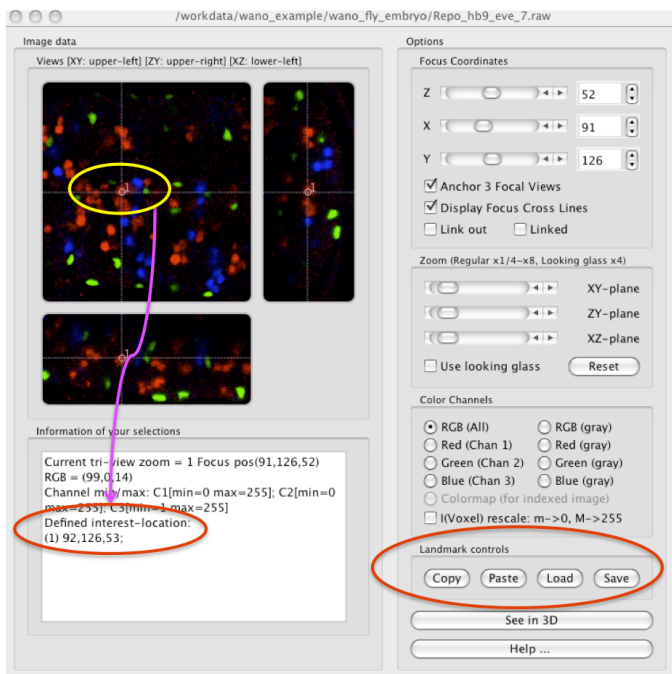
Note that these information can be conveniently save/reloaded/exported (explained below), thus this is a powerful tool to record and share quantitative information of the image content.

Apparently a user can use the landmark property window to measure the average and variance of a small region’s pixels. How about for the entire image? This can be conveniently done by setting the “radius” of the landmark to a big value (e.g. the largest dimension of the image), so that the statistics will be that of the entire image.

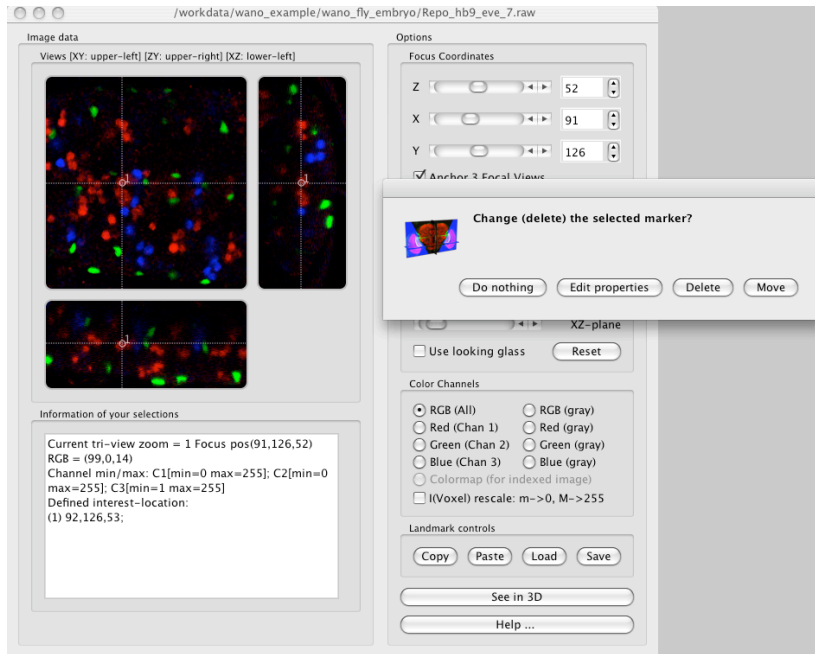


Once a landmark has been defined, i.e. after the “ok” button in the landmark property window is hit, the 3D location of a landmark will be displayed in the text information field of the tri-view, as shown below in the left-side red-circle.

All landmarks defined for an image can be copied and pasted from one image to another. They can also be saved and reloaded (the right-hand red circle below). The **landmark file has a default suffix “.marker”**, but it is just a plain text file with the CSV (comma separated value) format, - thus can be read/written/edited using any text editing programs.



In tri-view, a landmark can be accessed by double-clicking the mouse anywhere close to it (about 10 pixel distance should be fine). A pop-up window shown below will appear, asking if want to edit the landmark property (annotation, radius, shape, or just do more statistical measures), or delete the landmark, or to move its location. If choosing “move”, then the new location of the landmark can be defined by clicking the mouse primary key (normally the left-mouse button) while holding the “shift” key on the keyboard.



Usually a number of landmarks would be defined for an image via different ways. The **landmark manager** is a spreadsheet dialog that manages all the landmarks in the tri-view main window. To activate the landmark manager, the user can either press “Cmd-F” key on Mac (“Ctrl-F” key on Windows/Linux), or select the main V3D menu “Image/Data”->“Landmark”->“landmark manager”. It will look like the right-hand table in the following figure. With landmark manager, each landmark is displayed as a row in the spreadsheet, along with its

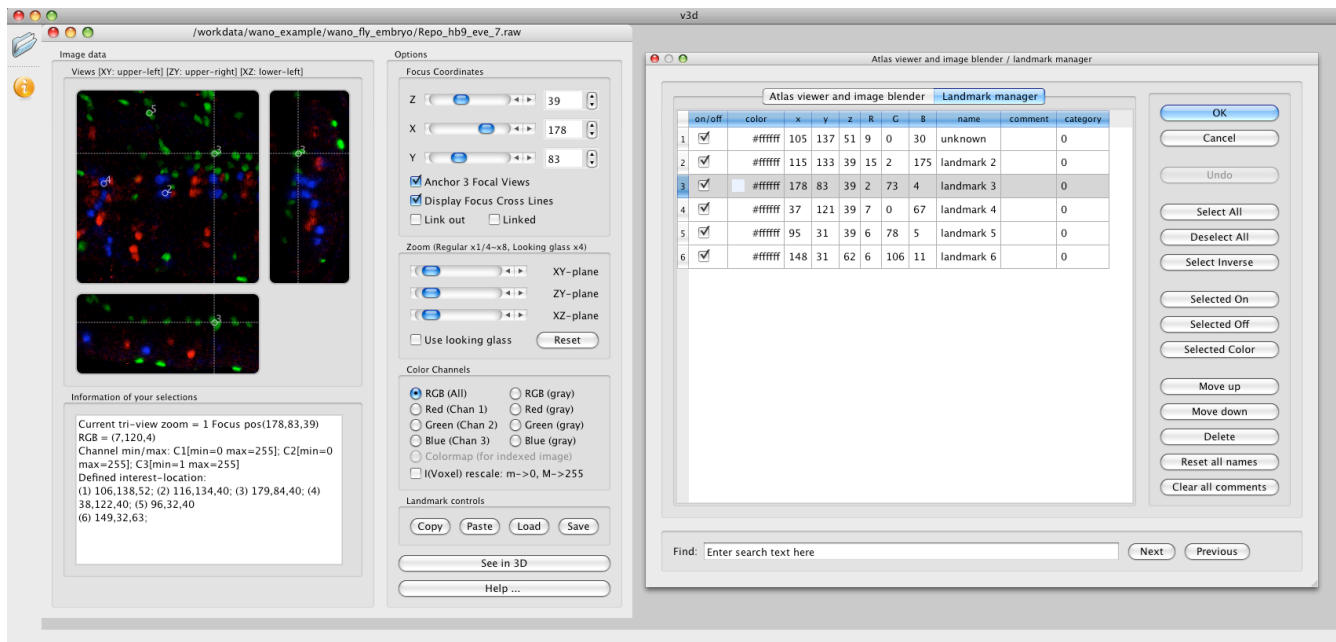
- on/off (display/hide) state,
- color,
- x,y,z locations,
- RGB value of the voxel of the landmark location,
- landmark name and comments, and
- category information which is also associated with landmark shape.

There are **three very useful features of the landmark manager**.

1. **Locating landmark in 3D image.** When a user click or select (via searching) a row, the focusing point of the tri-view display will move to the location of the landmark. Therefore one will always be able to keep track of a virtually unlimited number of landmarks conveniently. In contrast, it will be hard to eyeball the image to find where is the wanted landmark.

2. **Change landmark orders.** The order of landmarks is important for some applications, such as for 3D image registration, or image standardization. Landmark manager lets one to change the order of landmarks easily, by clicking the “move up” or “move down” buttons.
3. **Search landmarks** based on their names/comments.

Landmark manager also lets one reset all names, or comments; it also enable one to select multiple landmarks and do some batch processing such as hide those selected all at once.

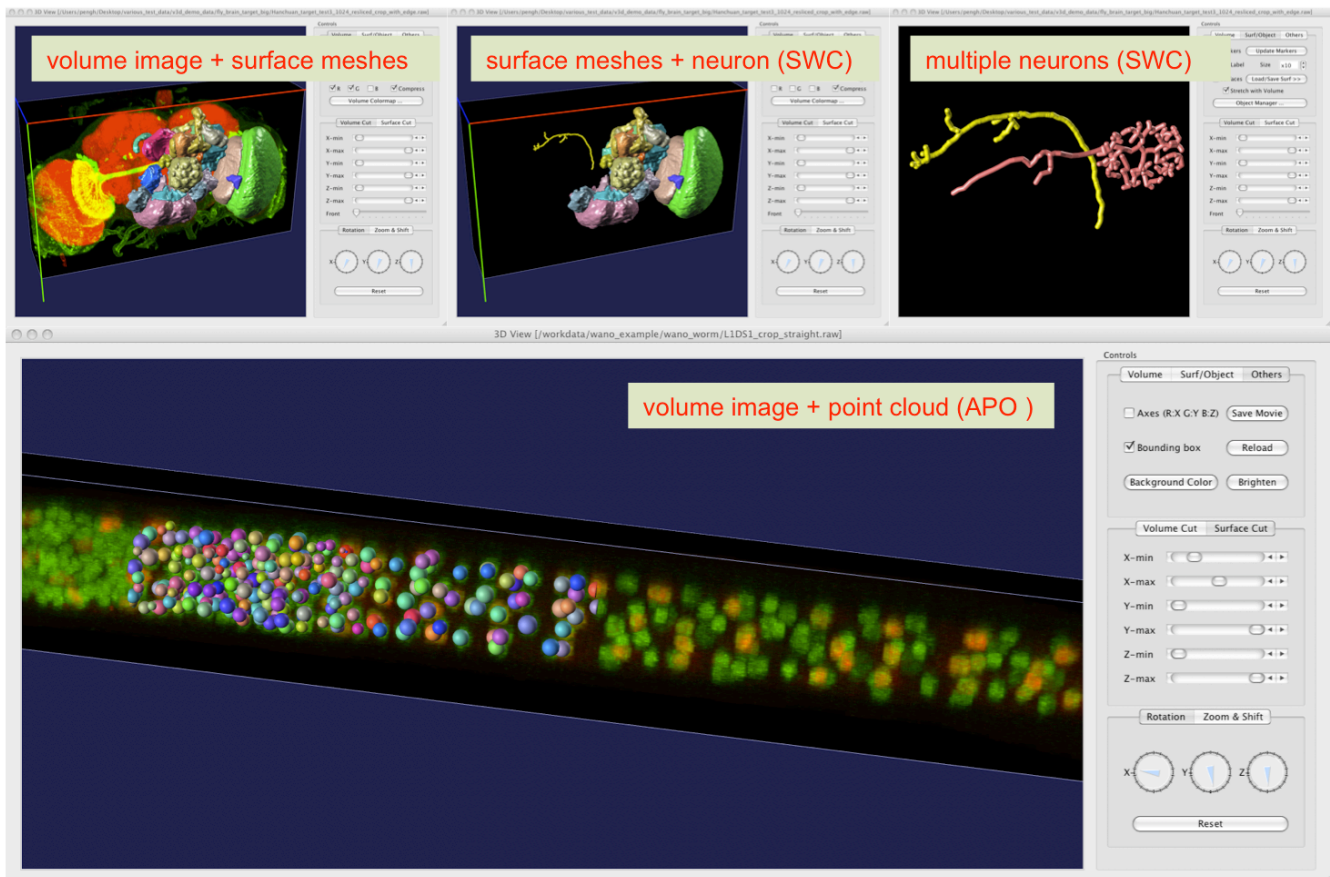


2.3 How to use the 3D Viewer

The 3D viewer of V3D is able to render both 3D volumetric and surface data at the same time. For volume image and surface object files, the 3D viewer will be opened in different ways.

- **For image file, the 3D viewer can be launched by clicking the right-bottom button of a tri-view window, or by pressing “Cmd-V” key on Mac (“Ctrl-V” on Window and Linux).** It can also be configured to open automatically when an image is open (just check on the option in “Adjust preferences” menu in the main menu). Note that due to the consideration of compatibility of most computers with various video cards, by default the 3D viewer will always use at most 512x512x256xRGBA size volume. That is, if the image data in the tri-view window has any XY dimensions bigger than 512, or Z dimension bigger than 256, the 3D viewer will downsample the volume data. If the image data has a smaller volume than 512x512x256xRGB, than no downsampling will be used. However, a user can force the 3D viewer to use the original volume without downsampling by pressing “Shift-V”. The risk is that if there is no sufficient amount of video memory in the video card, OpenGL library (not V3D code) may crash in the video memory, which is hard to catch. Thus unless absolutely necessary, or you have a really good video card with a lot of video memory (e.g. >1G video memory), we recommend not using “Shift-V” (see Development Plan section for some solutions discussed).

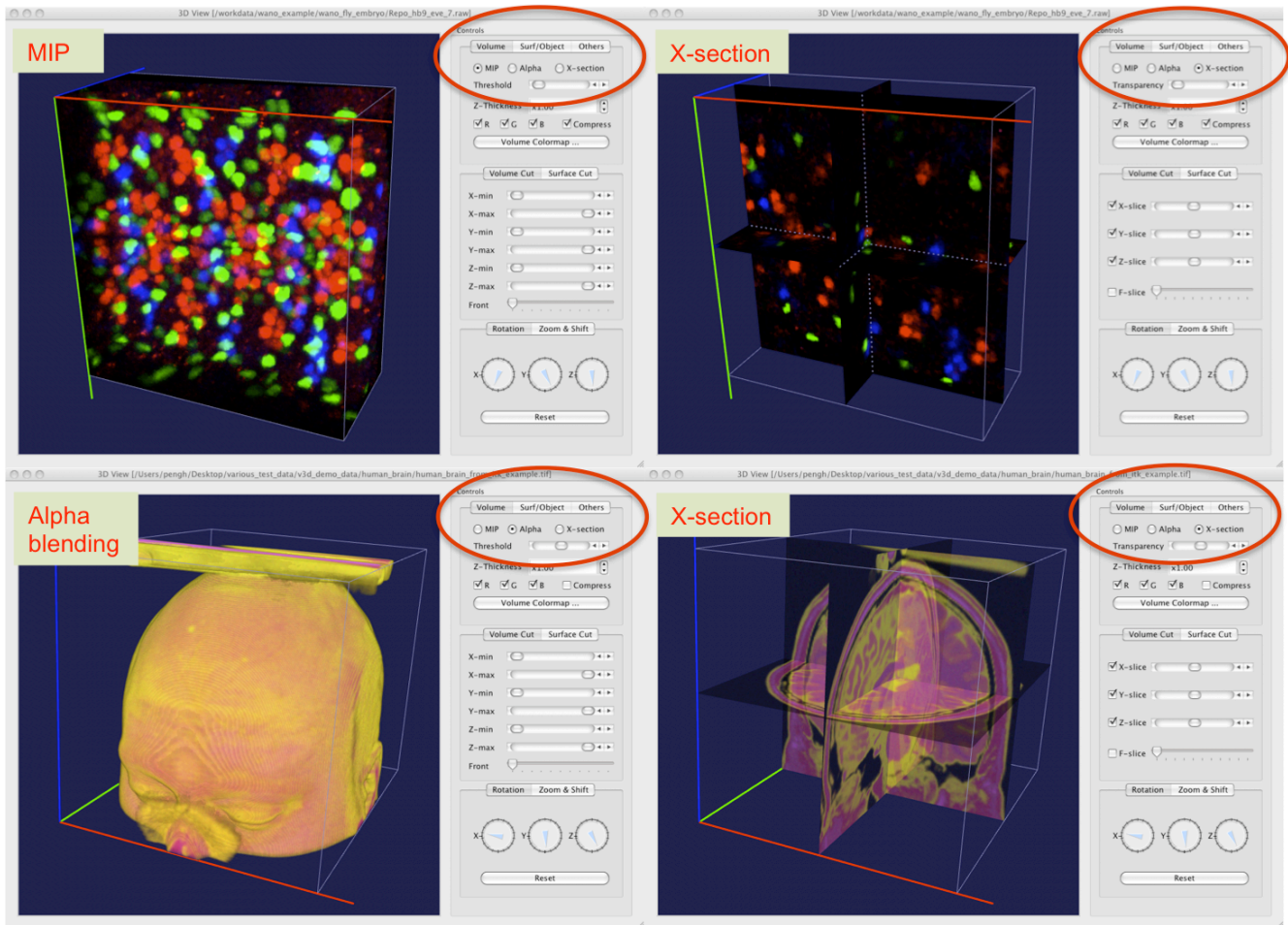
- **Surface file types (see Section 1 will be directly opened in a 3D viewer window** instead of first opened in a tri-view window. They can also be opened in an existing 3D viewer window, - in this case, the new image objects will either replace the existing image objects (if they are label surface meshes or point cloud or markers), or get displayed side by side with the existing image objects (for SWC neurons or other relational/graph data that are be described using SWC format). If the existing 3D viewer contains only volume image data, then surface objects are overlaid on the image data, with their relative depth relationship adjustable (explained later). The following figure shows examples of displayed (a) volume image + surface meshes, (b) surface mesh + SWC neuron, (c) three SWC neurons, and (d) volume image + APO point cloud. Many other rendering combinations can be conveniently generated similarly.



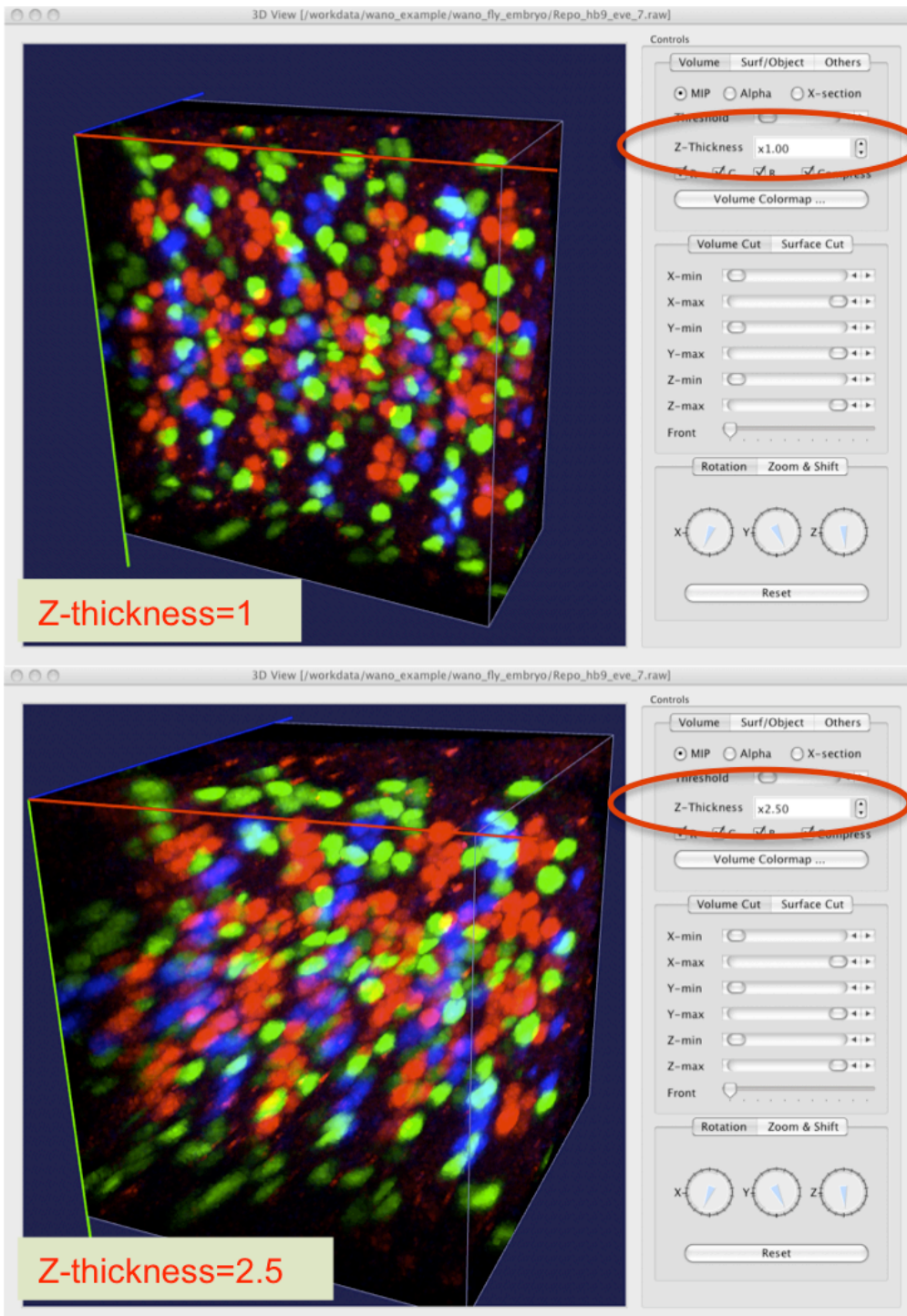
The 3D viewer has **three modes for rendering volumetric image**, namely **MIP (maximal intensity projection)**, **alpha (alpha blending)**, and **x-section (cross-sectional views)**. A user may choose the best one for a 3D image, or switch between them to better understand the image content. Usually MIP is better for sparse image contents such as cells or neurons, and alpha blending is better for visualizing the surface information in image, thus better for dense tissue. X-section view is similar to the tri-view, but with a better stereo effect. Several examples of different rendering modes are shown in the following picture. A user can switch between these modes by clicking the right-side “Volume” tab control of the 3D viewer.

For MIP and alpha modes, immediately below the three radio boxes of “MIP/Alpha/X-section”, there is a **“Threshold” slider**. This is a global threshold to remove the low intensity image

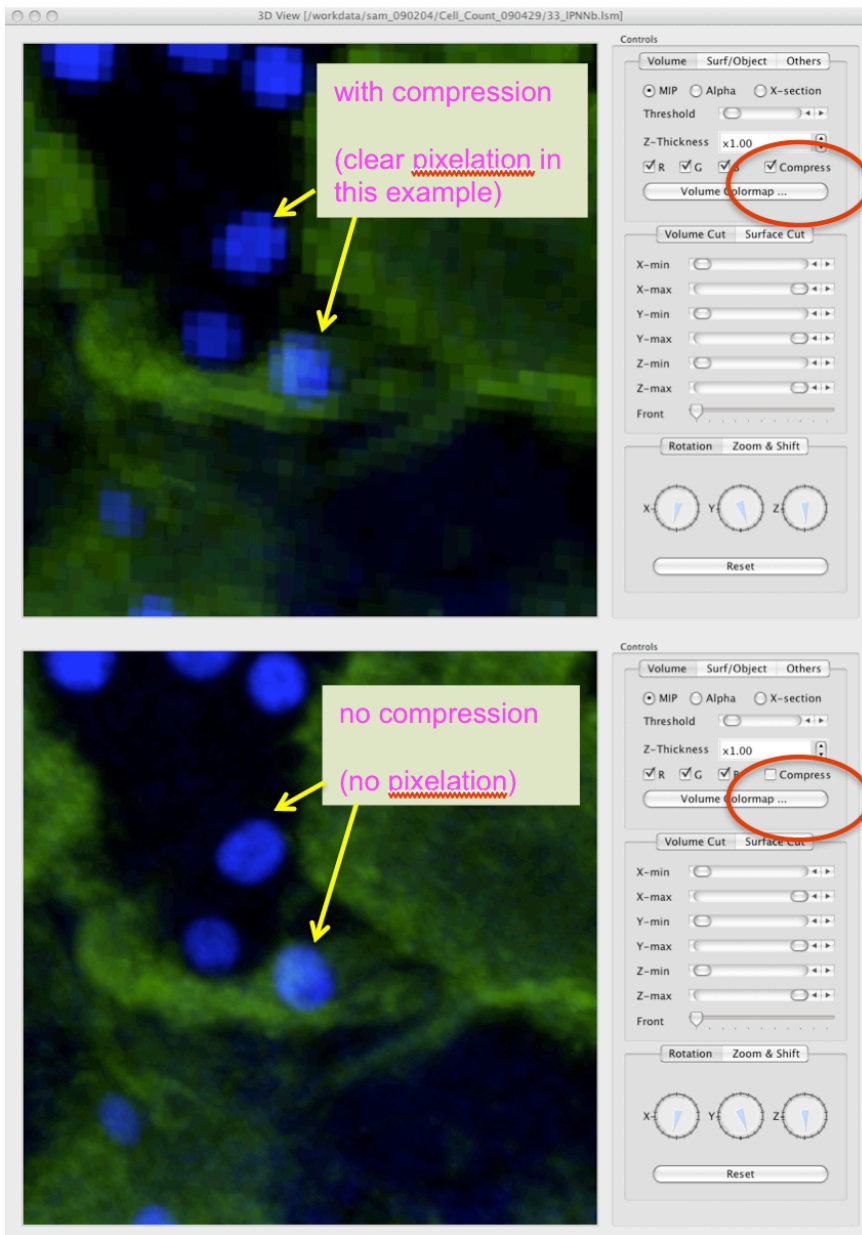
contents, thus can make some rendering visually clearer. For X-section, a different slider called **“Transparency”** replaces the location of “Threshold”. This controls the transparency of the three orthogonal cutting planes in the X-section display.



For microscopy images, often an image is anisotropic, in the sense that Z-direction pixel size is different from (and normally bigger than) that of XY dimensions. There is a **Z-thickness control** at the top-right of the “Volume” tab. V3D by default always assume the XYZ pixel size is 1,1,1. A user can select a floating multiplying factor (up to x10) so that the Z-pixel size is real, so that the data appears more realistic (instead of flattened). The following is an example without and with change of the Z-thickness. Warning: enlarging the Z-thickness will also enlarge the time to accomplish the 3D rendering, thus reducing the speed of the 3D viewer. Thus **the user should not abuse using this function and subsequently inappropriately complain V3D is slow!**



The **“Compress” checkbox control** in the “Volume” tab indicates whether or not use the lossy compression of the OpenGL in the video card. With compression, the actually data volume is smaller, thus the 3D rendering speed is faster. But sometimes the image quality reduces significantly when the “Compress” is turned on, - as shown in the following example where the pixelation is visible, especially when the image display is much zoomed-in . For many other cases, the difference of image rendering quality is minor.

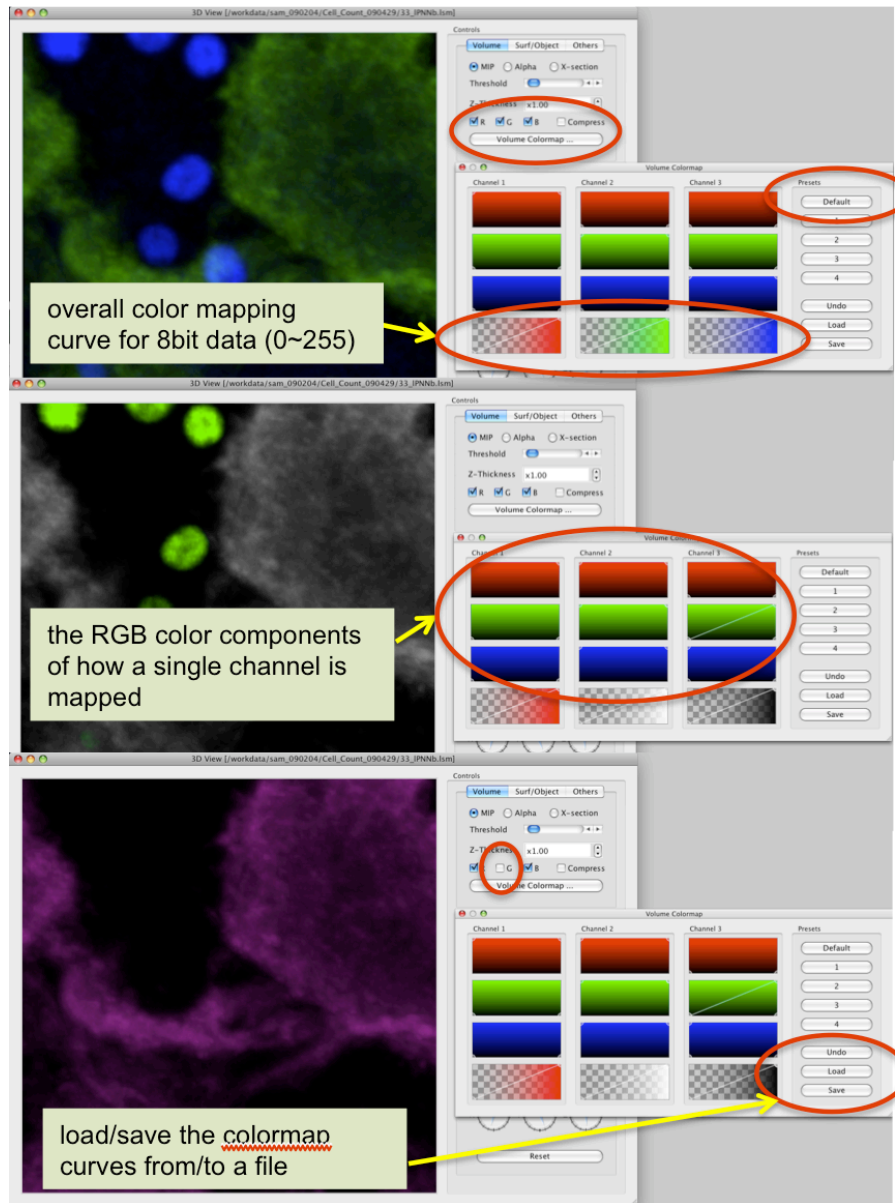


A related image-rendering issue is that **2D texture mapping versus 3D texture mapping of OpenGL**. V3D by default always use 2D texture mapping, which in many cases produce faster speed and better visualization effect. But a user can press “Cmd-T” on Mac (“Ctrl-T” on Window/Linux) when the 3D viewer is activated, thus switch between the 2D and 3D texture mapping modes.

For clarity, let’s use the term “view port “ to stand for the visible display/rendering area of a 3D viewer. The **Color controls of volume images** in the 3D viewer consist of two parts: “**Volume colormap ...**” button that controls the color mapping from the data to the view port, and the **RGB color channel visibility** checkboxes of the view port that controls if particular color of the view port will be visible.

When there is an image displayed in the view port, pressing the “Volume colormap ...” button will activate a “**Volume Colormap**” dialog in the following figure. This dialog box lets a user adjust

how voxel intensity values in an individual channel are mapped to 3-component RGB colors. Since the 3D viewer renders 8bit data, which has a range [0~255], thus for each channel there are 4 mapping curves: from top to bottom there are RGB color component curves that indicates how a voxel intensity value, say, 63, is mapped to a color, say (255,0,255), which is a purple color; the last curve at the bottom is a global weighting factor (with a range [0~1]) that determine the strength (brightness) of this mapped color. For example, when the weighting factor for voxel intensity 63 is 0.5, then 63 is mapped to color $0.5 * [255, 0, 255] = [127, 0, 127]$.



default colormap:
 1st channel → red
 2nd channel → green
 3rd channel → blue

changed colormap:
 2nd channel → grayscale (Black/white),
 3rd channel → green.

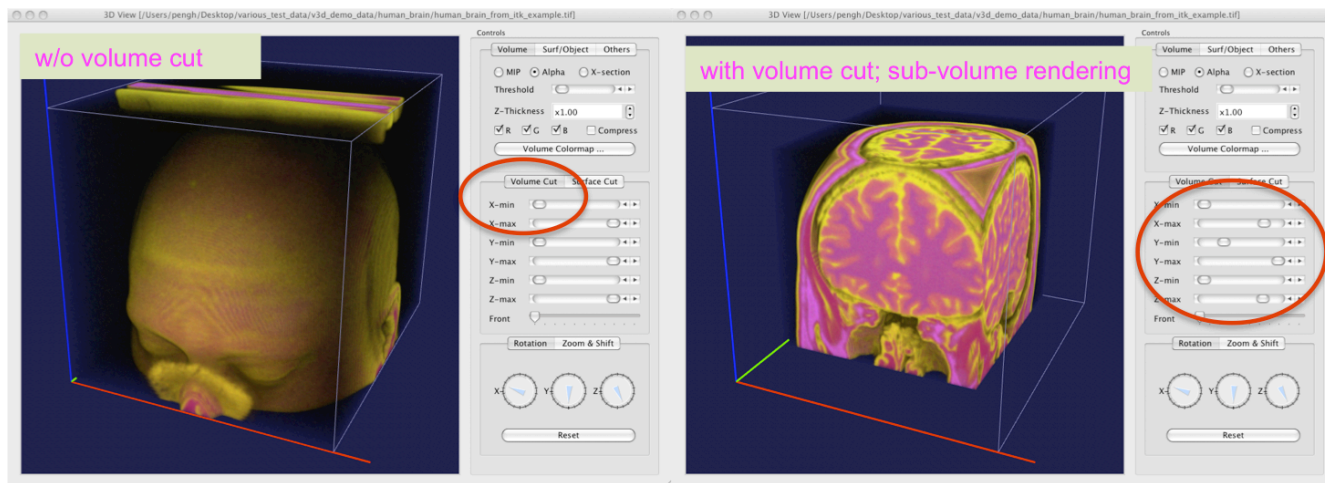
the view-port green channel is turned off:
 2nd channel → grayscale w/o green color, thus become purple
 3rd channel → green w/o green color, thus invisible.

The “Volume Colormap” dialog uses line segments to indicate how the color ranges are mapped. The default mapping range is [0~255] (also can be set by pressing the right-side “Default” button in the dialog), indicating low intensity value will mapped to dark color, and high intensity to bright color. This can be changed. A user can left-click the mouse (i.e. primary mouse button) to add break points that define new line segments (color mapping ranges). Holding the left-mouse button while moving the mouse can be used to adjust the actual mapping value. Right-click the mouse (the secondary mouse button) on a break-point can be used to delete this existing break

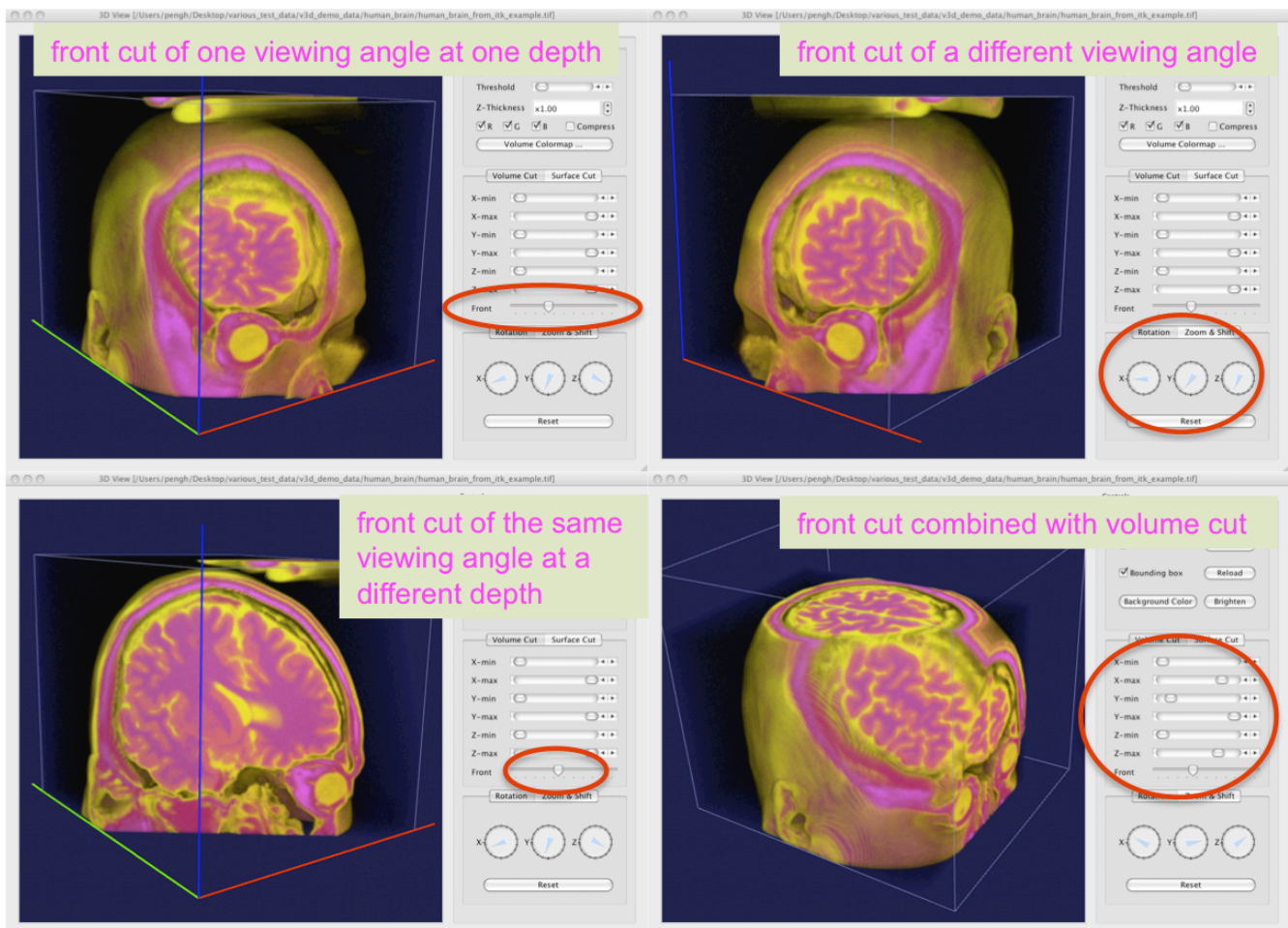
point. Additionally, there are several preset color map schemes, such as display all channels to grayscale, reverse color, etc. The adjusted colormaps can also be saved to file (with “.v3dcm” suffix), and reloaded again later. The load/save functions can be done by pressing the two right-bottom buttons in the dialog.

One concrete example is shown in above figure. In the top sub-figure, the volumetric data displayed using the default colormap has the normal green and blue color, as the 2nd channel is mapped to green and the 3rd to blue. In the middle sub-figure, the color mapping of the 2nd channel is adjusted to be displayed as grayscale, i.e. RGB components are equal to each other, and the 3rd channel is mapped to green color. This is done by adjusting colormaps in the “Volume Colormap” dialog. In addition, in the bottom sub-figure, the “G” checkbox, i.e. green color visibility checkbox in the “Volume” tab is turned off, that means the green color of the view port is turned invisible. Thus the mapped grayscale color of the 2nd channel loses the green component, and thus is seen as purple (red+green+blue -> gray; red+blue->purple). Accordingly, the 3rd channel is mapped to green color, but since the green component in the view port is not displayed, thus we don’t see the 3rd channel information in the bottom sub-figure.

The “**Volume cut**” control in the 3D viewer lets one visualize a sub-volume. Sliding the one of the min/max sliders in XYZ dimensions will set a range of the sub-volume to generate the visualization. In another word, the MIP and alpha blending will only compute the sub-volume. This is useful to visualize the dark or deep image content that can be occluded by the bright/shallow image content. One example is seen below.

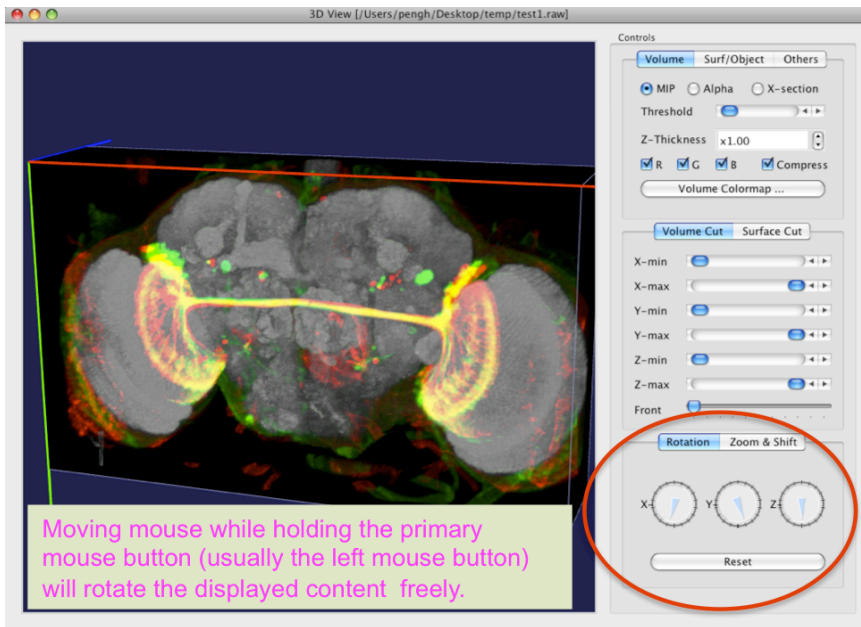


Front cut is a useful feature of V3D, which displays the image content projected to a plane which is perpendicular to the viewing rays of a user. A user can also select which depth along the viewing ray is inspected in the front cut plane, by adjust the slider right below the “Volume cut” min/max range sliders. One example is shown below, where the front cut is used to visualize the image content from an arbitrary angle and depth.

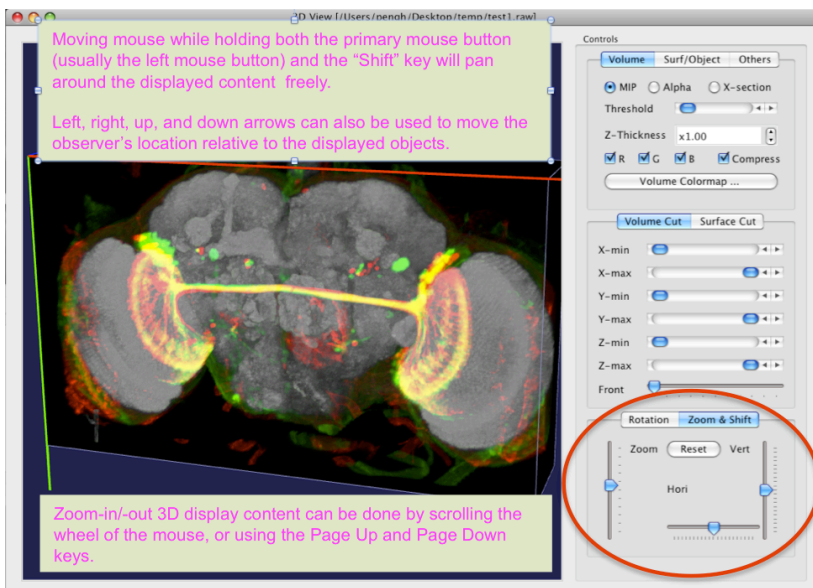


When the X-section display mode is selected, the “Volume Cut” tab changes to have four sliders of X-slice, Y-slice, Z-slice, and F-slice (for Front cut). A user can also turn on/off particular cutting planes, such as only displaying the Z-slice while keeping all other cross-sectional views off.

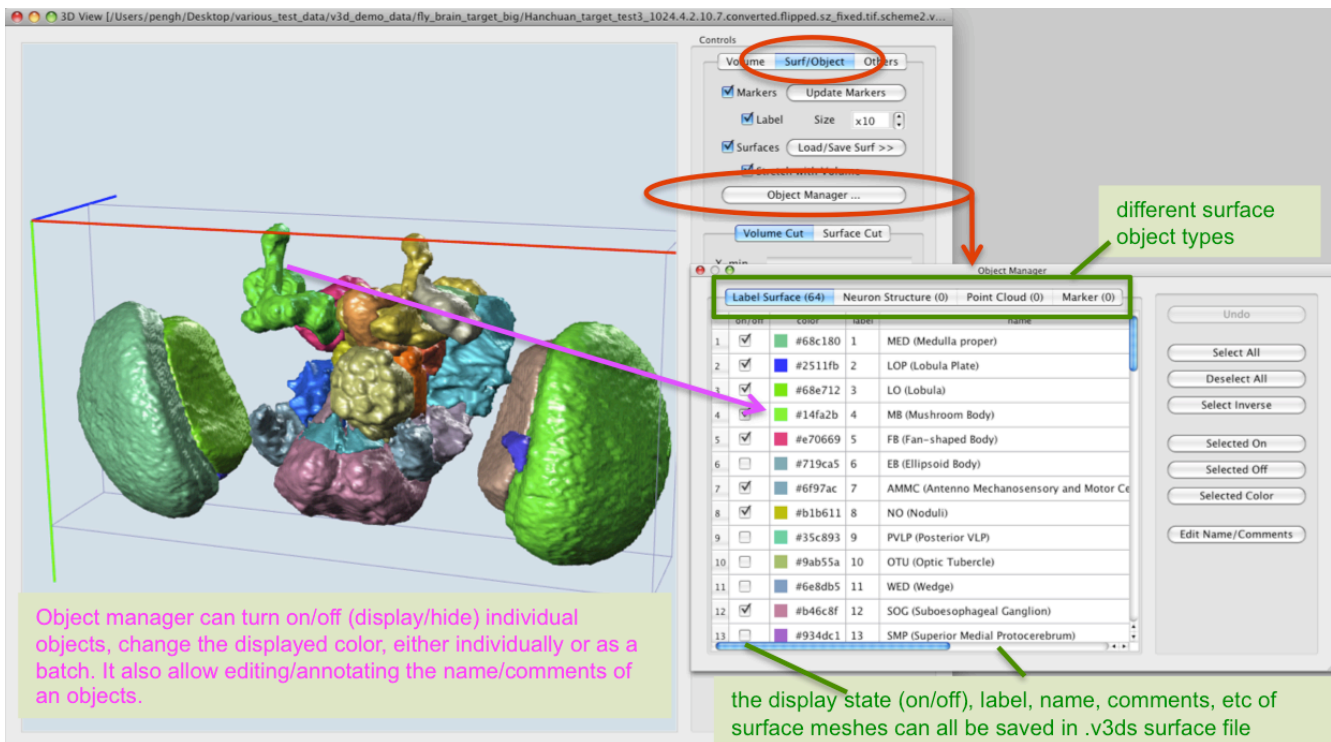
Freely Rotating the 3D displayed volume image and surfaces is simple, - just hold the left mouse button (i.e. primary mouse button) and move it around, -- the 3D display will rotate accordingly. A user can also control how the display is rotated around particular axis, one of X,Y,Z by using the dialing pads in the “Rotation” tab at right-bottom corner of the 3D viewer. The “Reset” button will reset the X,Y,Z rotation angles all to 0. One example is shown below. Note that a volume image and surface objects, if they displayed in the same 3D viewer, will normally get rotated together, in a synchronized way. There is one way to lock the scene and rotate only one surface object freely (explained below), but at this moment V3D only supports doing that for the SWC neuron objects.



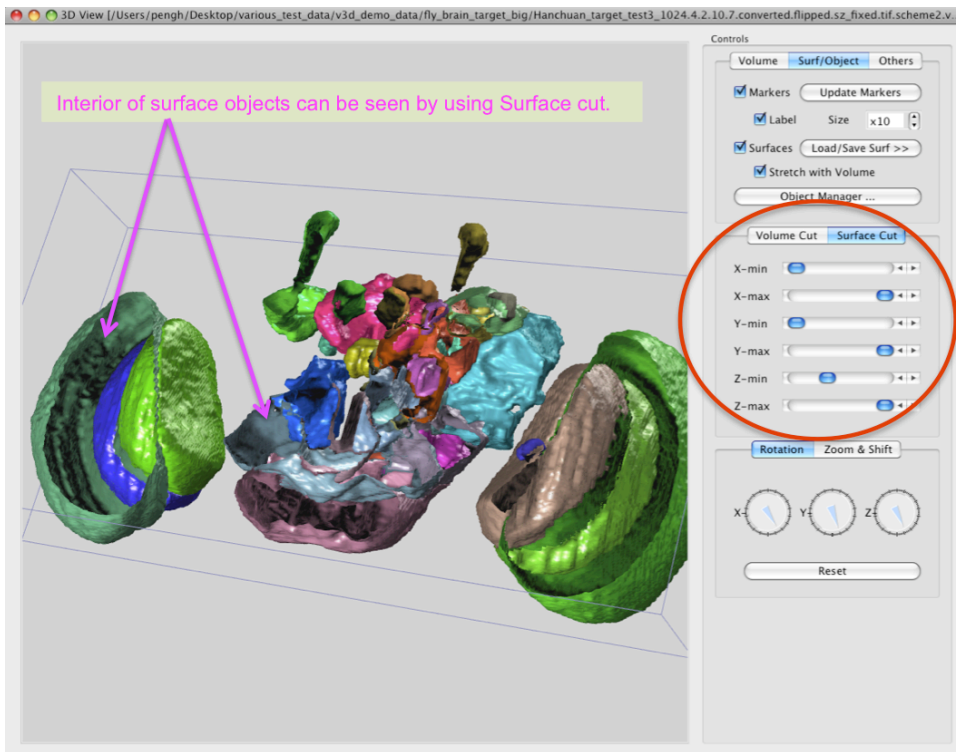
Freely zooming and panning around the 3D displayed volume image and surfaces is also simple, -- when a 3D viewer is the current active window, just scrolling the wheel button of the mouse, or using “Page Up” and “page Down” keys, or just drag the 3D viewer to be bigger/smaller, will zoom-in or zoom-out the display; use the “left” (←), “right” (→), “up” (↑) and “down” (↓) keys on the keyboard will move the display around in the four directions. As shown in the following figure, a user can also use the buttons in the “Zoom & Shift” tab at the right-bottom corner of a 3D viewer to control how the display is zoomed or moved. Panning around the displayed image/surfaces can also be done by holding “Shift” key and the primary mouse button (usually left mouse button) and move the mouse. Note that this “holding Shift + primary mouse button” method will move the displayed image/surfaces in the scene, whole the aforementioned “holding ←, →, ↑, ↓” method will move the observer’s location, thus these two methods will produce reverse effects, -- but both are correct! Also the “Reset” button will change zoom=1 and horizontal and vertical shifts to be 0.



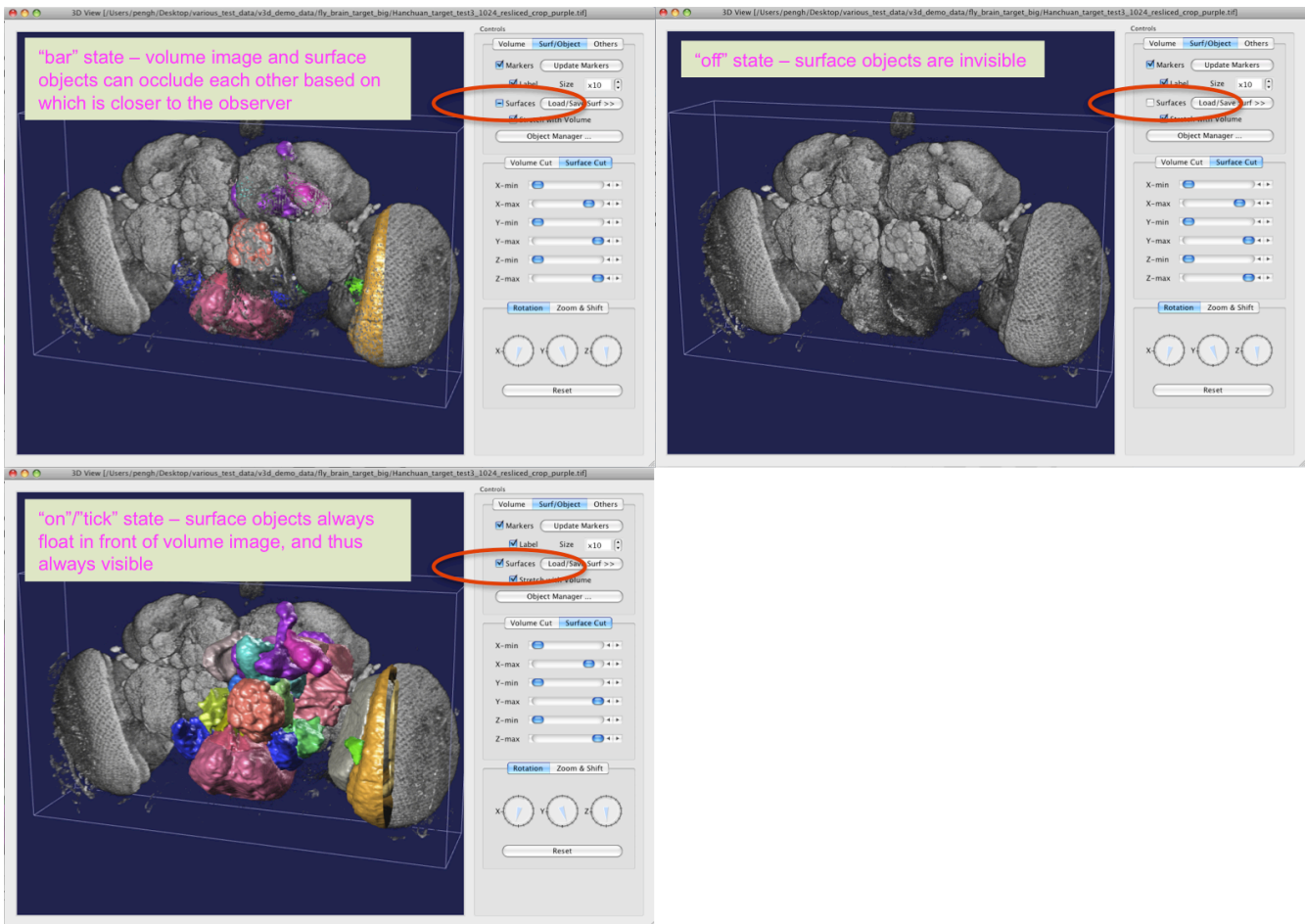
V3D supports the major types of possible surface objects, which are surface meshes (for irregularly shaped objects), point cloud (each point is a sphere in the 3D space), relational data that can be described using a graph and stored in the SWC format (which is usually used to describe the digitally reconstructed 3D neuron structures, but can be used to describe any graph), user-defined landmarks (markers), user-defined 3D line segments, etc. The 3D viewer of V3D manages all these objects using an **“Object Manager” dialog**, similar to the figure shown below, which can be accessed by clicking the “Object manager ...” button in the “Surf/Object” tab at the top-right corner of the 3D viewer. A user can turn on or hide some of these objects, or change their color, or annotate their names/comments easily in the object manager. For example, a user can double-click the color square of each row in the Object Manager spreadsheet and thus **change the color of the selected object**. Many objects’ color can be changed all at once to the same. This is good way to highlight some of these objects. Note that the Object Manager can also be directly launched by following the pop-up menu information when right-clicking (secondary mouse button) a surface object.



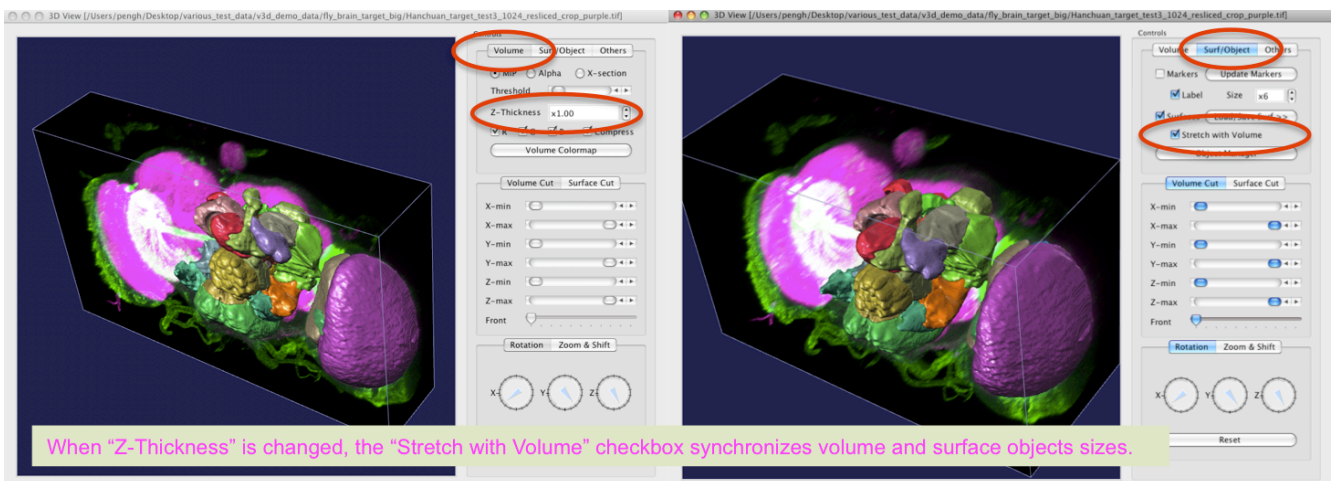
Surface objects can be visualized in different ways. Individual objects can be either displayed or hidden, -- this can be done using the Object Manager discussed above. They can also be cut, through the **“Surface cut” tab** in the right-middle of the 3D viewer. This is similar to the “Volume cut” for volumetric image display, -- however this useful sometimes for one to see the interior of a surface object. It is also useful to create different overlay effect when both volume and surface data are displayed together. One example is shown below.



Instead of (1) turning on/off individual surface object, and (2) cutting surface objects, a third way to visualize surface together with a 3D image is to adjust their relative depth relationship, i.e. which would occlude which. This is done through a 3-state checkbox “Surfaces” in the “Surf/Object” tab. When the **“Surfaces” checkbox** is checked off, all surface objects (except markers which are treated differently) will be hidden; when the checkbox is turned on as a tick, all surface objects (except markers) will always float in front of the display volume image, even the entire scene is rotated 180 degrees. This **floating display method** is a very useful feature of V3D, especially for the proofreading of digital reconstruction of image objects side by side with the original image data, because it lets a user always be able to see all surface objects defined. The third state of “Surfaces” checkbox is a bar, which means the volume image and surface objects are displayed using their true 3D relative locations, -- the objects/image content near to an observer/user will occlude those far away from the observer. In this case, the combination of volume cut and surface cut will be useful to visualize the data. Several examples of are shown below.

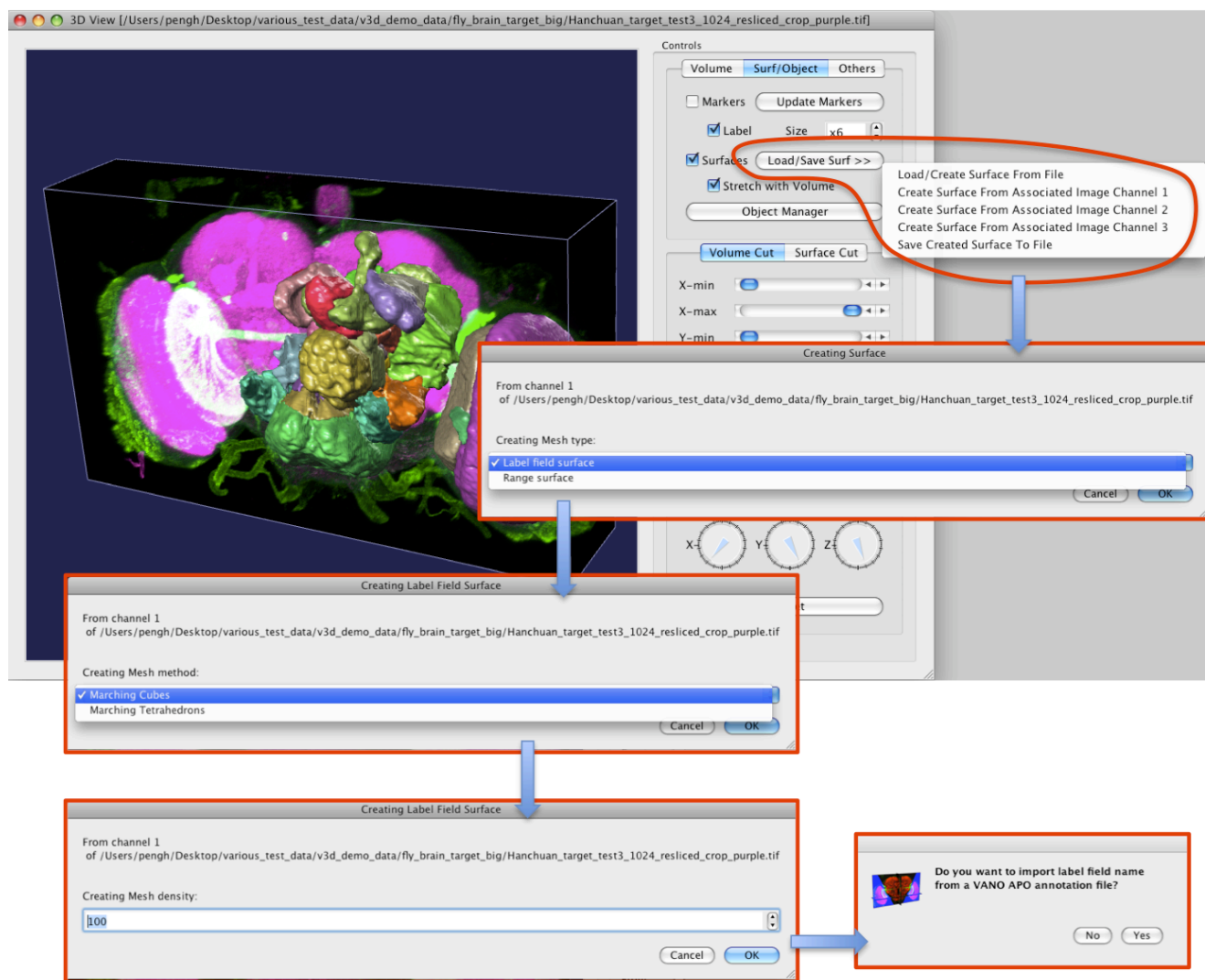


The **“Stretch with Volume”** checkbox in the “Surf/Object” tab allows the surface objects (including markers) change their displayed coordinates when a user change the Z-thickness spinbox in the “Volume” tab. The figure below shows the effect, which is useful to synchronize the image content and surface objects.



The **“Load/Save Surf>>”** button in the “Surf/Object” tab allows one to load/save surface objects files, or create surface-meshes from an image file or from the currently displayed volume image. It will display a pop-up menu showing all these options, as demonstrated in the following figure.

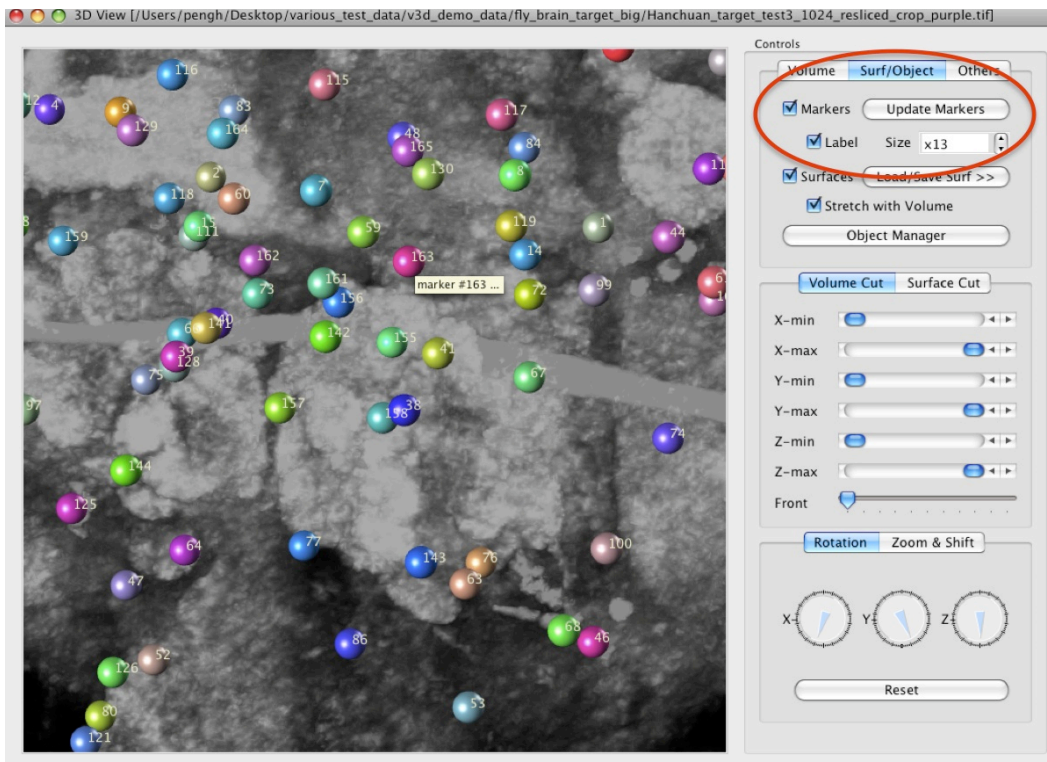
Loading an existing surface file is simple, and equivalently can be loaded into a 3D viewer by just dragged and dropped into the 3D viewer. If a surface object file (except the marker file) is dragged and dropped into a V3D main window area, but not in a 3D viewer window, a new 3D viewer will open to display that surface object. If the user chooses to **create surface meshes** from an image file (TIFF format, or any image file format supported in V3D), the 1st channel will be used to create the surface mesh. Or, V3D allows one to create surface meshes from one of the RGB channels, as indicated by the pop-up menu in the example below. The surface objects can be saved by selecting the last menu item in the pop-up menu, but can also save it by right-clicking (secondary mouse button) and following the information there (explained later).



Creating surface meshes in V3D can be accomplished by following a mesh-generating wizard dialog (shown as the series of dialogs in above figure). When selecting the 2nd to 4th menu items of the pop-up menu in the above figure, the source image associated in the tri-view main window must be available, -- that is the tri-view main window of the current 3D viewer should not have been closed. Assuming it is still open, the mesh-generating wizard will ask several questions about mesh type is **“Label field surface”** or **“Range surface”**. “Label field surface” means that a different intensity value in the channel indicates a different image object. This gets the name because normally such image is produced through manual painting as a label field. “Range

surface” means that user need to specify the min and max range of the intensity, -- image objects are defined as those within the range of the intensity. The wizard also ask mesh creating method to be “March Cubes” or “Marching Tetrahedrons”, -- normally March Cubes will be good. Next, mesh creating density needs to be determined. This value means how many grids will be used to partition an image’s dimensions, -- of course the higher density the better mesh quality, -- but it takes much longer time to generate the mesh, and also the resultant mesh has a bigger size. Finally the wizard asks if want to import names of the label fields from a VANO APO annotation file, which is a CSV file recording users’ annotation of image objects. Unless the user is using VANO software (<http://hanchuan.peng.googlepages.com/vano2>, developed by Hanchuan Peng), the answer be selected as NO, and the user can always manually annotate the surface object names later.

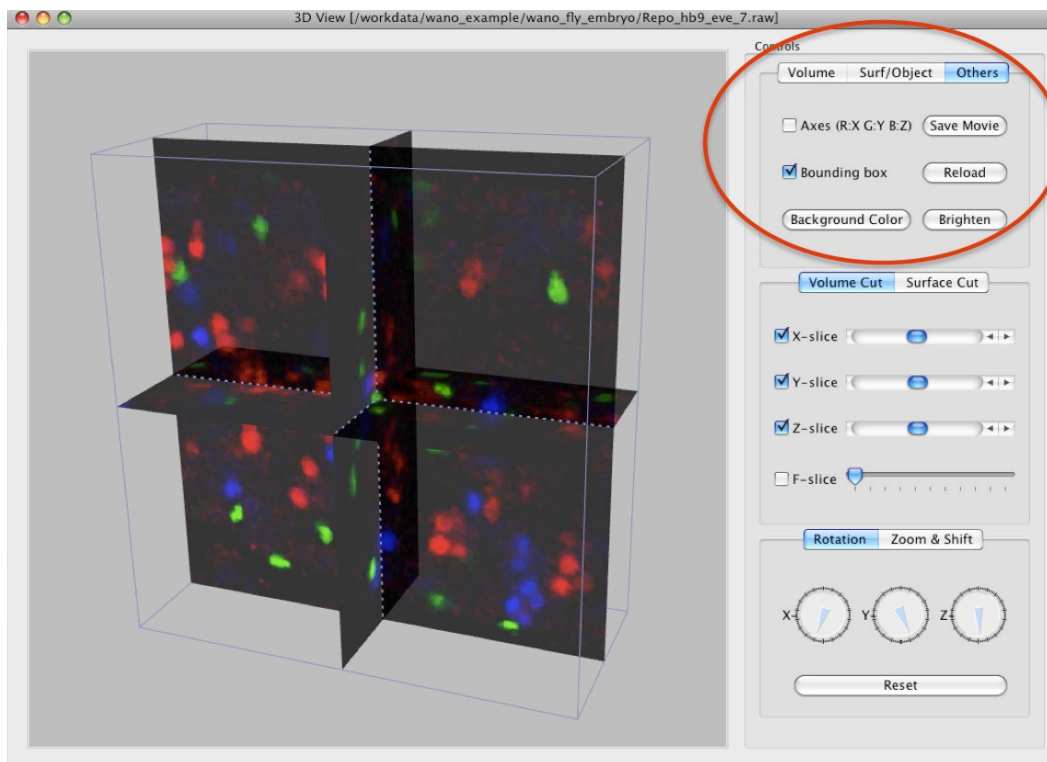
The 3D viewer manages landmarks in a way similar to how it handles general surface objects. Because markers are often directly defined in a tri-view main window, a user can click the **“Update Markers” button** in the “Surf/Object” tab to copy all markers from the tri-view main window; a marker file (“.marker”) can also be directly dragged and dropped into a 3D viewer to update all the markers. Similar to the floating display of other surface objects, markers can also be set be always floating on top of the image volume display, or using the real relative location information, or being hidden, by checking the **3-state checkbox “Marker”**. A marker’ order (called “Label”) can also be displayed close to it, -- this is controlled by a **“Label” checkbox**. The display size of a marker (not the radius of a marker in the Landmark Property window!) can also be adjusted by changing the value in the **“Size” spinbox**. All these buttons can be seen in the following figure.



The **“Others” tab** in the top-right corner of a 3D viewer contains buttons, as shown in the figure below: **“Axes (R:X G:Y B:Z)”** for if or not displaying the three XYZ axes, **“Bounding box”** for if or not displaying the bounding box of the image volume and surface objects, **“Background Color”**

for selecting a canvas background color in the view port, **“Brighten”** for brightening the entire rendered scene in the view port (including the background), **“Reload”** for reloading/refreshing all contents in case the rendering is messed up (this function is rarely needed according to a number of tests so far), and **“Save Movie”** for generating movies of the rotating scene of the entire 3D rendering.

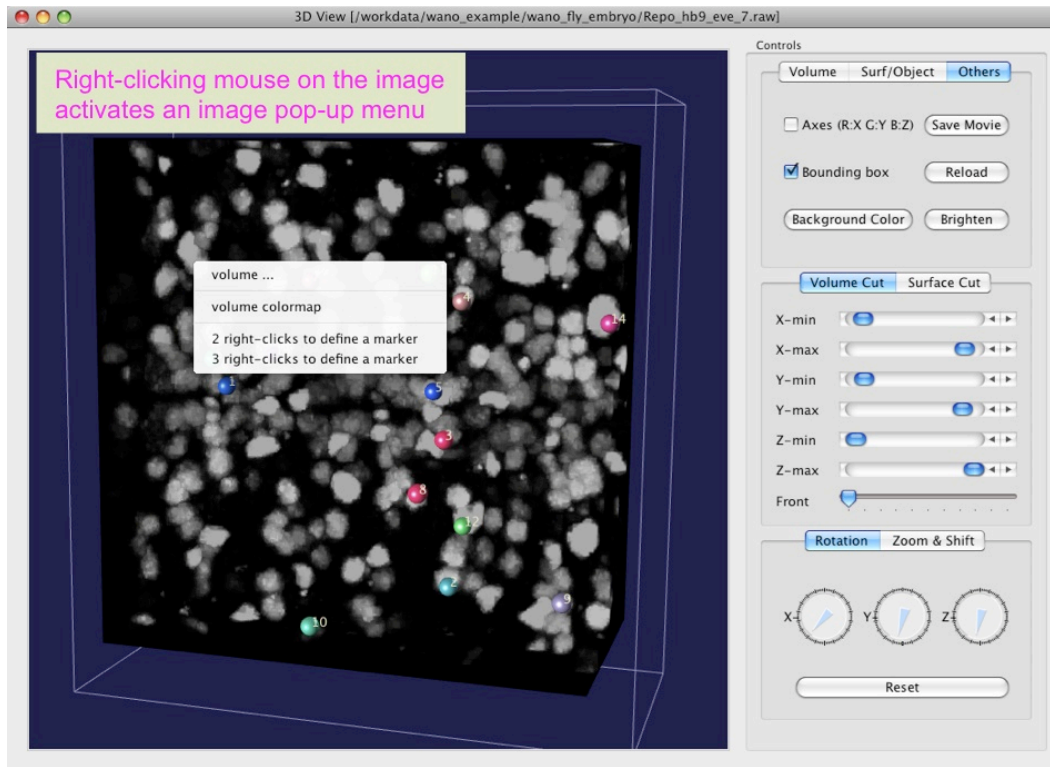
The **“Save Movie”** function will launch a wizard asking what is the rotating sequence the user likes, and what is the number of frames, as well as the folder these individual frames to be saved to. V3D currently only saves individual movie frames, instead of generating an already-encoded movie, -- there is pro and con to do this, -- but it is really easy to use other tools such as ImageJ to import these movie frames and re-save to a movie file in QuickTime (.mov) or AVI format. Providing the direct access to individual movie frames give a user more flexibility to compose complicated movies by combining frames from different sequences.



One feature of the 3D viewer is that it provides a number of interactive functions via **pop-up menus in the view port**. These pop-up menus can be activated by right-clicking the mouse (the secondary mouse button). Depending on what is clicked and what add-on modules have been provided to a user, different menus will appear.

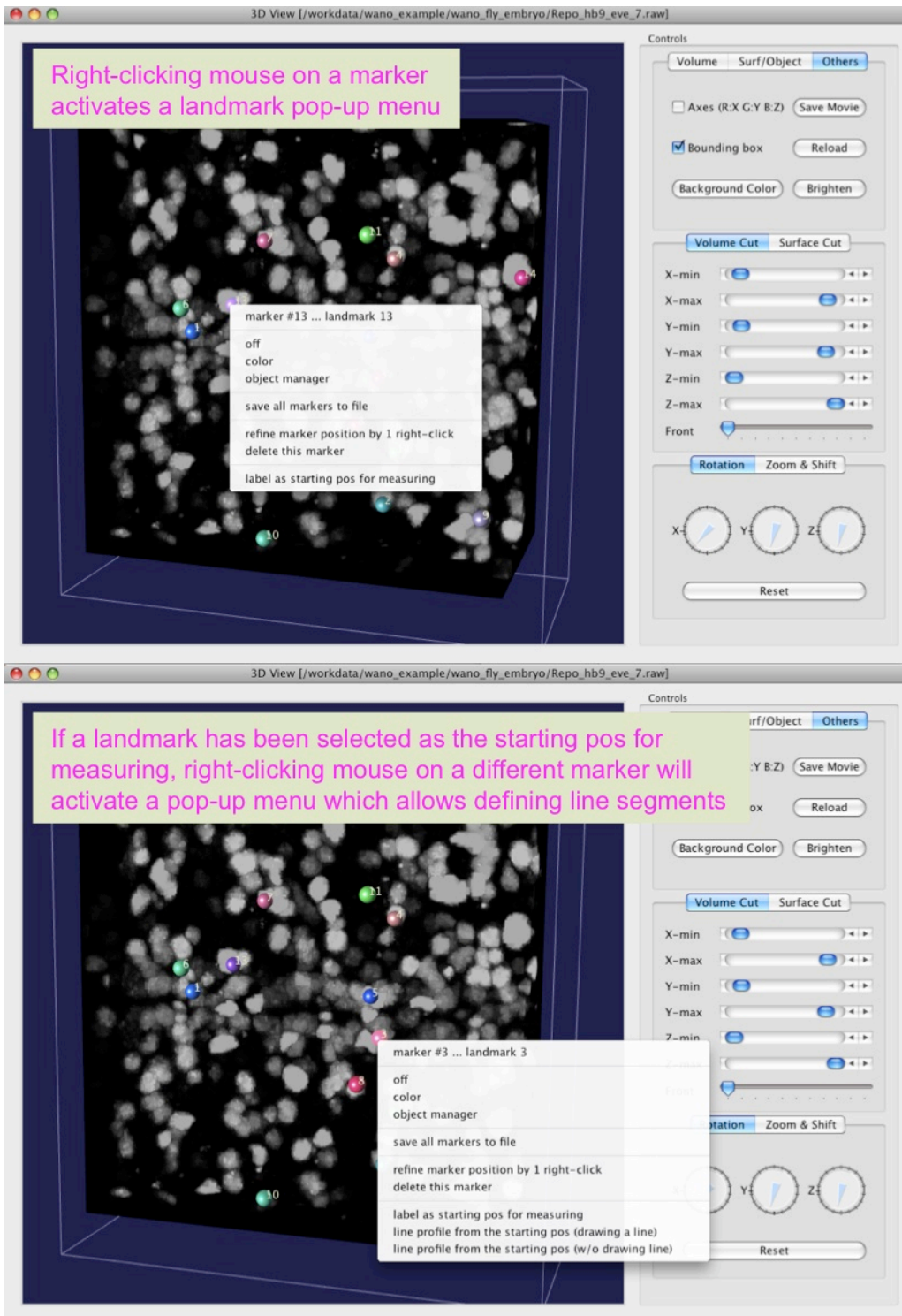
The following figure shows that the **volume image pop-up menu**. A user can choose to access the volume image colormap dialog explained above. A user can also **define landmark directly in the 3D space**, via the “2-right-clicks to define a marker” or “3 right-clicks to define a mark”. When the user chooses 2-right-clicks, the user needs to click the right mouse button at the location the marker needs to be defined, and then hold the left mouse button to rotate the volume an angle, and then right-click the mouse again, then a marker will be define in the 3D space directly. The idea is that these two right clicks define two straight lines along the observer’s viewing direction of the 3D object, but at the locations of clicked locations. Then a least square

error estimation of the nearest point of the two straight lines is calculated and used as the location of new marker. The option “3 right-clicks” is similar to the 2 clicks case, except that it adds one more straight line and thus the estimation is more reliable. **Note that this “2 clicks” function is very useful and basically enables a user to directly access 3D image contents.**

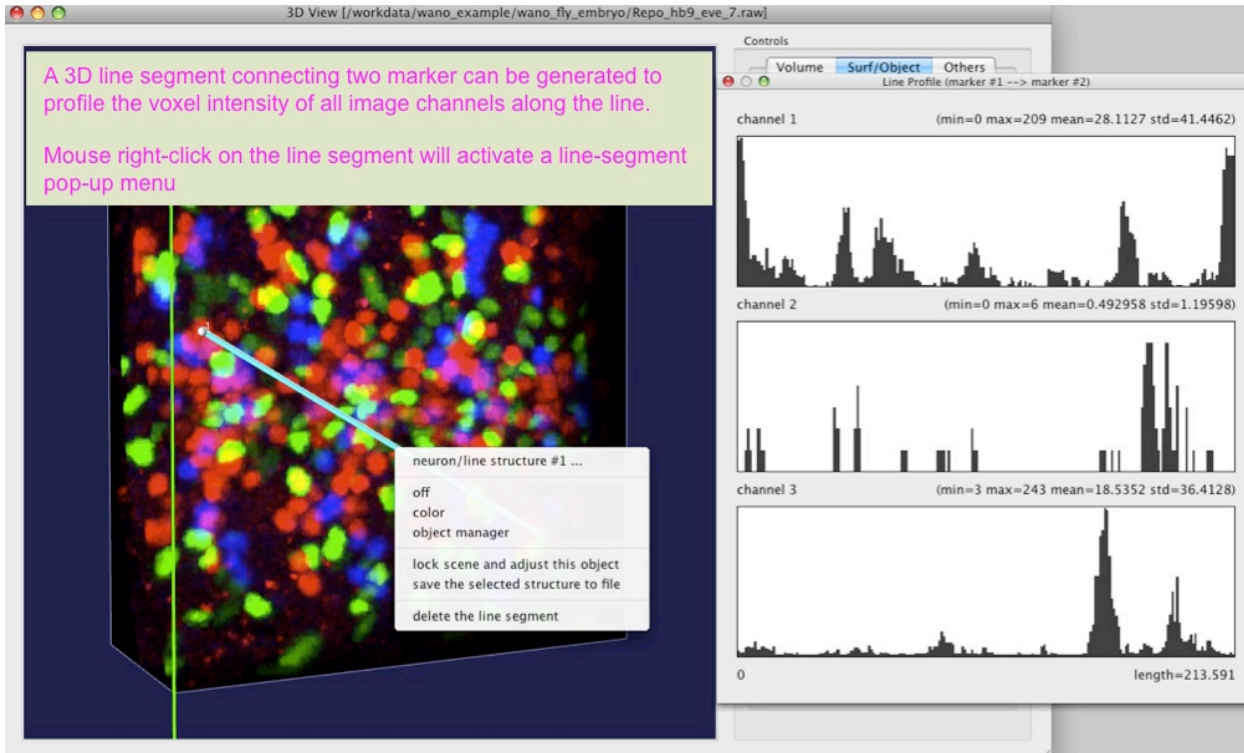


When there are landmarks displayed in the 3D viewer, right clicking the mouse on any of them will activate a different **marker pop up menu**. Depending on if a marker has just been selected as the starting position for quantitative measuring (or neuron tracing), the pop up menu will be a little different, as shown in the figure below.

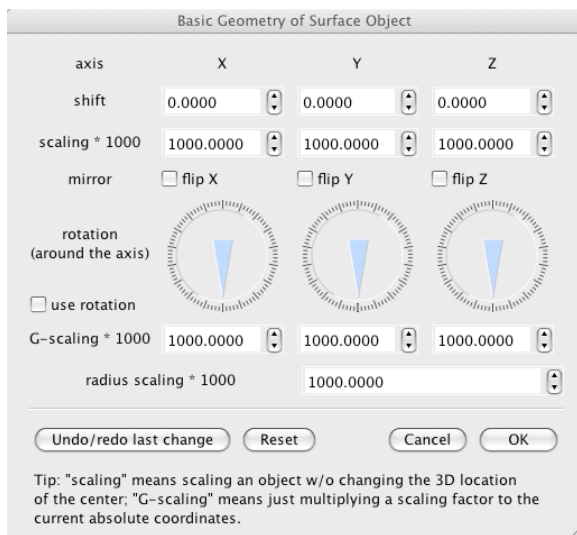
The common parts of these two pop-up menus include a row which shows the **basic information of the marker order and its name**, options to turn it off (hide), option to change displayed color, or option to access it in the object manager. Clicking the basic information row will open the **landmark property window** which lets a user see all information of this marker and also quantitative measuring of local image region around this marker. A user can also choose to save all markers to a file, or delete it. The last option in the pop-up menu, if this is the first marker selected, is to label it as the starting position for measuring. In this case, if a second marker is right-clicked, two more menu items will appear in the pop-up menu: one is **“line profile from the starting pos (draw a line)”** and the other is **“line profile from the starting pos (w/o draw line)”**. Selecting these two options will open a line profile window (see the figure after the one below), which displays the line intensity profile of all image channels from the starting position (the first marker) to the ending position (the second marker), along with the statistics such as min, max, mean, and standard deviation.



The generated **3D line segment** is another type of surface objects. Currently it is managed using together with another type of structural data, i.e. SWC neuron, because both of them are essentially relational data that can be described using a graph. In the future version of V3D, they will be separated managed. However, it can be noted that the line segment object is useful for quantitative measuring of image content. In the figure shown below, a user can define a line segment by connecting two markers, and thus produce a **Line Profile window** showing the length, intensity profiles of different image channels, as well as statistics of these profiles.

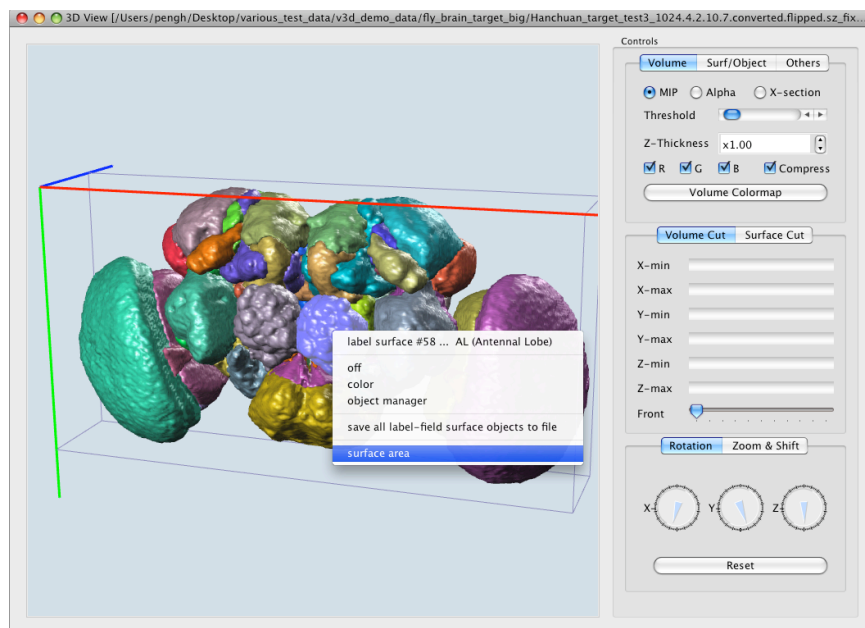


Right-clicking the mouse on the line segment will activate a **line segment pop-up menu**, as shown in the figure above. Selecting the first option “neuron/line structure ...” will let a user to annotate this line segment and see some associated properties; a user can also choose to hide (off) this line segment, or change its color, or manage it in the Object Manager, or save it using a SWC format file, or delete the line segment. One function that is useful is to “lock scene and adjust the object”, which will display the following **Basic Geometry of Surface Object** dialog to scale, displace, rotate, flip, adjust the thickness of the line segment.

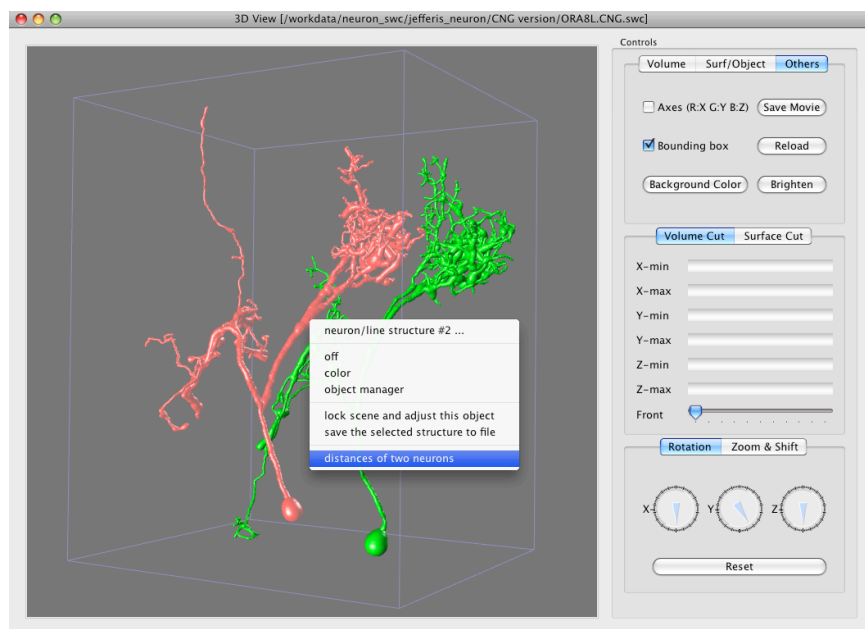


The **pop-up menu of label surface meshes** is similar, -- as shown in the figure below it lets a user annotate label surface name/comment and see other information, or turn a surface mesh object off, or change color, or access all surface objects using Object Manager. It also let the user

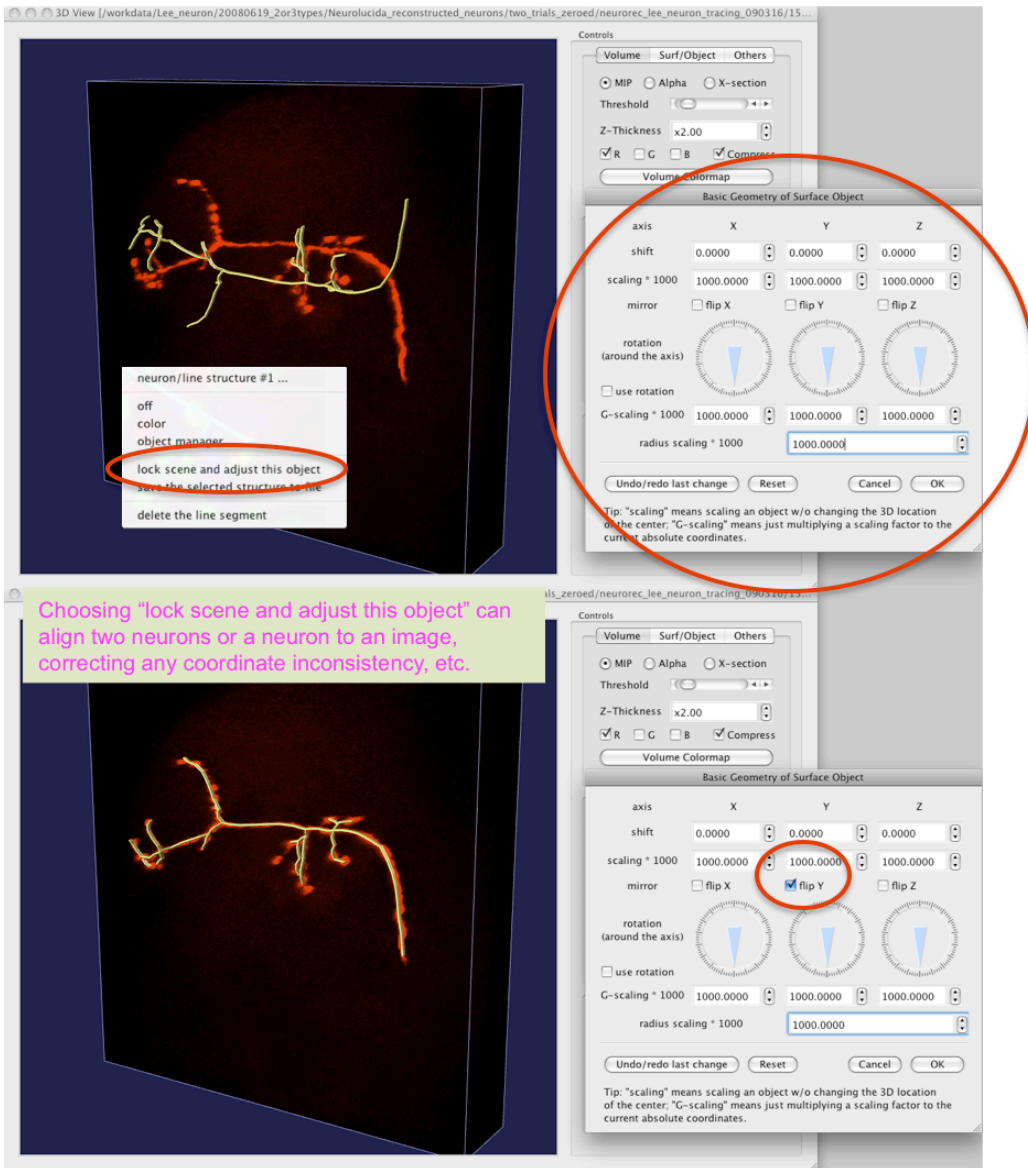
save label surface meshes to a file, which is normally a .V3DS file that can also contains the display status information. It also has a menu item to display the surface area information.



The **pop-up menu of neurons** as shown in the figure below lets a user annotate neuron name/comment and see other information such as number of branches, total length, total number of nodes. This pop-up menu also allows turn off (hide) a neuron, change the color of a neuron, or all neurons using Object Manager. It also let the user save a neuron to a file, which is a SWC format file. It also has a menu item to compare the **“distance of neurons”** which tells how similar of the selected neuron to all the remaining neurons currently displayed. Another very useful function, especially for neurons and the related digital reconstruction from 3D images, is the **“lock scene and adjust this object”** menu item explained below. **(Note that if you have got a version of V3D with the add-on module of 3D neuron tracing, the neuron pop-up menu will have more options.)**



For comparing different neuronal structures, or just proof-read the digital reconstructed 3D neuron structure with the original 3D image, it often needs to move one neuron (or other surface object) from one place to another, or apply certain geometrical transformation (e.g. affine transform). The **“lock scene and adjust this object”** menu item in the neuron pop-up menu is suitable for these purposes. In the example shown below, the originally reconstructed neuron has a flipped coordinate, compared to the 3D image. The dialog **“Basic Geometry of Surface Object”**, which is activated when pressing the “lock scene and adjust this object” menu, makes it very easy to flip the neuron along a direction, scale it, and perform other geometric transformations.



3. Main Menu and Icon Buttons

3.1 Main icon buttons

V3D has been intentionally designed to have **only TWO icon buttons**, one is “open” an image and the other is “information/help”. These two buttons are shown at the left side toolbar of the main V3D window.

3.2 Main Menu

If you find some of the main menu is empty (e.g. “Process”) , it is because you are using a special release version of V3D. You can contact Hanchuan Peng if you need particular image analysis modules or plugins to see if they are currently available.

4. Tests, Known Problems & Bugs of V3D

4.1 V3D Test Results on Various Computer Platforms

V3D’s cross-platform compatibility has been tested on a variety of laptop/desktop machines. The following machines & operating systems can run V3D including 3D visualization modules without a problem.

- Windows: XP, Vista
- Mac: PowerPC G4/G5, OS X 10.4 (Tiger)
- Mac Pro: Intel CPU, OS X 10.4 (Tiger), OS X 10.5 (Leopard)
- Mac Air: Intel CPU, OS X 10.5 (Leopard)
- Linux: Redhat 64bit, Fedora 64 bits

V3D has been successfully tested on many different video cards; the following is an out-of-date subset as of Sept 2008:

- NVidia GeForce 7900GTX
- Mobile Intel® 945GM Express Chipset Family
- Quadro FX 1700
- ATI Radeon 300
- NVidia GeForce Go7300
- GeForce 8600M GT (ROM v3212)
- GeForce 8800GT
- ATI Radeon X1900
- ATI Radeon HD2600
- NVidia GTX 280
- GeForce 7800GT
- GeForce 8600M GT (ROM v3175)

V3D has also been tested on virtual machines or X Window tunneling. For VMWare virtual machine, it runs without a problem, except the OpenGL code is slow because it is pure software implemented. For X Window running from a Linux machine and displayed on a Mac, the running is very smooth, except when exiting V3D there would be a X Window crashing sometimes (not consistently appear), - the same phenomenon is always seen for other QT examples running using the same configuration. Thus it is likely not a V3D bug; instead, it may be due to some problems of X Window and QT.

4.2 Known problems and bugs

So far V3D has been tested on hundreds of machines with different software/hardware environments; in most cases (Mac, Linux, Windows) the software has a nice performance. Yet the following are some known problems.

* Mac machines with Tiger (Mac OS X 10.4) and defective NVIDIA GeForce 7300GT video cards: GeForce 7300GT does not support multisampling while it claims to be. This incompatibility makes 3D viewer run very slow. A special V3D version disables the multipling support on this video card solves the problem.

* MacBook Pro with Tiger (Mac OS X 10.4) and defective NVIDIA GeForce 8600MGT video card driver: Sometimes the labels of markers may be hidden in the 3D volume image rendering. Upgrade to Leopard OS (and thus also upgrade Mac OpenGL driver) solves the problem.

5. Acknowledgements

V3D is a software package developed primarily in Hanchuan Peng lab at Janelia Farm Research Campus, Howard Hughes Medical Institute (HHMI), 19700 Helix Drive, Ashburn, VA, 20147, USA. The project was started in 2006. The core V3D development team includes Dr. Hanchuan Peng, Dr. Zongcai Ruan, and Dr. Fuhui Long, with help and feedbacks from a number of users.

This manual has used examples of 3D images generated in Chris Doe lab (HHMI, Univ Oregon), Stuart Kim lab (Stanford Univ), Julie Simpson lab (Janelia, HHMI), Gerry Rubin lab (Janelia, HHMI), Tzumin Lee lab (Janelia, HHMI, and Univ Mass.), NeuroMorpho Database (George Mason Univ), and couple of other places. We are also grateful to Gene Myers lab (Janelia, HHMI) for the discussion and support of the developing of V3D.

