

The viewer3d: Viewing 3D images with v3dlib

Contributed by Petr Matula
Last Updated Tuesday, 03 November 2015

The viewer3d is a cross-platform C++ computer program designed for displaying and inspecting 3D volumetric images. It is based on our i3dlibs and v3dlib libraries. Thus, it supports file formats that the i3dcore library supports including, optionally, the Bioformats library. The visualization features those of the v3dlibs including traditional orthogonal projections and OpenGL (shown below) based volumetric rendering of the data.

Note that there is an advanced version, that allows to display and inspect a sequence of 3D images, of this software utilized in the new viewer4d application.

{mospagebreak title=The v3dlib library}

v3dlib library is a cross-platform C++ library written in wxWidgets for displaying 3D images. The most important part of the library is a window that can display 3D images of various types (RGB, RGB16, GRAY8, GRAY16, and float). The window can easily be extended of a new functionality and is used in many applications (e.g., viewer3d, Acquiarium, batchcrop, and others).

The main purpose of this article is to provide basic information about the 3D window and its efficient usage. The article describes also the functionality that is accessible only from keyboard.

The main window contains three projection windows - xy, xz, and yz. Each projection window can display either orthogonal projection or maximum intensity projection. The type of projection in all windows can easily be changed by command View->Auto Focus.

Information about the loaded image is available in info window (Press **I**). It contains several tabs (Pixel info, Image info, and View info). "Pixel info" tab shows information about the voxel under the mouse pointer. "Image info" shows parameters of the loaded image: its type, size and voxel size and "View info" tab informs you about the current view.

B
Bbox
" information about the bounding box of the image (offset is usually zero and the size corresponds to the image size in microns).

V
Visible
" the portion of 3D space that is displayed in projection windows. It is always contained in the bounding box.

C
Cut
" current cut position. It is an intersection point of all three orthogonal cutting planes.

Remaining parts of this article show how to

C
Change
3D cut position

M
Move
visible part

C
Change

brightness and contrast

• Display greyscale images in colours

{mospagebreak title=Change of 3D Cut Position}

The position of the orthogonal projections is shown by a cut pointer. It is shown with the yellow dashed lines in the Figure below. Its drawing style can be changed in menu View->Preferences.

Its position in 3D space is stored in microns, therefore it can be set with sub-voxel precision. To try this, zoom in the image until you see individual voxels. Select the xy-projection window with the left mouse click. Then, you can change the cut position in xy-plane by pressing Space button. The new position is set to pass through the current mouse position. In order to change the position with sub-pixel precision also in z-dimension, you have to select xz- or yz-projection window.

Cut position can also be changed by PgUp and PgDn buttons. They move the selected orthogonal projection along its perpendicular axis (e.g. along z for xy-projection window). The increment is set to the voxel size, which allows browsing through the voxels slice-by-slice.

{mospagebreak title=Moving visible part}

The moving of visible part is needed when the visible box is smaller than the bounding box. To test the short-cuts described here load an image into a viewer, select View

info tab of the info window and zoom in the image until the visible part is smaller than the bounding box.

Position of the visible box can be changed in these ways:

Arrow keys " move view in the selected projection window in the pressed direction.

" move the view so that the voxel under the mouse position is displayed in the centre of the selected projection window (or close to the centre if the voxel is near the image boundary).

" Alt-C " move the view so that the cut position is shown in the middle of all projection windows (or close to their centres if the cut position is near the image boundary)

{mospagebreak title=Changing Brightness and Contrast}

Brightness and contrast can easily be changed in menu Mapping. The most general tool is a Levels dialog. In this dialog a mapping from input values (voxels in the image) to output values on the screen can be modified. Linear ($\gamma = 1$) as well as non-linear ($\gamma \neq 1$) mapping is supported.

To increase contrast, move the black and white triangles under the histogram closer to each other. To use the whole intensity range of the image set the black and white triangles to the beginning and end of the intensity histogram, respectively.

To increase brightness, move the middle grey triangle closer to the white triangle. Move it closer to the black triangle to decrease brightness.

Intensity levels can be altered also by the selection of Linear or Percentil stretch in the menu Mapping.

Linear stretch maps the minimal value in the image to 0 and the maximal value to 255. Other values are lineary interpolated.

Percentil stretch maps 5% of the smallest values to 0 and 5% of the largest values to 255. Other values are lineary interpolated. This stretch is less sensitive to outliers in the data (e.g. hot pixels).

{mospagebreak title=Colour mapping}

Each channel of an RGB or RGB16 image is processed individually and mapped to the range [0..255]. These channels are then combined to a resulting RGB value. Individual channels can be selected from the menu Mapping.

Voxel values of greyscale images of all types are first mapped to a value from the range [0..255], which defines an index to a colour palette. This allows displaying of greyscale images in colours. The possibilities are:

- Gray - no palette is used. Only grey values are shown
- Saturation - voxels mapped to 0 are shown in blue colour, voxels mapped to 255 are shown in red colour, and all other values are shown in grey. This enables visualization of under- and over-exposed values. For example this choice together with Percentil stretch shows 5% of the lowest values in blue and 5% of the greatest values in red.
- Labels - a distinct colour is assigned to each value. It is suitable for displaying of indexed (labelled) images after segmentation.
- Periodic - a palette with gradual colour and the same colour for 0 and 255. This mode is, for example, good for the visualization of orientations or intensity levels in images.
- Palette - a user defined palette can be read from a file.

{mospagebreak title=Overlaying}

The overlay displaying regime may come handy when inspecting segmentation results. That is, when one have original grey scale or colour image along with a mask image.

The mask image is supposed to display results of the segmentation. It must be of the same size and resolution as the original image. It should display background with black (zero intensity). The foreground may be labelled (i.e., each component is displayd with its unique intensity) or the mask image is binary (i.e., foreground is just some non-zero value). The library assignes its own set of labels to the content of the mask image anyway.

The overlay is activated from File->Open overlay. First the original image is given, then it asks for the mask image. Notice that the new menu Overlaying becomes available.

The original image can be displayed and inspected as usually. On top of it, an overlay image can be displayed (Overlaying->Enable) either as whole components (default option, Overlaying->Components) or only as a boundary (Overlaying->Boundary). Especially in the former case, the Overlaying->Increase transparency option may be worth trying.

The example of overlaying the whole components with slightly increased transparency (upper) and with just boundary (lower).

{mospagebreak title=License, Download, Authors}

License and Download

The program viewer3d, which includes the binary of the v3dwins library, is available under the GNU LGPL license and is freely available as a windows installer package . The source codes are available upon an e-mail request.

Authors

The program and the library have been

developed in the Center for Biomedical Image Analysis, Faculty of Informatics, Masaryk University, Brno, Czech Republic, by several people.

The principal authors of the library are Petr Matula, Martin Maříka, and Pavel Matula. Contact Petr Matula for more information about the library.

A follow-up application: viewer4d

Note

that there is an advanced version, that allows to display and inspect a sequence of 3D images, of this software utilized in the new viewer4d application.