Free-D Manual

Version 1.11

March 2015

Philippe ANDREY (philippe.andrey@versailles.inra.fr)
Institut Jean-Pierre Bourgin, UMR1318 INRA-AgroParisTech
INRA, Centre de Versailles-Grignon, Bâtiment 2
Route de Saint-Cyr (RD 10), 78026 Versailles, France

Contents

About this manual

This manual is divided into three main parts:

- the first part is a one-page tour of *Free-D*, describing briefly how to proceed from a series of image files to a first 3D reconstruction;

- the second part is the "how-to" section, describing in detail the software functionalities (segmentation, registration, etc.);

- the third part is designed for the experienced user who looks for a minute description of an item in the software interface: it describes software components such as windows, menu items, tool bars and buttons.

It is recommended that the three parts should be read in that order.

$\boxed{v1.11}$ Major changes in version 1.11 are highlitghed in blue.

Other sources of information

You can subscribe to the Free-D newsletter at the following address:

https://listes.inra.fr/sympa/info/free-d

Citing *Free-D*

Any publication of results generated using *Free-D* must include the following reference:

Andrey P & Maurin Y (2005). *Free-D*: an integrated environment for three-dimensional reconstruction from serial sections. *Journal of Neuroscience Methods*, **145**, 233-244.

1 One-page manual

For the impatient user, this page explains how to quickly obtain a first reconstructed model using *Free-D*. It is assumed that a series of slice images stored in the TIF format is available and that the objective is to reconstruct a structure that can be delineated as closed contours on the images.

1.1 Stack initialization

Select **File**→**New stack**. Give the stack a name. Enter the pixel calibration and the inter-slice spacing. Click on the **Add...** button and select the image files (it is assumed that filenames follow the physical order in the stack). Click **Done** when finished. The stack now appears in the browser on the left side of the application main window and the image of the first slice is displayed on the right side.

1.2 Image segmentation

The structure to be delineated has first to be declared. Select **Edit**→**Models...** to call the model dialog. Click the **New** button. Leave the default name ("Model 1") and the default object type ("Polygon") unchanged; simply click **Apply** to complete the declaration, then close the dialog.

The contours of the structure can now be drawn on the images. In the image window, click on the right mouse button and select **Draw**→**Manual**→**Model 1** in the contextual menu. Enter contour points one at a time using mouse left clicks. Type **Enter** to complete the drawing. Select the next image by clicking on its name in the browser and repeat the drawing operation until the whole stack is segmented.

1.3 Image registration

Click on the registration icon to call the registration window. Click on **First** to display the first two images. Translate the second image atop the first one using mouse left click-and-drag. Similarly, rotate the image using right click-and-drag. When a correct position has been found, click **Next** to register the third image with respect to the second one, etc. Click **Apply** when the last image has been processed and close the window.

1.4 Model rendering

Click on the model rendering icon to call the rendering window. The structure appears as a solid surface that has been generated by fitting a 100×100 quadrangular mesh on the stacked contours. Depending on your data, this resolution may be either too large or too small. To find an appropriate resolution, select **Edit**→**Models...** and select "Model 1" in the model dialog. Change the width and height of the quadrangular mesh and click **Apply** to visualize the resulting surface. Close the dialog when the correct settings have been found.

2 Usage manual

2.1 Typical usage cycle

Free-D offers functionalities for initializing image stacks, applying geometric transformations to bring consecutive images into correspondance (**registration**), delineating the contours or marking the positions of objects of interest in the images (**segmentation**), and generating, displaying and manipulating 3D representations of these objects (**reconstruction** and **rendering**).

Free-D also includes a quantitative analysis module. The first version of this module provides counting and geometrical measurements over slices and models.

A typical session generally consists in the following steps:

1. load an already existing stack description file;

2. perform image registration;

3. perform image segmentation;

4. view the reconstruction;

5. repeat *ad libitum*, in any order, any of steps 2, 3, and 4;

6. save the modified stack to disk;

7. load and similarily process other stacks.

The first session will generally differ in that an additional stack initializiation will be performed first. It is also possible to load stack files that have been generated externally (see the next section).

2.2 Creating and saving stacks

2.2.1 Creating stacks: the simple method

A stack can be created using one out of three methods. The first, and simplest, one can be used to initialize a stack in which each slice image is stored as a unique file (this should cover most practical situations). Select **File→New stack**. Give the stack a name.

Select the appropriate length units (default is μm). All subsequent length data should be expressed in the selected units. Note that unit lenghts are only symbolic and are not numerically processed. Hence, changing unit lengths will not alter previously entered length values. Enter the distance between consecutive image planes. To specify the pixel calibration of the images, select one out the three combinations of pixel width, pixel height, and aspect-ratio (= pixel height/pixel width) and enter the two corresponding values. The third value will be automatically computed by the software. Remember that inter-slice distance and pixel calibration should be expressed in the same length units.

$\boxed{v1.11}$ Click on the **Add...** button and select the image files. At present, *Free-D* can load 8 and 16 bits TIF files, as well as IMG files. The alphanumerical order of filenames is assumed to correspond to the physical ordering of the slices in the stack. The **Add...** operation can

be repeated. If necessary, click **Remove** to suppress selected files in the current image list. Click **Done** when finished.

The stack now appears in the browser on the left side of the application main window and the image of the first slice is displayed on the right side. The stack also appears in the browser of all other existing main windows.

2.2.2 Saving stacks

To save a newly created stack on disk, select **File→Save stack** or click on the floppy disk icon. From the dialog, select a directory and enter a filename. This will be used to create a new, or to overwrite an already existing, Stack Description File (SDF). If omitted, the filename extension ".sdf" is automatically appended. The SDF is an ASCII text file that contains all the information describing the created stack. This is also the place where any subsequent segmentation, registration or rendering information will be stored. Consult the Reference Manual for more information on SDFs and their syntax.

Upon the next saving operation, the stack will be automatically saved in its associated SDF. A backup copy of the original SDF is generated, with a "∼1" appended to the filename. Upon subsequent savings, this backup copy will be renamed with a "∼2", etc. When the maximum number of backup files (currently fixed to 5) is reached, the oldest one is overwritten. To save the stack in another SDF, select **File→Save stack as. . .** .

Lastly, note that the SDF is not necessarily located in the directory containing the images. Images can for example be stored on a CD-ROM, a DVD, or in a remote directory mounted by NFS or SAMBA, while the SDFs are stored in a local directory. The location of image files can be stored as an absolute or relative path from the directory of the SDF file. The default is to store a relative path. In some situations (e.g., images on a CD-ROM or located on a NFS mounted directory), it may be preferable to use an absolute path. This can be changed in the **Edit→Stack Properties. . .** dialog.

2.2.3 Creating stacks: advanced methods

The second stack creation method consists in creating a SDF file using an external text editor. Though less convenient, this second method is more powerful and flexible. In particular, it allows the initialization of stacks for which slice images are to be recomposed as mosaics of tile images stored in separate files. SDF created using the first method described above can also be edited to perform any operation not yet available through the software graphical interface. Consult the Reference Manual for information on the SDF syntax. When more and more advanced operations become available through the software interface, such direct SDF manipulations will of course no longer be necessary.

The third stack creation method consists in automatically generating a SDF file from an external program (e.g., acquisition software). A stand-alone program (*sdfmake*) is available on our web site to generate SDF from a list of image files. It is particularly useful for generating stacks with images that are selected using pattern matching on filenames. This program is only available as a Linux executable file.

2.3 Stack browsing and image display

An arbitrary number of main windows can be opened (**File→New window**) to allow the simultaneous viewing of different stacks and slices. When a stack is created or loaded from a SDF, it is inserted in the stack browser of all currently opened main windows. The stack browser is composed of two lists displayed one above the other. The first list displays the names of all loaded stacks and allows the user to select the current stack. The second list displays the names of the slices in the current stack and allows the user to select the current slice. To navigate upwards or downwards throughout the stack, the user can also click on the corresponding arrows in the toolbar. It is also possible to go directly to the first (bottom) or to the last (top) slice by clicking the corresponding arrow icons. The slider located below the slice list also allows to navigate through the current stack. Note, however, that memory management in *Free-D* is designed to handle stacks of large images. Hence, images are loaded upon request so that image display cannot always proceed as fast as the slider does.

By default, the current slice image is resized so that it fits in the main window. The actual image size and the display scale appear in the status bar at the bottom of the window. The user can zoom in (resp. out) by clicking on the corresponding icon or using the keyboard shortcut **+** (resp. **-**). You can restore the fit-in-window mode by using the icon or shortcut **=**. Upon zooming in, the viewed image area can be located within an overview of the global image in a thumbnail window obtained by clicking the corresponding icon or the keyboard shortcut **v**. The current image view is displayed as a black and yellow rectangle in the overview window.

The displayed image portion can be moved in all four directions one pixel at a time using the keyboard arrows. Simultaneously pressing down the **Ctrl** key and an arrow key shifts the view by one-third of its size. Such large moves can also be obtained using the corresponding icons in the toolbar. The view can also be moved from the overview window using mouse left click-and-drag. Lastly, hitting the **c** key centers the view on the pixel currently under the mouse cursor.
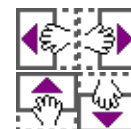
An overview of a stack contents can be obtained through the panorama (syn. gallery) mode. In panorama mode, the main window displays a thumbnail collection of all slice images of the current stack. This image gallery is shown in place of the single image of the current slice. Click on the corresponding icons to switch between the two display modes. In panorama mode, the size of thumbnail images can be decreased or increased using the **Ctrl-L** and **Ctrl-Shift-L** key combinations. The maximum thumbnail size is set to 512×512 pixels.

In both single and panorama display modes, the images can be displayed either with or without the segmented items. It is also possible to display the items without the underlying images. The desired representation is selected using the corresponding icons that are available in the toolbar.
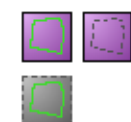
A double slider tool is available in the tool bar to adjust the contrast of the image. The left (black) circle can be moved over the displayed grayscale to set the pixel value that is mapped to black on screen. Symmetrically, the right (white) circle can be used to adjust the pixel value that is mapped to white.

Text notes can be attached to stacks. To add, modify or view the note attached to a stack, select **Edit→Annotate stack. . .** or click on the note icon located at the right of the current stack name in the status bar. The note icon color is grey by default and turns to orange if a note is available for the current stack.

Text notes can be attached to slices. To add, modify or view the note attached to a slice, select **Edit→Annotate slice. . .** or click on the note icon located at the right of the current slice name in the status bar. The note icon color is grey by default and turns to orange if a note is available for the current slice.

2.4 Image segmentation

Segmentation is the process whereby the contours of structures are delineated on slice images or whereby the locations of punctual elements such as cells are pointed at. In *Free-D*, a given structure or cloud of points is represented throughout an image stack by a *model*. Each model is uniquely identified by its name, and is characterized by a set of attributes that determine its graphical appearance. Once a model has been created, items (contours or points) can be defined on the images and added to the model. The attributes of existing models and model items can be edited at any time.

2.4.1 Creating a new model

In the main application window, select **Edit→Models. . .** to call the model dialog (there are several other places in the application wherefrom this dialog can be launched). Click **New** to start the declaration of a new model. Enter a name that corresponds to the biological element (organ, territory, cloud of cells, etc.) that is to be represented by the model. Set properties (see descripion below) as desired. Click **Apply** to validate the creation of the model. Click **Cancel** to abort the creation.

2.4.2 Creating a new item

Entering item points

In the image canvas, select **Draw→Manual→*model*** in the contextual menu to start creating an item of *model*. The cursor shape changes to a cross. Click **Left button** or type key **x** to enter points one at a time. Multiple consecutive clicks at the same position result in only one point being actually added to the item, thus avoiding unnecessary duplicates. Points can be entered in batch by dragging the mouse while maintaining the **Left button** pressed down. However, this usually results in a large number of adjacent points, thus making any subsequent vertex edition time-consuming and possibly unpracticable. As a consequence, this drawing mode should be used with caution. Press **Backspace** to remove points in the reverse order to their addition. Removing the first point cancels the creation. Double-click to end and validate the item creation. It is however preferable to validate the creation by typing the **Return** key because this guarantees that no additional point will be entered. Press **Escape** or **Right button** to cancel the item creation.

Moving the view while drawing

When only part of the image is displayed in the window, it is generally necessary to move the field of view while creating an item. The view can be shifted in any of the four main directions without interrupting the drawing by using the **Arrow keys** (combined with the **Ctrl** key if large motions are desired). If shifts in arbitrary directions are required, press down the **Space bar**. The cursor shape changes to a hand and dragging the mouse while pressing down the left button now results in the field of view being translated in any direction. Release the **Space bar** to restore the cursor shape and continue item drawing.

2.4.3 Editing an existing model

The attributes of an existing model can be modified from the model dialog. Call the model dialog and select the desired model. The model can be renamed and any of its properties, with the exception of its geometry, can be modified. Click **Apply** to validate the changes. Click **Cancel** to discard the modifications and restore the initial properties of the model. Upon validation, the modifications are immediately propagated and applied throughout the application. These modifications apply to already existing items and of course to any future item. The properties of several models can be modified simultaneously by selecting more than one model in the list.

2.4.4 Editing an existing item

Editing item properties

An item can be assigned properties that override the default item properties of its model. Select an item by clicking on it with the **Left button** and select **Edit→Properties. . .** in the contextual menu (**Right button**). The item properties dialog appears. Modify as desired the properties, then click **Apply** to validate. The default item properties can be restored by clicking the **Defaults** button. All modifications can be discarded, and the initial properties of the item restored, by clicking **Cancel**.

Editing item points

To alter the points of an item, select the item. Select **Edit→Edit points** in the contextual menu to enter edition mode. In this mode, the contextual menu has changed and a vertex edition toolbar is shown. Select **Move points** to enable vertex displacement. Click on a vertex and drag the mouse to move the vertex. Select **Insert points** to enable vertex insertion. Click on an edge of an item to insert a new vertex on that edge. Select **Delete points** to enable vertex deletion. Click on a vertex to remove it from the item. Large numbers of vertices can also be deleted by dragging the mouse while keeping the **Left button** pressed down. To leave the edition mode, select **End edition** or type **Escape**.

Editing contours

Vertex decimation can be applied to contours to remove superfluous vertices. Because it only removes colinear vertices, this function simplifies contours without altering their shape. To run this function, select **Decimation** in the contextual menu. A dialog box is shown to inform the user about the number of removed vertices. No dialog is shown if no vertex was removed.

Contour simplification by vertex decimation can also be performed using the Douglas-Peucker algorithm (Douglas and Peucker, 1973). To run this tool, select **Decimation (Douglas-Peucker)** in the contextual menu. The tolerance parameter of the algorithm can be interactively adjusted using the slider and the previsualisation of decimation results provided by the tool.

Uniform resampling of vertices can be applied to contours. This function consists in redistributing a given number of vertices at regular intervals along the original contour. To run this function, select **Resampling** in the contextual menu. The displayed dialog allows you to dynamically select the number of vertices to use for resampling.

Contours can also be smoothed using Fourier descriptors. Simply stated, the principle of this tool is to decompose the selected contour into a multi-scale representation and to

recompose a simplified contour using only the descriptors above a given scale. To use this tool, select a contour and select **Smooth (Fourier)** in the contextual menu. Contour simplification can be done either by selecting the actual number of retained descriptors (the maximum number is given by the number of points in the contour) or by selecting the desired fidelity to the original contour (technically, fidelity is the percentage of energy retained in the Fourier spectrum). Fidelity has the advantage that a comparable simplification can be performed over contours having different number of points. It is advised that a uniform contour resampling be performed before using this tool.

2.4.5 Automatic image segmentation in 2D

Apart from pure manual segmentation, closed contours can be delineated using semiautomatic methods. These tools are not fully automatic insofar as they require some form of initialization by the user.

Image thresholding and contour following

The first method is based on histogram thresholding and contour following. The principle is the following: the user defines a rectangle somewhere across the boundary of the structure to delineate. Ideally, this rectangle should equally overlap the object and its background so that its histogram should be bimodal. The software automatically analyzes the histogram and computes a threshold that is used to binarize the image. The contour of the object is automatically extracted from a started point located in the user-defined rectangle.

This method is run as follows. Select **Draw→Automatic→*Model***. Drag the mouse while pressing down the left button to define a rectangle over some portion of the boundary of the structure to delineate. Upon mouse release, the user is asked for a smoothing parameter. This corresponds to the half-width of a mean filter that may be applied to the image before computing the histogram and extracting the contour. A value of 0 corresponds to no smoothing. The user is then asked for a vertex decimation value. Setting this parameter to 0 generates contours having one point per pixel. Increasing this parameter reduces the number of points on the contour. This allows faster contour processing (in particular at the surface reconstruction stage) and contour simplification. Upon validation, the contour extraction procedure is run. If successful, the resulting contour is displayed over the image (as if it has been manually delineated). If not, an error message is displayed. You may then retry, using another region and/or another smoothing parameter.

Active contours (snakes)

The second method is based on an active contour (snake) model. Snakes are contours that, when initialized in the vicinity of an image contour, progressively move and deform towards that contour while simultaneously preserving geometrical constraints such as smoothness. To run this method, manually define a closed contour near the desired boundary. This initial contour can be a very coarse approximation of the actual target shape. Generally, a few points only are necessary. Select the contour (left click) and choose **Edit→Run snake** from the contextual menu or use the keyboard shortcut **s**. A dialog appears from where the snake can be parameterized. **Alpha** is the parameter of the algorithm (Deriche gradient) used to estimate image gradients. This can generally be left unchanged, unless you know what Deriche method is. The **Threshold** value should generally be decreased if the snake collapses. Alternatively, it should be increased if the snake is stopped by small, local intensity variations that do not correspond to the desired boundary. Increase the **Regularization** parameter to make the snake smoother. Click

OK to start the snake. A new contour is seen that progressively moves from the initial one to the closer boundary on the image. The snake can be interrupted by pressing **Escape**. When the snake has converged, the initial contour is still present and selected: press **Delete** to remove this contour if the final one is correct. Otherwise, select and delete the final contour and retry using another set of parameters. Note that the snake can be computationnally demanding. When running on large images, an option is to work at a smaller scale. The size of the image over which the snake evolves can be decreased using the **Scale** parameter: both dimensions are divided by the specified value (default to 1).

The snake can be automatically re-run on all slices of the stack by selecting the **Run on every slice** option. In this case, the contour defined by the user will be automatically copied on each slice in turn to initiate an independent snake. Additionnally, check the option **Use propagation** to initiate the snake on each slice from the contour obtained on the previous slice. If this option is activated, the snake is propagated from the current slice in both directions towards the bottom and the top of the stack. Note that in the present version, registration data are not taken into account in the propagation. The snake propagation is therefore recommended for image stacks that do not require registration, such as confocal microscopy stacks. Unexpected results may be obtained when applying to stacks of physical sections that require registration, such as series of histological images.

2.4.6   Automatic image segmentation in 3D

v1.11 | A tool is available to automatically extract spot-like objects from 3D image stacks. Only regular stacks (constant section image size, no registration, as in 3D confocal microscopy) can be processed. The tool is called from **Process→Find spots** and will run on the current stack. It works in three main steps:

1. image pre-processing to enhance signals at spot locations;

2. interactive thresholding of the pre-processed image stack;

3. addition of models and items to the stack.

In the case the current stack contains true colors (3 color components per pixel), the user has to select on which channel the spot extraction tool should run. The choice is between red, green or blue values, or intensity (average of the three color components). The software automatically detects stacks with only grey-level images or images with pure color maps (black to red, black to green, black to blue). In these cases, channel selection is automatic and the corresponding part of the dialog is not enabled.

Three different filters can be optionnally applied, in order, as pre-processing steps. Noise reduction can be applied using an average, median, or Gaussian filter (in the latter case, the σ value is half the specified radius). A top-hat transform can be applied to remove objects larger than a specified size. The structuring element used for the opening step of the top-hat can be flat or spherical. A spherical structuring element will provide more robustness to noise, but will require more computing time. Finally, spots can be enhanced at the expense of structures with other geometries (such as membranes or envelopes, which have laminar shape) using the isotropy filter (Biot et al., 2008).

The user should also configure how the results of the spot extraction tool will be organized. Each spot can be represented using a different model. In this case, the models will be named by appending a number to the model basename. Alternatively, the spots can

be represented as different instances of a single model. Each spot can be represented as a 3D connected component (a set of connected voxels above the threshold) or with its equivalent sphere (the sphere with the same volume as the set of voxels). In the current version, the latter option involves a rounding of the actual sphere radius to the nearest integer. This behavior may change in the future.

When the pre-processing steps are completed, a dialog is displayed to let the user enter a threshold. Voxels above the threshold value are considered as belonging to spots. Selected voxels are displayed using a yellow overlay over the current image of the stack (only in single-slice mode, not available in panoramas). This preview allows to interactively adjust the threshold until all desired spots are selected. Remember that the process is run in 3D: a spot can appeared missing on one slice and be detected in the neighbouring slices.

2.4.7 Transferring segmentations between stacks

Segmentations made on one stack can be transferred to other stacks. This is useful for example in confocal microscopy where several image stacks are frequently acquired along different channels. Cell or nucleus contours can be identified in one channel and then copied to other channels. This is also useful when several users are processing the same image stack through different SDF. When all structures have been segmented by different users and are stored in different SDF referring to the same image stack, a unique SDF containing all structures can be generated with this functionality. Similarly, this function can also be used when different parts of the same structure have been segmented by different users and are stored in several SDF.

Select **Process→Transfer models...**. All currently loaded stacks appear in the two upper lists. Select one stack in each of these lists. Only stacks having the same number of images, with images of the same size at each level, can be simultaneously selected. Selection of incompatible stacks is automatically rejected by the software. Select the model(s) to transfer in the corresponding lower list. Click on the **left-to-right** or **right-to-left** button, as appropriate, to apply the transfer. If a model with the same name already exists in the destination stack, the user is asked whether the selected source model should replace it, be appended to it, or be duplicated.

2.4.8 Filtering objects based on containment analysis

v1.11 It is possible to remove segmented points or whole models that fall outside a given structure. Select **Process→Filter models...**. A dialog with two lists is displayed. In the left-hand list, the user selects the reference structure. Only one reference structure can be selected. In the right-hand list, the user selects the models to process. Several models can be selected. Two types of containment analysis can be performed. If the user selects "Points not included in the reference", then all points from the selected models that are not included in the reference structure will be deleted. If the user selects "Models not intersecting the reference", then the selected models that do not intersect the reference structure are deleted. Note that in this second type of analysis, parts of the models may be retained even though they are outside the reference, as soon as other parts are included or intersecting the reference.

2.5 Image registration

Clicking on the registration icon in the toolbar of a main window opens the registration window for the current stack of the main window. A given stack cannot simultaneously appear in several registration windows. Thus, if a registration window is already opened on the current stack, clicking on the registration icon only results in that window being brought to the foreground.

2.5.1 Display

When the registration window appears, no image is selected. Click on a slice to register in the list. The image of the selected slice (foreground image) appears and is displayed as if transparent atop the image of the previous slice (background image). Transparency is computed by combining pixel values of the two images using an arithmetical (addition, multiplication, subtraction) or ordinal (minimum or maximum) operation. The operation is selected from the **Transparency mode** combo box. Select another slice image in the list when a correct transformation has been found. You can also navigate throughout the stack by clicking the **First**, **Last**, **Next** and **Previous** buttons located underneath the list or by using the arrows in the toolbar. The size of the displayed images can be adjusted (buttons **Bigger** and **Smaller**). Clicking on **Display frames** will superimpose the two image frames on the display. By default, images are displayed together with their segmentations (contours, points), if any. Clicking on the appropriate icons will display the images without their segmentations or, conversely, the segmentations alone on a white background.

The bottom right frame displays the stack name and the names of the foreground and background slices currently being registered. The horizontal and vertical translation components are given in pixel units relative to the original image data file, i.e., before any aspect ratio or image scale (for display purpose) transformation. The rotation angle is expressed in degrees.

2.5.2 Geometric transformations

The foreground image can be translated using mouse left click-and-drag. Similarly, it can be rotated by using right click-and-drag. The registration of the current foreground slice is validated automatically when another slice is selected. Any registration change is immediately taken into account by the other components of the software, in particular by the rendering windows. To validate the registration of the current slice without selecting another slice, click **Apply**. Click **Restore** to cancel any modification made to the registration of the foreground slice. Clicking **Reset** zeroes translation and rotation parameters. Note that you need to click **Apply** to validate registration changes due to a restore or reset action. Click **Cancel** to close the registration window without applying the modifications made to the current foreground slice.

2.6 Displaying 3D reconstructions

Clicking on the reconstruction icon in the main window toolbar opens a new rendering window for current stack in the main window. An unlimited number of rendering windows can be simultaneously opened, either to display the same reconstruction from a given

stack (e.g., to compare different points of view) or to display reconstructions from different stacks. Alternatively, a single window can also be used to browse through the 3D reconstructions of several stacks, using the next/previous stack buttons.

A rendering window has a menu bar, a tool bar, a control panel and a 3D display canvas. Use the menu **Show** to hide/show the tool bar and the control panel.

2.6.1 Geometric transformations

The displayed reconstruction can be interactively manipulated from the canvas. Moving the mouse while pressing down the left button rotates the reconstruction around the center of the displayed structures. The applied rotation combines both a rotation around the horizontal axis and a rotation around the vertical one. Rotation around the axis perpendicular to the screen is obtained by dragging the mouse while simultaneously pressing down the **Shift** key. To zoom in, drag the mouse towards the right border of the canvas while maintaining the **Ctrl** key pressed down; zoom out is similarly obtained by moving to the left. Zooming in and out can also be achieved using the mouse **roulette wheel**. Dragging the mouse while pressing donw the **Alt** key translates the view horizontally and vertically. Click twice on the mouse Left button or click the home icon to reset all geometric transformations to their default values (no rotation, no translation, 100% scaling).

Current geometric transformations values are displayed and continuously updated in the control panel of the rendering window. These values can also be set using the sliders and the input boxes of the panel. Should rotation angles be specified this way, note that rotation around the z-axis is applied first, followed by y-rotation and lastly x-rotation.

A random motion that continuously rotates the displayed reconstruction about its center can be toggled by selecting **Pose→Toggle random motion**, by clicking the **r** key or by clicking the dice icon. The animation speed can be controlled using two parameters. Select **Pose→Set motion angle...** to specify the rotation angle between two consecutive motion steps. Select **Pose→Set motion timeout...** to specify the time interval between two consecutive animation steps. Increasing the rotation angle or decreasing the time interval speeds up motion. The rendering scene can also be animated using a continuous rotation around the vertical axis by selecting **Pose→Toggle spinning motion**. The behaviour of this motion can be controlled using the same parameters (step angle and timeout) as random motion.

2.6.2 Model display properties

The appearance of the displayed structures can be modified from the model dialog. Select **Edit→Models...** to launch the model dialog. Select a model in the list. In the **Rendering mode** combo box, select **Vertices** to display the vertices of the structure only, or **Curve** to display the contours delineated on the image slices, or **Wireframe** to display the surface as quadrangular mesh, or **Surface** to display it as a solid or transparent surface. In the **Wireframe** mode, the displayed mesh is an approximation surface that is obtained by resampling the segmented contours into a user-defined number of equally spaced points and by then interpolating a user-defined number of contours between the first and the last resampled contours. The dimensions of the mesh are specified in the **Quadmesh width** and **Quadmesh height** input boxes. In the **Surface** mode, surface transparency is modified by changing the **Opacity** value. Fully opaque surfaces are generated by setting **Opacity** to 255. Decreasing this value increases surface transparency (minimum is 0,

invisible surface). Caps can be added to close surfaces at their beginnings and ends. Begin/end caps are independently set using the **Begin cap** and **End cap** check boxes.

To increase the readability of a 3D rendering, it may be necessary to hide a whole model. To this ends, uncheck **Rendering visible** in the **Item properties** panel of the model dialog. Note that for structures displayed in **Vertices** or **Curve** mode, some selected items may be hidden as well. To hide an item, select it from either the main window or the full view window, select **Edit→Properties...** in the contextual menu and uncheck **Rendering visible** in the model item properties dialog.

The background color can be changed from the control panel. The amounts of ambient, diffuse and specular light that illuminate the scene can be specified from the control panel. Specular light is relevant only for surfaces that are displayed with shininess.

2.6.3 Storing and restoring poses

A pose corresponds to a particular set of rendering parameters. A pose defines the three rotation angles, the two translation components, the scaling factor, and the three light components. Model display properties are not specified by poses. The six predefined poses correspond to the canonical orientations (coronal, horizontal and sagittal). Their are available through the **Pose** menu. User-defined poses can be added to the list of available poses. To store a pose, select **Pose→Record...**, give the pose a name and click **OK**. The pose can now be restored at any time in the rendering window wherefrom it was defined as well as in all other rendering windows that simultaneously display the same stack reconstruction. User-defined poses are automatically saved into the SDF when the stack is saved from the main window. Poses can thus be restored from one session to the next. To remove a user-defined pose, select the corresponding name in the **Pose→Delete** submenu.

2.6.4 Capturing rendering scenes

Snapshots of rendering scenes can be stored on disk as image files. To make a snapshot, either select **Window→Snapshot** or click on the snapshot icon. A dialog appears that asks for the desired image file. The default image size is the current size of the rendering canvas. The size can be increased or decreased as desired. Note that the aspect ratio is always preserved: modifying one dimension automatically updates the other one. Once the size has been entered, the user is asked for an image file name. The image format under which snapshots are stored is PNG.

2.6.5 Exporting 3D reconstructed models

v1.11 Reconstructed surfaces can be exported and saved under the STL format. Select **Rendering→Export to STL...** to save all surface models of the current stack into separate files in the desired directory. For each model, the export file name is built by concatenating the stack name and the model name. Note that only surface models can exported in this format. Point models will not be exported with this functionality.

2.7 Quantitative analysis

The quantification module allows for counting and for geometrical quantification, either in two (model items) or in three dimensions (models). In addition, this module can also report a summary of all segmentations than have been made over stack slices. Select **Analysis→Run** to call the quantification window. The three analysis modes (items, slices, models) are available as tabs.

2.7.1 Two-dimensional analysis

Select the **Items** table in the analysis window. The left side of the window now displays the list of all model in the stack. Select the model for which 2D analysis is desired in this list. The right side of the window then displays the results of measurements made on all items belonging to the selected model. Each row corresponds to an item. The first column gives the name of the slice to which the item belongs. The remaining columns display quantitative measures. Depending on the model geometry (close curve, open curve, or set of points), these measures include number of vertices, length, area, centroid (average of all item vertices), and slice z-position. Note that for sets of points, each row correspond to one point and gives its x, y, z coordinates.

Note that in this mode, vertex coordinates are expressed in each slice coordinate frame. Registration data, if any, are not taken into account.

2.7.2 Three-dimensional analysis

Select the **Models** table in the analysis window. Three sub-tables can be selected, corresponding to the three model geometries: Polygon (**Closed surfaces**), Polyline (**Open surfaces**), and Polypoint (**Point clouds**). When selected, each table displays the results of measurements made over all models of the associated geometry. Each row corresponds to one model and the columns to measured variables. Depending on model geometry, the measures include the total number of vertices, the centroid (average of all vertices), the total number of contours, the area of the reconstructed surface, and its enclosed volume.

Note that in this mode, vertex coordinates are expressed in a unique coordinate frame that is common to all stack slices. Hence, registration data, if any, are taken into account.

2.7.3 Slice-by-slice analysis

Select the **Slices** table in the analysis window. A table is displayed that gives for each slice (row) the number of segmented items of each model (columns).

2.7.4 Saving and exporting results

To save the current result table, select **File→Save. . .** . Select an existing file or enter a new filename, and then validate. The current table is then saved as a text .xls file, with columns separated by tabulations. This file can then be reopened from another program such as a text editor, a spreadsheet, or a statistical software for further analysis.

To copy-paste results from *Free-D* to another program, you can also select all (**Ctrl-A**) or part of the current table, then copy the selection to the clipboard (**Ctrl-C** or **Edit→Copy**) before pasting it into the destination software.

2.8   Model registration and averaging

Free-D is specifically designed to process collections of stacks in the objective of generating statistically representative models of structures and spatial distributions. With the spatial normalization module, users can register 3D reconstructions generated from different image stacks, can average surfaces, and can produce statistical maps of object densities.

2.8.1   Registering and averaging 3D models

The first task that can be achieved with the spatial normalization module is to register models from different stacks. The module is called from **Process→Normalize**. From the displayed normalization window, select **Normalize→Normalization...**. A normalization dialog appears. The user must select the stacks to process in the list on the left. Pressing the **OK** button at this stage will display the mere superimposition of the 3D reconstructed models. The result will generally not be satisfactory since this involves no registration.

Registration between 3D reconstructed models is computed using one or more reference structures. At present, only closed surfaces (corresponding to objects with a Polygon geometry) are allowed as references. The module automatically identifies the surface structures that are present in all selected stacks and list them in the middle part of the dialog. The user must select the structures to use as references. All instances of the references will be resampled using the number of contours (Height) and number of vertices per contour (Width) given here.

Registration can be computed according to different methods. The simplest method is based on the alignment of the main axes of the reference structures across stacks (**Step 1**). This is generally used as a rough initialization. This alignement is not recommended if there is no major difference between the three main axes, as axis swap can occur, resulting in completely incorrect alignement between models.

Using registration (**Step 2**), possibly after axis alignment, will generally produce more precise results. Registration of a collection of models can be obtained by independently registering each model onto a target (**Pairwise**). In this case, the target is selected from the **Template stack** list. Alternatively, a groupwise registration algorithm can be run (**Global**) to simultaneously register all models of the collections. This is generally the recommended option. The algorithm is detailed in Maschino et al. (2006).

Clicking the **OK** button will run the registration and averaging process. The result appears in the normalization window. The interface here is similar to the 3D rendering window, except that a list of structures, including the average ones, is listed on the right. Note that all surface structures are averaged, even those that were not selected as references. Point-like structures (models with Polypoint geometry) are not averaged (but see below about density estimation). The listed structures can be selected and their visual attributes modified. The interface here is identical to the one of the standalone *sviewer* companion software (also available from `http://free-d.versailles.inra.fr`).

2.8.2 Spatial normalization

With only registration, non-reference structures passively follow the rigid transformations (translations and rotations) of the reference structures. Hence, non-reference structures are not correctly repositionned relatively to reference averages. Non-linear deformations can be computed to replace all structures within the space defined by the reference structures. For each stack, a parametric model of the deformations mapping the reference structure(s) onto its(their) average(s) is computed and propagated to all other structures of the stack. An initial descritpion of this warping strategy is given in Andrey et al. (2008).

Warping is performed when selecting **Step 3** in the normalization dialog. Two deformation models are available. The first one is a multivariate polynomial deformation (Andrey et al., 2008). The degree of the polynomials is given by the user. Higher degrees correspond to more complex deformation models, i.e., larger number of degree-of-freedom. The second model is the thin-plate spline deformation (Bookstein, 1989). A major difference between the two deformation models is that polynomials entail an approximation of the actual mapping of the reference structures onto their averages, while the thin-plate splines entail an interpolation. For noisy data, we recommend using the polynomial transforms. Note that using thin-plate splines can also be computationnally very demanding when the total number of vertices over reference surfaces is large.

2.8.3 Statistical density estimation

The density mapping tool can be used to generate average representations of point-like structures (generally used to represent cells within tissues or vesicles within cells). Integration is performed here not by averaging point positions across stacks, but by estimating the "average" density at any location in the averaged reference space.

To compute a density map following registration and warping, select the **Density maps** tab in the normalization window. A density configuration dialog appears. Select the point datasets to include into the density map in the list on the left. In the **Method** list, select the density estimation method. **Histogram** counts the number of points in each box of a 3D partitioning of the space. **Kernel** corresponds to non parametric kernel density estimation. In this case, the user has to additionally select the kernel (Gaussian, uniform, Epanechnikov, biweight) and the bandwidth. **Poisson** corresponds to a parametric density estimation method (Burguet et al., 2009, 2011) that relies on measured distances to the kth nearest neighbor in each dataset. For the last two methods, density is estimated at the nodes of a 3D grid. When clicking **OK**, the density map is computed and appears in the list of density maps.

To visualize a density map, select it in the list an select **View**. Density maps can be rendered according to three modes. **Counts** displays a sphere for each node of the grid, with a radius proportional to density. This option is highly not recommended when the number of nodes in the grid is large, as the rendering may be very slow. **Isosurface** displays the density isosurface corresponding to the specified density threshold. **Percentile** displays the density isosurface corresponding to the specified percentile of the density distribution. Several views of the same density map can be displayed simultaneously.

3 Reference manual

3.1 Stack Description File (SDF) format

The SDF format is an ASCII format. A SDF contains a collection of attribute/value declarations. Each declaration starts on a new line, can extend over several lines, and is closed by a semicolon. If a declaration is omitted, the corresponding parameter is assigned a default value. The sequence of declarations is divided in three parts that must appear in the following order:

1. Global stack declarations: this includes a number of default values declarations (e.g., default inter-slice distance), standard images and calibration curves, and rendering poses.

2. Slices declarations: this part contains the declaration of all slices in the stack, together with registration data.

3. Models and model items declarations: models are declared first, followed by their items, which can override some of the default model properties.

The table below lists the keywords to use in the declarations together with their syntax (NA = feature not yet available). When applicable, default values are given within square brackets.

Attribute	Description
Stacks	
STACK *<name>*;	Starts stack declaration
IMAGESPATH *<dirname>*;	Location of image files
AbsoluteImagesPathPolicy *<flag>*;	Flag for absolute/relative image path
PIXASPECT *<value>*;	Default pixel aspect-ratio [1.0]
PIXWIDTH *<value>*;	Default pixel width [1.0]
PIXHEIGHT *<value>*;	Default pixel height [1.0]
RELPOSITION *<value>*;	Default inter-slice distance [0.0]
Slices	
SLICE *<name filename>*;	Starts slice declaration (single image)
SLICE *<name rows cols>*;	Starts slice declaration (mosaic image)
IMAGE *<r c filename>*;	Image filename at mosaic position r,c
PIXASPECT *<value>*;	Overrides default pixel aspect-ratio
PIXWIDTH *<value>*;	Overrides default pixel width
PIXHEIGHT *<value>*;	Overrides default pixel height
POSITION *<value>*;	Absolute z-position from origin
RELPOSITION *<value>*;	Relative z-position from previous slice
FLIP *<hflag vflag>*;	Flags for image flip (NA)
ROTATE *<angle>*;	2D registration: rotation angle [0.0]
SHIFT *<dx dy>*;	2D registration: translation vector [0 0]
Models	
MODEL *<name>*;	Starts model declaration
BRILLIANCY *<value>*;	Surface shininess
COLOR *<r g b>*;	Color
ECORENDERING *<flag>*;	Economic rendering flag
FILLEDIN *<flag>*;	Flag for filling-in on 2D displays
HIDDEN3D *<flag>*;	3D visibility status
POINTSIZE *<value>*;	Symbol diameter
PRIMITIVE *<name>*;	Geometric primitive
RENDERINGMODE *<name>*;	3D rendering mode
RESAMPLING *<height width>*;	Quadrangular mesh dimensions

THICKNESS *<value>***;**	Line thickness	
TRANSPARENCY *<value>***;**	Surface transparency	
VERTEXSYMBOL *<name>***;**	Vertex symbol	
Model items		
MODELITEM *<name>***;**	Starts model item declaration	
MODELITEMDATA *<x,y . . .>***;**	Model item vertex coordinates	
COLOR *<r g b>***;**	Overrides model color	
FILLEDIN *<flag>***;**	Overrides model filling-in flag	
HIDDEN3D *<flag>***;**	Overrides model 3D visibility status	
POINTSIZE *<value>***;**	Overrides model symbol diameter	
THICKNESS *<value>***;**	Overrides model line thickness	
VERTEXSYMBOL *<name>***;**	Overrides model vertex symbol	
Poses		
POSE *<name>***;**	Starts pose declaration	
POSEROTATION *<x y z>***;**	Pose rotation angles	
POSESHIFT *<horiz vert>***;**	Pose translation vector	
POSEZOOM *<value>***;**	Pose zoom value	
AMBIENTLIGHT *<value>***;**	Pose ambient light intensity	
DIFFUSELIGHT *<value>***;**	Pose diffuse light intensity	
SPECULARLIGHT *<value>***;**	Pose specular light intensity	
Miscellaneous		
# *<comment>***;**	Comment (ignored)	
FileFormatVersion *<value>***;**	SDF format version	
VERSION *<value>***;**	Software version (NA)	
Annotation *<text>***;**	Stack or slice annotation	
NOTE *<text>***;**	Model annotation (NA)	
CREATED *<date>***;**	Creation date (NA)	
MODIFIED *<date>***;**	Modification date (NA)	
CREATEDBY *<username>***;**	User on creation (NA)	
MODIFIEDBY *<username>***;**	User on last modification (NA)	
INCLUDE *<filename>***;**	Includes another SDF (NA)	

3.2 Main window

3.2.1 Stack and slice browser

A stack browser is displayed on the left hand side of each main window. It displays the list of stacks currently loaded in the application. The list of slices of the current stack is displayed in the lower part of the browser.

3.2.2 Tool bar

Launches the file dialog from which the user can select a SDF to open.

Saves the current stack in its SDF. The user is asked a file name and directory if this is the first time that the stack is saved since it was created.

Closes the current stack and makes another stack the current one. The user is asked confirmation if the stack has been modified since the last save operation.

Selects the panorama display mode.

Selects the single-view display mode.

Opens or raises the registration window for the current stack. Only one registration window can be opened on a given stack.

Opens a new rendering window for the current stack. An arbitrary number of such windows can be opened on each stack.

Recursively undoes the last action. Only segmented items actions (creation, edition, deletion) are stored in the action history.

Recursively redoes previously undone actions. Only segmented items actions (creation, edition, deletion) are stored in the action history.

Launches or raises the overview window associated with the main window.

Increases the magnification level.

Decreases the magnification level.

Adjusts the magnification level so that the image fits into the window.

Displays slice images together with their segmentations, if any.

Displays slice images without their segmentations.

Displays segmentations without slice images.

Makes the first (bottom) slice the current one.

Makes the next (upper) slice the current one.

Makes the previous (lower) slice the current one.

Makes the last (top) slice the current one.

3.2.3 Image canvas

MOUSE ACTIONS

Left button On or near an unselected item, selects the item and deselects any previously made selection. Away from any item, clears any previously made selection.

Shift + Left button Adds unselected pointed items to the current selection and removes selected pointed items from the current selection.

Right button Displays contextual menu.

Draw→**Manual**→*model* Enters manual item creation mode. The created item will belong to *model*.

Edit→**Properties...** Calls the properties dialog that lets the user override the default model properties for the selected item.

Edit→**Edit points** Enters the item edition mode.

Select→*model* Selects all items belonging to *model* on the current slice. Only models with at least one item in this slice are displayed in the **Select** submenu.

Convert→*model* Converts all selected items on the current slice to *model*. Conversion is authorized only across models sharing the same geometry (e.g., closed polygon to closed polygon).

Delete Deletes all selected items on the current slice.

VERTEX EDITION FLOATING TOOLBAR (ITEM EDITION MODE)

Move points Enables vertex displacement for the selected item.

Insert points Enables vertex insertion for the selected item.

Delete points Enables vertex deletion for the selected item.

End edition Leaves item edition mode.

KEYBOARD ACTIONS

Arrows Shifts the current view by one pixel.

Ctrl-*Arrows* Shifts the current view by one third of its width or height.

Ctrl-A Select all items on the current slice.

Backspace In item creation mode, removes the last entered point. Leaves item creation mode when the first point is removed.

C Centers the view on the current mouse position.

Delete Deletes all selected items on the current slice.

Escape In item creation mode, cancels the operation and leaves the mode. In item edition mode, leaves the mode.

Return In item creation mode, validates the currently created item and leaves the mode.

X In item creation mode, adds a vertex at the current mouse position. Similar as **Mouse**→**Left click**.

3.3   Registration window

3.3.1   Toolbar

Displays both the foreground and background images together with their segmentations (contours, points), if any.

Displays the foreground and background images without their segmentations.

Displays the segmentations without the images.

Validates the current registration, then selects and displays the second image of the stack for registration on the first one.

Validates the current registration, then selects and displays the next image in the stack for registration on the current foreground image, which becomes the background one.

Validates the current registration, then selects and displays the current background image for registration on its preceding image in the stack. The current background image becomes the foreground one.

Validates the current registration, then selects and displays the last image in the stack for registration on the before-the-last one.

3.3.2   Command panel

First Validates the current registration, then selects and displays the second image of the stack for registration on the first one.

Next Validates the current registration, then selects and displays the next image in the stack for registration on the current foreground image, which becomes the background one.

Prev Validates the current registration, then selects and displays the current background image for registration on its preceding image in the stack. The current background image becomes the foreground one.

Last Validates the current registration, then selects and displays the last image in the stack for registration on the before-the-last one.

Display→Bigger Increases the display size of both the foreground and background images. The maximum size allowed is such that both images can be completely displayed on the screen.

Display→Bigger Decreases the display size of both the foreground and background images.

Display→Transparency mode Selects the image transparency mode.

Display→Display frames Activates the display of the foreground and background image frames. Displaying image frames is a visual source of feedback that may assist the user in the registration process.

Reset Sets to zero all registration parameters (translation and rotation).

Restore Cancels any modification made to the current foreground image registration. Restores the last accepted registration parameters.

Cancel Discards any modification made to the registration of the current foreground image and closes the registration window.

Apply Validates any modification made to the registration of the current foreground image.

3.4   Model dialog

New Starts the creation of a new model.

Delete Deletes the selected model(s) and all its(their) items. Asks user confirmation if this action will delete at least one item. Note that there is no undo for this operation.

Apply In creation (or edition) mode, validates creation (or edition) and leaves the mode.

Cancel In creation (or edition) mode, leaves the mode without validation.

3.4.1   Model properties

Model properties are attributes that either are shared by all model items (e.g., geometry) or that are not relevant at the item level (e.g., surface attributes). The first model property is the model geometrical primitive:

Polygon to represent structures that form closed contours on the images

Polyline to represent structures that form open contours on the images

Polypoint to represent set of points.

This property is specified upon model creation and cannot be changed subsequently.

3.4.2   Item properties

Item properties can be overriden on each model item. By default, model items share the values stored with their model.

Color Defines the color that is used on 2D and 3D displays.

Rendering visible When unchecked, removes the model or the model item from 3D displays.

Vertex symbol Defines the geometrical appearence of vertices on 2D(3D) displays. Available symbols are dot(dot), circle(sphere), square(cube), triangle(tetraedron) and losange(diamond). This attribute is only applicable to models with Polypoint geometry. Check **Fill vertex symbols** to fill symbols with model color on 2D displays.

Line width Defines the line thickness for models with Polygon or Polyline geometry.

Point size Defines the symbol size for models with the Polypoint geometry. Sizes are expressed in pixel units.

3.5   Rendering window

3.5.1   Menu bar

Edit→**Edit models. . .** Launches the model dialog.

Pose→**Record. . .** Asks the user for the name under which the current display parameters will be stored in the list of user-defined poses for the current stack.

Pose→**Select**→***name*** Selects a previously recorded pose by its name.

Pose→**Delete**→***name*** Removes the selected pose from the list of user-defined poses in the current stack.

Pose→**Toggle random motion** Toggles the random rotating motion of the rendering scene around its center.

Pose→**Toggle spinning motion** Toggles the rotational motion of the rendering scene around the vertical axis.

Pose→**Set animation angle. . .** Selects the rotation angle between two consecutive animation steps.

Pose→**Set animation timeout. . .** Selects the time interval between two consecutive animation steps.

3.5.2   Tool bar

Resets geometric transformations to their defaults values (null rotation, null translation, 100% scaling factor).

Takes a snapshot of the current display and saves it in an image file.

Toggles the random rotating motion of the rendering scene around its center.

Displays the 3D reconstruction of the previous stack.

Displays the 3D reconstruction of the next stack.

3.5.3 Command panel

Rotation Sliders and input boxes can be used to enter rotation angles. Rotations are applied first around the z-axis, then around the y-axis, and lastly around the x-axis.

Translation Sliders and input boxes can be used to enter translation components. Values are expressed in percent of the total width and height of the bounding box surrounding the displayed structures.

Zoom Defines the scaling factor of the display. At the default value (100%), the displayed structures are completely contained within the canvas and occupy most of its extension.

Light Defines the intensity of the light source. The light source has three components. The **Specular** component only applies to structures with a surface shininess defined.

Background Opens a dialog wherefrom the user can select the background color of the display.

References

Andrey, P., Maschino, E., and Maurin, Y. (2008). Spatial normalisation of three-dimensional neuroanatomical models using shape registration, averaging, and warping. In *Fifth IEEE International Symposium on Biomedical Imaging (ISBI'08): From Nano to Macro*, pp. 1183–1186, Paris.

Biot, E., Crowell, E., Höfte, H., Maurin, Y., Vernhettes, S., and Andrey, P. (2008). A new filter for spot extraction in N-dimensional biological imaging. In *Fifth IEEE International Symposium on Biomedical Imaging (ISBI'08): From Nano to Macro*, pp. 975–978, Paris.

Bookstein, F. L. (1989). Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**, 567–585.

Burguet, J., Andrey, P., Rampin, O., and Maurin, Y. (2009). Three-dimensional statistical modeling of neuronal populations: illustration with spatial localization of supernumerary neurons in the locus coeruleus of quaking mutant mice. *Journal of Comparative Neurology*, **513**, 483–495.

Burguet, J., Maurin, Y., and Andrey, P. (2011). A method for modeling and visualizing the three-dimensional organization of neuron populations from replicated data: properties, implementation and illustration. *Pattern Recognition Letters*, **32**, 1894–1901.

Douglas, D. H. and Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, **10**, 112–122.

Maschino, E., Maurin, Y., and Andrey, P. (2006). Joint registration and averaging of multiple 3D anatomical surface models. *Computer Vision and Image Understanding*, **101**, 16–30.