

SPIERSedit 2.14

*Serial Palaeontological
Image Editing and
Rendering System:*

Specimen Interpretation and Editing Tool

Users Manual

Coding: Mark Sutton

Documentation: Mark Sutton & Russell Garwood

Contents

Contents	2
Introduction.....	3
Requirements	3
System Requirements.....	3
Data requirements	4
Basic Concepts	4
How SPIERSedit Stores Data	4
Downsampling	5
Creating a New Dataset.....	5
Window Layout, Zoom, and Navigation.....	5
Threshold image and Linear Generation	7
‘Correct’ Generation.....	8
Generating Multiple Slices	8
Editing Requirements.....	9
Simple Editing.....	10
Simple Output (Rendering)	11
Saving and Opening Datasets	11
Advancing Beyond Basic Concepts	12
Masks.....	12
Concepts	12
Applying masks.....	13
Advice on use of masks	13
Masks manipulation	14
Copying Masks	15
Masks and Segments	15
Masks and Curves	15
Segments	15
Concepts	15
Working with segments.....	16
Advice on use of Segments	17
Segment Manipulation	17
Range Generation	18
Concept	18
Range Generation Panel	19
Hard and Soft Fills	20
Polynomial Generation	21
Concepts	21
Creating a Sample	22
Poly Tab Controls	22
Advice on polynomials	23
Output	23
Concepts	23
Settings Tab	24
Objects Tab	25
Output Commands.....	26
Deskewing	26
Measuring volumes	27

3D Brush	27
Curves	27
Concepts	27
Curves Panel	28
Editing Curves	29
Copying curves	29
Interpolating Curves	30
Creating Masks from Curves	30
Lock/Selection Mode	32
Slice Spacing	32
Histogram	33
Info	33
Dataset Downsampling	33
Concept	33
Changing dataset downsampling	33
Dataset downsampling for rough masking	34
Advanced Prefs	34
Update System	35

Introduction

SPIERSedit is a package for editing (i.e. preparing) tomographic (slice-based) datasets from any source, for example serial grinding, serial sectioning, or computed tomography, and exporting them as three-dimensional models. It is the core part of the SPIERS package, which includes two other separately documented programs, SPIERSalign (used to prepare data for SPIERSedit) and SPIERSview (used to view and manipulate the 3D models SPIERSedit produces).

Requirements

System Requirements

SPIERSedit has no minimum requirements as such, and will run on most systems; it has, however, not been tested on versions of Windows older than Windows XP. Performance will benefit from high processor speed and a large amount of system RAM; 2Gb is a recommended minimum. A 64-bit version is available; the 32-bit version is restricted to using 2Gb of RAM, but the 64-bit version can use all available RAM, and is hence recommended for 64-bit systems; it is likely to prove more stable when handling large datasets. Graphics card speed is not relevant (although it is to SPIERSview). A fast hard disk is recommended; SPIERSedit is I/O intensive, and hard-disk speed will be the principle constraint on speed for many bulk-file operations. Large datasets can easily exceed 5Gb in size, so hard disk size may also be an issue.

Mac OSX: OSX 10.5 or higher is required. SPIERSview makes use of the right mouse-button; on single-button Mac OSX systems use the control-click combination to simulate right-clicking.

A high quality optical mouse or digitising tablet is strongly recommended for editing work.

Data requirements

SPIERSedit can work with any dataset consisting of serial 'slice' images, so long as the resolution and scale of each image is the same, and no images are missing from the sequence. Datasets should comprise registered (aligned) serial images, ideally in uncompressed Windows Bitmap (.bmp) format. SPIERSedit can also work with datasets in Portable Network Graphic (.png) or JPEG (.jpg) format; these take up less hard-disk space, but will normally result in slower processing; JPEG datasets will be considerably more compact than Portable Network Graphic datasets, but are not recommended as JPEG compression causes artefacts which can degrade data and hence final models.

Format: Images can be 8-bit grayscale (.png, .bmp) or 24-bit colour (.bmp, .png, .jpg). Images that lack colour information (e.g. CT data) should be converted to an 8-bit rather than 24-bit format – you will not see a difference visually, but the software will run substantially faster.

Names and directory: Although SPIERSedit will load any file sequence in alphabetical/numerical order, for clarity it is recommended that images should be serially numbered with three or four digits before the file extension (e.g. 0001.bmp, 0002.bmp etc). Images must all be in the same directory (called the *working directory* subsequently). If the dataset has been prepared using SPIERSalign, this will be the 'cut' directory underlying the SPIERSalign working directory. Note that changing filenames once the dataset has been created will effectively corrupt SPIERSedit's data.

Size and cropping: Images can be any size (resolution), but a key restriction is that all images in the sequence must be the same resolution. The speed of many SPIERSedit operations will be proportional to resolution; cropping of the image sequence before loading into SPIERSedit is hence strongly recommended. SPIERSalign can be used to perform this cropping, and hence it is normal to pass even pre-registered data (e.g. from a CT scanner) through SPIERSalign before loading into SPIERSedit. Note that once editing work in SPIERSedit has begun it is not possible to re-crop images without all work being lost.

Basic Concepts

This section contains key introductory information on using SPIERSedit, and is arranged in sequence to guide a new user through the process of constructing and viewing a simple viewable model. More detailed information on the different facilities SPIERSedit provides is provided in later sections.

How SPIERSedit Stores Data

The data a user supplies to SPIERSedit is an image 'stack'; a sequence of registered images of consistent resolution.. These images are referred to as *source images*. SPIERSedit does not alter source images, which are retained in their original form as a reference; instead it creates a corresponding stack of *working images* which it manipulates and uses as the basis for 3D reconstructions. Working images are always grayscale, and may be lower resolution than the source images (see downsampling below). Each dataset has a single SPIERSedit settings file (with a '.spe' extension), which is stored in the same folder as the source images. Working images, together with other files corresponding to each source image, are stored in a subfolder with the same filename as the dataset's '.spe' file. For example a dataset with a settings file called *fossil.spe* will have store working images in a folder called *fossil*. Note that SPIERSedit *requires* that the working images folder is named in this way – if you rename either folder or settings file the program will not be able to open the dataset.

Downsampling

Before using SPIERSedit, you will require suitable data using the guidelines provided above. Prior to importing a new dataset you should consider the downsampling required. By default, SPIERSedit creates working images in at the same resolution as source images; for high resolution datasets this may drastically slow SPIERSedit and produce models that are too complex for SPIERSview to visualise. One solution is *dataset downsampling*, where resolution is divided by an integer factor. For instance, 1000x800 pixel source images might be downsampled by a factor of 4 to produce 250x200 pixel images for reconstruction purposes. Downsampling can also be useful to reduce high-frequency noise. It does of course reduce the detail level in the final model, but in some cases the extra resolution contains little real information, and many CT datasets can safely be downsampled by a factor of 2. Dataset downsampling is defined as a downsample factor within each image (XY downsampling), and a separate factor for averaging between images (Z downsampling). Note that source images are never altered by downsampling or any other SPIERSedit operation. Dataset downsampling has to be defined when creating a new dataset (the default is XY of 1 and Z of 1 - no downsampling); while it is possible to change it later on using the *Change downsampling* command on the *File* menu, it is normally preferable to get it right initially! Being able to change this downsampling at a later date allows the majority of preparation work to be done on a downsampled dataset, which can then be upsampled and regenerated. This time-saving technique is outlined later.

Creating a New Dataset

The first step in making a reconstruction is to import the image stack into SPIERSedit. To do this use the *New* command on the *File* menu. Select the source images required for the reconstruction (normally all the images in a particular directory) and click open. In the window which appears (Fig. 1) provide a file name (the more descriptive this is the easier it will be to find in the recently opened files list) and a brief description of the dataset if required. Alter downsample settings if required (see above), and tick the 'back to front' box if applicable (see below). Click OK, and after some initial processing a black rectangle will appear. This is the *threshold image* (see Fig. 2).

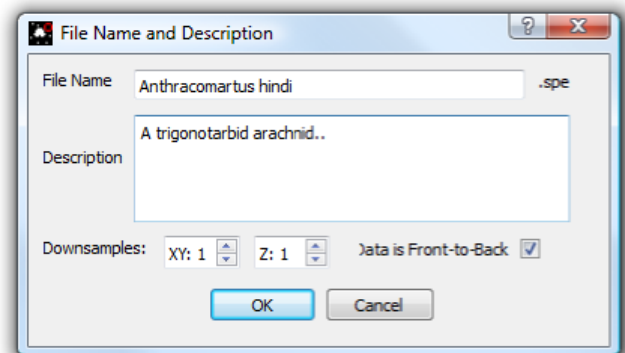


Figure 1. The new dataset window

Is my dataset front-to-back or back-to-front? Tomographic datasets come in two flavours; to understand the difference imagine looking down on top of the specimen, perpendicular to the slice-plane. In a front-to-back dataset the first (i.e. lowest numbered) image is the closest to the viewer, at the top of the specimen; subsequent images move away from the viewer (down the object). Serial-grinding datasets are normally of this type. In a back-to-front dataset the first image is *furthest* from the viewer, at the bottom of the stack; subsequent images move towards the viewer (up the object). Most CT datasets are of this type. This setting can be changed later on (using the *Output* panel), but if you get it the wrong way round your model will be mirrored!

Window Layout, Zoom, and Navigation

The components of the SPIERSedit window are shown in Figure 2 (overleaf).

On the left and right are panels; these are dockable windows containing tools, controls and information each designed for different operations. The default layout is shown in Figure 2, but panels can be dragged and placed anywhere on the screen, resized, docked in the user interface on either side (in any desired

arrangement), or on top of each other to create tabbed panels. They can also be 'floated'. Panels can be shown/hidden using commands in the *Window* Menu, or by using the keyboard shortcuts, which are: *F1* - *Main Toolbox*; *F2* - *Slice selector* panel; *F3* - *Generation* panel; *F4* - *Masks* panel; *F5* - *Segments* panel; *F6* - *Curves* panel; *F7* - *Output* panel; *F8* - *Histogram* panel; *F9* - *Info* panel.

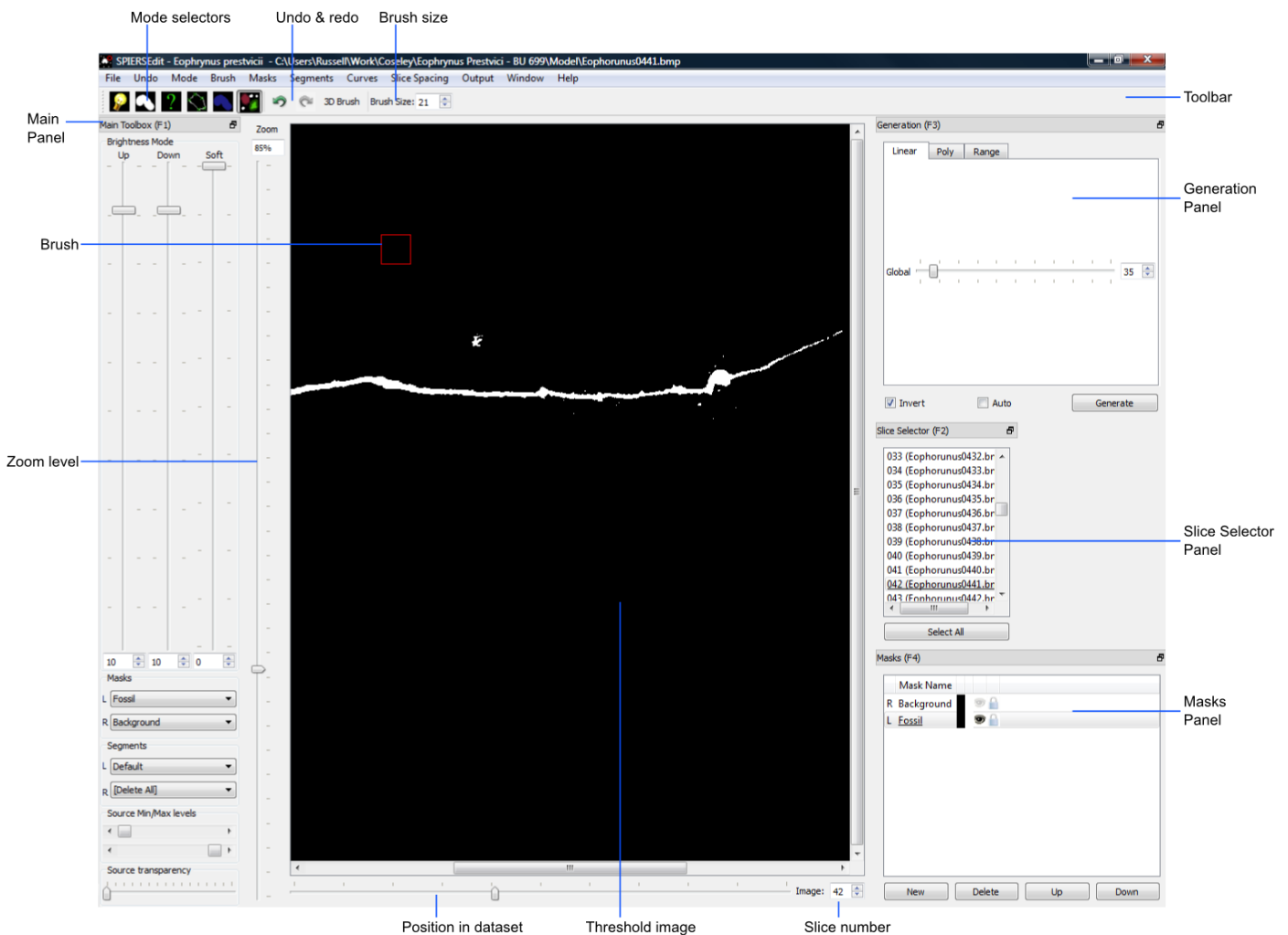


Figure 2. The SPIERSedit default window layout

The large initially black box in the centre of the screen is the *Threshold image* (see below); this may or may not incorporate scroll bars depending on the size of the image relative to the SPIERSedit window. To its left is the *Zoom level* slider, and below is the *Position in dataset* slider. The former controls the magnification of the images shown; shortcut keys *Q* and *A* also zoom in and out respectively, and zoom values can be typed into the associated 'spinbox'.

SPIERSedit only displays one image at a time from the dataset. The ',' (back, <) and '.' (forward, >) keys are used to move forward or backward through the dataset one slice at a time. The *Position in Dataset* slider allows the user to move through the dataset at will; the current slice number is displayed in the associated spinbox (labelled *Slice number* in Fig. 2); values can be entered in here to jump to slices. The title bar displays the current dataset, and image file path. The *Slice Selector* Panel, if visible, will also show the current file, as moving to a slice automatically selects it. Note however that changing selections in this panel does not move to a different image, but selects images for various operations discussed later.

At the top of the window, below the menus, is a permanently visible toolbar – this contains buttons for selecting/indicating which of its six main editing modes SPIERSedit is currently in, undo/redo controls, and brush size settings.

Threshold image and Linear Generation

The threshold is a binary (black/white) image created from the working image using a simple rule – all working image pixels brighter than mid-grey (index value 128 and above) are treated as on, representing 'object' (white), and all pixels darker than this value (127 and below) as off, representing background (black). For palaeontological datasets, 'object' normally means fossil, and 'background' means matrix (or empty space for isolated specimens). The final three-dimensional model exported to SPIERSview simply consists of all 'on' pixels.

Initially all pixels are off, as no proper working images have yet been generated; when creating a new dataset SPIERSedit simply generates black 'dummy' working images. The first step in handling any new dataset is hence to recalculate working images from source images using appropriate rules; SPIERSedit calls this process *Generation*, and the appropriate tools are found on the *Generation* panel.

To understand how working images, threshold images and simple generation work, the user is strongly advised to run carefully through the following exercise for their first dataset.

- a) Press the *Spacebar* a few times, noting that this toggles overlay of the source image. Leave the source image on, and alter the zoom with the *Zoom level* slider so the image fits comfortably onscreen. Note that the keyboard shortcuts for zoom in and zoom out are Q and A respectively.
- b) For most datasets the first slice does not contain any specimen; using the *Position in dataset* slider move to a slice that does; if in doubt move to somewhere near the middle of your dataset.
- c) Make sure the *Generation* panel is visible (see above). On this panel make sure that the tab at the top is set to *Linear*, leave all the settings at their defaults, and click the *Generate* button. Nothing will happen initially as you are viewing the source image, but if you press the *Spacebar* again a few times to flick between source and threshold images you will now see some pixels in white on the threshold image – these are 'on' (i.e. SPIERSedit has picked them as 'object' using its default rules).
- e) With the threshold image visible, try moving the *Global* slider in the *Generation* toolbox around to alter the rules used by SPIERSedit to generate the working image; you can either click the *Generate* button after each change, or tick *Auto* for changes to be automatically applied as soon as they are made (the latter is normally best, but can be slow for big images and/or slow computers). Note how more or fewer pixels can be turned on by moving the slider. If your images have objects of interest *darker* than matrix (for instance you want to image voids in a CT dataset), tick the *Invert* box – try this even if your data are not inverted.
- f) In the *Mode* menu, untick *Threshold*. This stops SPIERSedit thresholding the image, and shows the underlying working image – alter generation setting again and note how they alter this image. At all times, the threshold image is simply a version of this working image with all pixels darker than mid-grey off, and all pixels lighter than mid-grey on. In normal use you never turn thresholding off, but it is important to understand how the working image underlies the thresholded image. When you are happy, turn thresholding back on.
- g) So far we have only generated a working image for one slice. Normal procedure is to determine 'correct' settings using a typical slice (see below), then generate working images for the entire dataset. To do this, first ensure the *Slice Selector* panel is visible, click the *Select All* button in this panel. Note that all slices are now underlined – All SPIERSedit panels used the convention that underlined = selected. Now click the *Generate* button in the *Generation* panel. It may take a few minutes for SPIERSedit to generate working images for the entire dataset. When it has done so, you should be able to move through the dataset and check that all images now have non-blank working images

The *Generation* panel includes two other tabs, *Polynomial* and *Range*; these provide more complex rules for generating working images, and are dealt with under advanced topics. For most datasets, however,

linear generation is adequate. Note that for monochrome datasets (e.g. CT), only the *Global* slider is available, and essentially just controls brightness of the working image. For colour datasets (e.g. from serial grinding) there are also three values for Red, Green and Blue; these are used to weight the contribution of each of the primary colours to the monochrome working image. In most colour datasets there is no need to alter these values from their defaults, but they may occasionally be useful.

Note that while the source image files are never altered, the brightness/contrast with which they are displayed can be modified using the *Source Min/Max levels* slider in the *Main Toolbox* panel. The source and threshold images can also be combined using the *Source transparency* slider in the *Main Toolbox* panel, allowing the threshold image to be viewed below a semi-transparent source image.

'Correct' Generation

The goal of Generation is to as near as possible automatically correctly identify which pixels are 'object' and which 'background'; this involves choosing the best settings before generating for the entire dataset. Unless datasets are entirely free from noise there will be no absolutely correct setting, and the goal is to find a setting that is 'about right' – i.e. where the edge of the object is being correctly identified as much as possible, and as little background as possible is coming out as white. Toggling between the source and

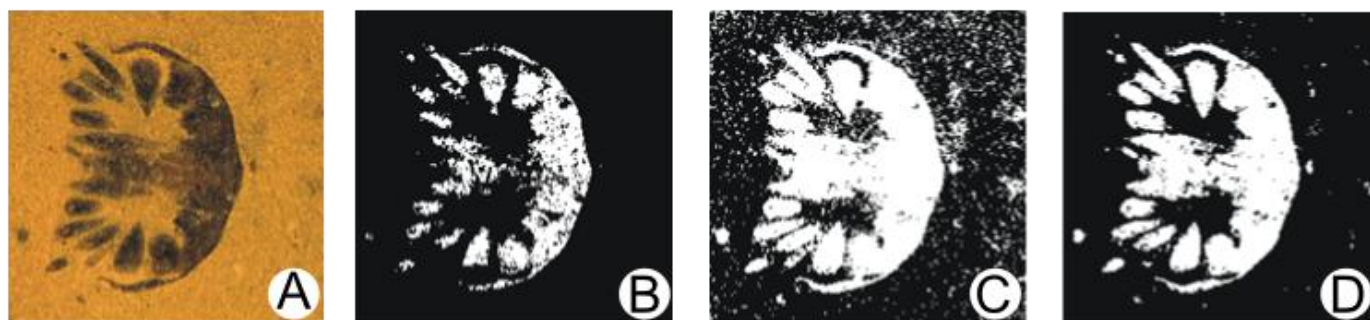


Figure 3. Linear Generation examples. A; source image. B; too dark.(to few white pixels) C; too light (too many white pixels). D; about right.

threshold image by using the spacebar is helpful while judging this. Figure 3 shows a colour slice-image from a serial grinding dataset (A), and three attempts at generating a threshold image from it (B-D). Here the fossil is darker than the matrix, so *Invert* is on. In the first attempt (B), the *Global* slider is set too high; although little or none of the matrix is white in the threshold image and much of the fossil is absent. In the second attempt (C) the *Global* slider is set too low – although most of the fossil is white in the thresholded image, far too much non-fossil material is also coming through. The final attempt (D) is about right – most of the fossil is white and only a little non-fossil material is present.

Generating Multiple Slices

Clicking *Generate* (or altering generation settings with *Auto* ticked) generates new working images for the selected slices. Normally only the currently visible slice is selected, but arbitrary sets of slices can be selected using the *Slice Selector* panel. For simple datasets (e.g CT) it is only normally necessary to do this once, to all slices, but in some datasets it may be necessary to use different thresholds for different slice ranges. For example, in CT, if the strength of the source varies towards the edge of the detector panel and the first 100 slices and last 100 would need a different global to those in between. For another example, suppose that in a 100-slice serial grinding dataset the lighting conditions had to be changed between slices 50 and 51. In this case, you would generate in two batches – first find good settings for image 1-50, then select those and generate for them. Next, move to image 51 (or anywhere else in the second half), find new settings appropriate for this image, then select images 51-100 and click *Generate* again.

Editing Requirements

Almost universally, datasets will need cleaning (here referred to as editing); without any such attention, undesirable scattered 'on' pixels and other noise (caused by cracks, artefacts, or other imperfections) will all be rendered into the final model as floating dots, planes, etc. Some relatively faint structures may not be picked out by simple generation rules, and require manual intervention to be identified. Figure 4 shows some typical problems that could be corrected with editing.

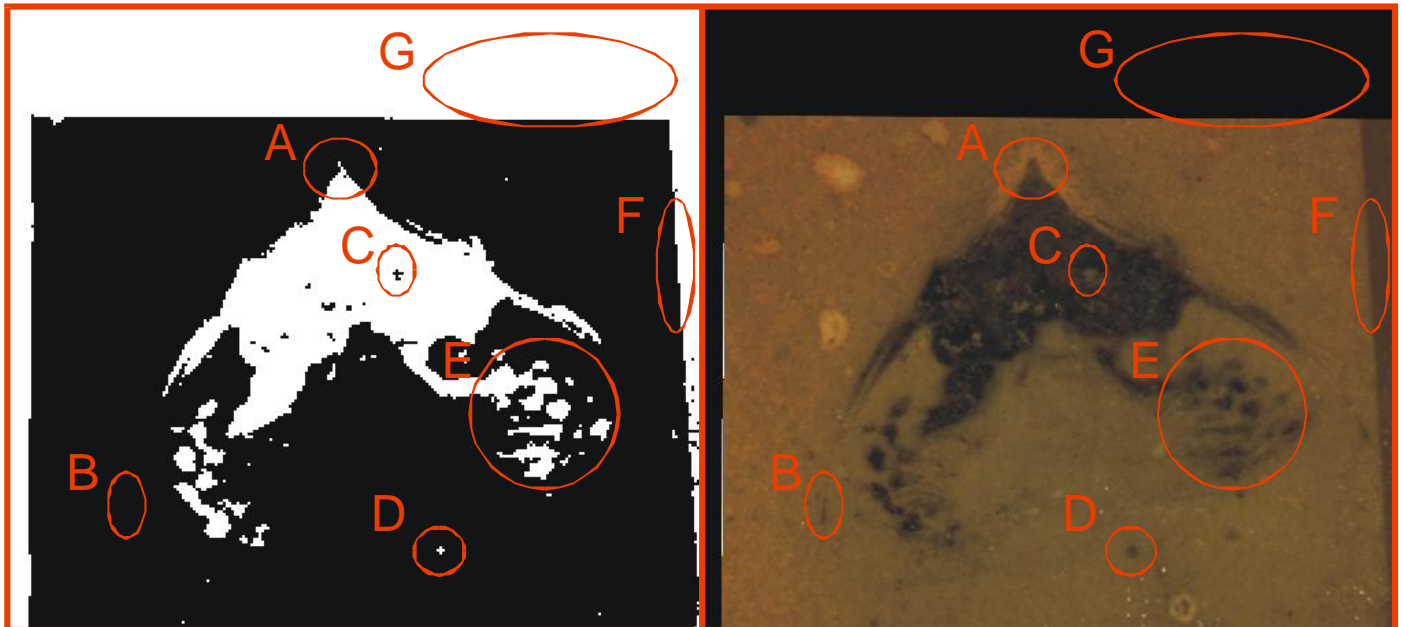


Figure 4. An image requiring editing. Source image is shown on right and thresholded image on left. A; lighter than normal structure not fully picked out. B; Thin structure not picked out at all. C; lighter area of fossil-fill identified as matrix. D; dark blob in the matrix (noise) identified as fossil (this identified by eye as non-fossil by tracing it through several images, and confirming that it does not attach to the rest of the specimen). E; set of structures appearing a 'fatter' than they should and merged into each other. F; darker background material misidentified as fossil. G; edge-padding introduced during alignment misidentified as fossil.

SPIERSedit does not *require* editing work to be undertaken; once an initial generation of working images has taken place, any dataset can be output (visualised) – see below. An initial visualisation prior to any editing work is in most cases strongly recommended to better assess the quality of the model possible, and provide a clearer picture of structures which are hard to identify in slice images.

Editing work can be time consuming, especially if done entirely manually with the brush, and involves a degree of interpretation, which could be considered to reduce the objectivity of the data. However carefully edited datasets produce cleaner-looking three-dimensional models, and more importantly can bring out anatomical detail not apparent in 'raw' unedited reconstructions, for example thin or impersistent structures. The degree of editing required will depend on the quality of the data, the time available, and the use intended for the resulting model.

Important Note: Noise consisting of small isolated objects (not joined to the main specimen, e.g. error type D in Fig. 4) can be removed automatically using the *Island Removal* facility of SPIERSview; in many cases this is a much quicker approach than editing the noise out in SPIERSedit.

Simple Editing

Editing is undertaken by dragging the *Brush* over the image to alter it. Editing is a per-slice process – what you do only affects the visible slice. It can be performed with or without the source image overlay. The brush appears as a red square or circle (see Fig. 2) superimposed on the image at the current mouse position. Brush options are found in the *Brush* menu; these include ten preset sizes and a toggle to change between a circular and square brush. Brush size can also be set using the Brush Size spinbox on the menu bar (see Fig. 2). The ‘3D Brush’ setting allows edits to affect multiple slices, and is described later in this document.

The effect of the brush depends on the current mode, indicated (and set) in the mode menu and by the toggle buttons on the toolbar (see Fig. 5). Mode can also be changed using the keyboard shortcuts, which are *Ctrl-B* (Brightness mode), *Ctrl-S* (Segment mode), *Ctrl-R* (Recalc mode), *Ctrl-C* (Curve mode), *Ctrl-L* (Lock/selection mode) and *Ctrl-M* (Mask mode). Brightness, recalc and segment mode are described here (the latter only briefly); a full treatment of segment, Curve, Lock/selection and Mask modes is given later in this document.

Brightness mode: allows manual cleaning of data by locally adjusting brightness level of the underlying working image. Brightening (left click / drag) the area under the brush can ‘push’ certain pixels above the threshold level, i.e. turn them ‘on’; this is the best method for ‘bringing out’ structures that are not appearing in the thresholded image as they are a little too dark in the working image (e.g. errors A and B in Fig. 4). Figure 6 shows the effects of brightening – on the left is the brush about to brighten the threshold image shown, and on the right is the threshold image following a single left drag with the brightness brush. Conversely, darkening (right click / drag) will do the opposite, and is the best method for handling regions where too much material is ‘on’ (e.g. error E in Fig. 4).

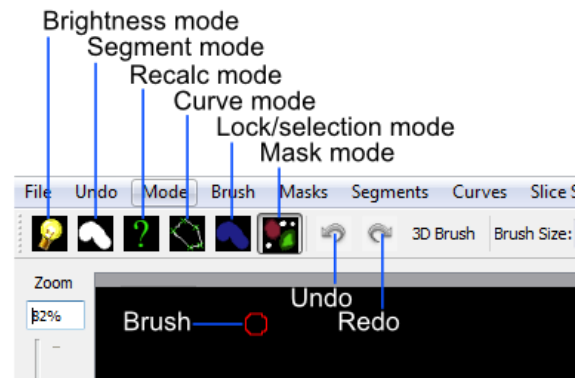


Figure 5. Mode selection toggle buttons.



Figure 6. Effects of brightness brush.

The strength of the brightening and darkening effect from a single brushstroke can be modified using the *Up* and *Down* sliders in the *Main Toolbox Panel*; repeated brush strokes over an area (releasing mouse button between strokes) will strengthen the effect. The brush effect can be ‘feathered’ so it is stronger nearest the brush centre – the *Soft* slider in the *Main Toolbox Panel* controls the strength of this feathering effect.

Segment mode: for simple (single-segment) datasets, acts as an ‘on’ and ‘off’ drawing tool; the left mouse button turns pixels on, and the right turns them off. More complex uses of this mode for multi-segment datasets are dealt with below. Left-clicking in segment mode (turning pixels on) could for instance be used to cure error type C in Figure 4, and right-clicking in segment mode (turning pixels off) could be used to cure error types D, F and G. Note that other approaches exist for removing large blocks of material; see *Masks* section below.

Recalc mode: This re-generates the pixels under the brush using the current generation settings in the *Generation Panel*, allowing manual application of different generation rules to isolated areas, or a reset of an area to an unedited state.

Undo: the toolbar also incorporates undo controls; *Ctrl-Z* is the shortcut for undo, and *Ctrl-Y* is the shortcut for redo. Undo works for editing actions performed using the brush, but not for operations that affect entire slices or groups of slices (e.g. generation).

Tips: editing successfully using the segmentation and especially the brightness brush is an acquired skill, practice will considerably speed a user up! An important trick is to have one hand on the mouse and the other on the keyboard to constantly overlay and remove the source image as a guide (using the spacebar), as well as to switch brush modes and sizes.

Simple Output (Rendering)

SPIERSedit does not itself handle 3D modelling, but exports its datasets to SPIERSview for viewing in three dimensions. At the simplest level this is done by using the *View in SPIERSview* command on the *Output* menu – *F12* is the keyboard shortcut. This command initiates an export (which may take an appreciable amount of time), then launches SPIERSview on the exported file. SPIERSview has a separate manual.

In most cases however you will want to review output settings before using the *View in SPIERSview* command to ensure the model is displayed correctly; these are available in the *Settings* tab of the *Output* Panel. The *Slices/mm* and *Pixels/mm* boxes are the most important of these settings, specifying the scale and aspect ratio of the output model. *Slices/mm* is the number of slice images per millimetre in the source dataset; this will be $1/s$, where s is the slice-spacing in millimetres. For example if slices are spaced every 40 microns, *Slices/mm* should be $1 / 0.04 = 25$. *Pixels/mm* is the scale of each (source) slice image, specified as the number of Pixels per millimetre – for instance if the field of view is 3mm and the source image is 300 pixels wide, *Pixels/mm* should be 100. For CT data and these two values are normally the same, and can be calculated as $1 / v$ where v is the voxel size (converted to millimetre).

The *Sequence front to back* tick box is the same ‘front to back’ setting discussed above under *Creating a New Dataset*.

Other output settings are described later in this manual.

Saving and Opening Datasets

Saving in SPIERSedit is essentially automatic. Changes made to individual images are made directly to disk. Other information, stored in the ‘.spe’ settings file, is automatically saved when SPIERSedit exits and autosaved by default every five minutes (the *Advanced prefs* dialog, accesible via the *File* menu, can be used to change this default). A manual settings save can be triggered at any time using the *Save* command on the *File* menu.

SPIERSedit datasets can be re-opened (a) by double-clicking on the SPIERSedit settings file (.spe), (b) by using the *Open* command on the *File* menu to open an existing .spe file, or (most conveniently) by using the *Open Recent* submenu on the *File* menu. The *More...* command at the bottom of the *Open Recent* submenu shows all datasets ever opened by this installation of SPIERSedit.

SPIERSedit does not support multiple datasets – opening a dataset with the *Open* command will save and close any dataset already open. Multiple copies of SPIERSedit can be opened instead, though be aware that SPIERSedit uses a substantial amount of memory for its file cache (see *Advanced Prefs* section below), and hence that multiple copies may quickly use up all available system memory.

The *Save As* command on the *File* menu creates a second copy of the entire working dataset for backup or other purposes. It does not duplicate the source data files, but creates a new ‘.spe’ file and working dataset folder within the source dataset directory.

The *Import SPIERSedit 1.1* command on the *File* menu is not documented in this manual.

Advancing Beyond Basic Concepts

The remainder of this manual documents many other features of SPIERSedit. It is recommended that users intending to make extensive use of SPIERSedit read through all these sections, but for those in a hurry the most important sections (in no particular order) are *Masks* (which allow splitting of a model into colour-coded parts), *Segments* (which allow for multiple types of material in a model), and *Output* (which covers how to export and view models with multiple masks and segments).

Masks

Concepts

The SPIERSedit mask system allows the user to colour-code different parts of the model. The two most common reasons to do this are:



Figure 7. Mask as region of interest

Defining a region of interest. While it is possible to manually delete all undesired pixels in the background/matrix in each slice, the mask system provides a quicker way to separate off the part of the image with 'object' in it from the rest. In Figure 7, a 'fossil' mask (light brown) has been used to indicate the region of interest prior to any editing work. The remaining background (still in the black/white colour of the default mask) can then be hidden and excluded from visualisation with a few clicks. A similar approach can be used to identify unwanted structures such as cracks, and exclude them from visualisation.



Figure 8. Masks used to identify structures

Splitting a model into sub-units: The most important application for masks is to split the model into several separately coloured units (e.g. split an arthropod fossil into carapace, appendages and trunk). This enables colour-coding for clarity in the final three-dimensional model, and also allows the user to perform 'virtual dissections' by interactively hiding/unhiding these units in SPIERSview. In Figure 8, a 'body' mask (light brown) has been applied prior to editing (as above), and the background mask hidden from view. After editing, further masks were applied to reassign some parts of 'body' to other structures - a carapace mask (blue), and three different appendage masks (purple, red and green) are visible.

Masks are only visible in *Mask* mode; however if the *Always Show Masks* option in the *Mode* menu is turned on, you will see masks (in a washed-out form) in all other modes (except Lock/selection).

Note that masking affects both 'on' and 'off' pixels – a light version of the colour is used for 'on', and a dark one for 'off'. Only the 'on' pixels will appear in the final model of course, but the colour-coding of 'off' regions is there to remind you of the mask that pixels subsequently turned on here will be assigned to. Note that the mask-boundaries in empty regions of the examples above are rough and ready - this doesn't matter, as there are no 'on' pixels here to appear in any reconstruction. Those in 'on' regions (such as the inside of the carapace, above) will be rendered as object boundaries, so more care is required.

Colour: Mask colour only affects the colour masks appear in SPIERSedit; while this colour is also used to provide a default colour for output objects when they are created, these can be set independently, and

there is no requirement for SPIERSedit mask colours to be those used for objects in SPIERSview; mask colours instead should be chosen so that the user can easily distinguish masks which abut each other when editing. Dark 'on' colours should normally be avoided.

Visibility: Masks can be visible or hidden. Hidden masks are still visible in the threshold image as their 'off' colour, but no 'on' pixels are shown. This only affects viewing of images in SPIERSedit; hidden masks behave just like visible ones for export to SPIERSview. Hiding masks only affects program behaviour if the *Hidden masks locked for generation* option in the *Masks* menu is ticked; as expected this option restricts all generation actions performed from the *Generate* panel to visible masks. This allows different generation rules to be applied for different regions of interest – often useful if there are preservational differences between different structures. Note that the recalc mode brush still affects hidden masks even with this option ticked.

Locking: Masks can be locked or unlocked. A locked mask cannot be overwritten by another mask, either with the mask brush, mask copy commands, or through mask from curve operations (see below). There are many uses for this facility, the most important being the locking of a 'completed' mask to avoid accidental changes to it.

Applying masks

Masks are applied by using the brush in *Mask* mode. The left and right mouse buttons can be set to apply different masks; these are chosen from the L and R mask dropdowns in the *Main Toolbox* panel, or alternatively by left or right clicking in the left-hand column of the *Masks* panel. In this column the R indicates the right mouse button mask and the L indicates the left mouse button mask; if these coincide a 'B' (for both) is displayed. Note additionally that selecting a mask in the *Masks* panel automatically chooses it as the left-mouse mask.

Masks can also be applied using the *Masks from Curves* and *Copy* commands (see below); these provide mechanisms for rapidly masking large numbers of slices.

Advice on use of masks

Masking procedure: When masking a dataset it is normally best to begin masking an isolated part, such as a limb. Find such a structure, and adjust the brush size, create and select a mask, and apply it using the brush to create a coloured zone around a structure to encompass all its boundaries (see Fig. 9). Moving forward or backward through the dataset a slice at a time (most conveniently using the shortcut keys for movement) allows this process to be repeated and structures followed throughout the dataset. It is normally easier to create one mask at a time and apply it to all required slices, rather than create all masks at once and apply them all, slice by slice.

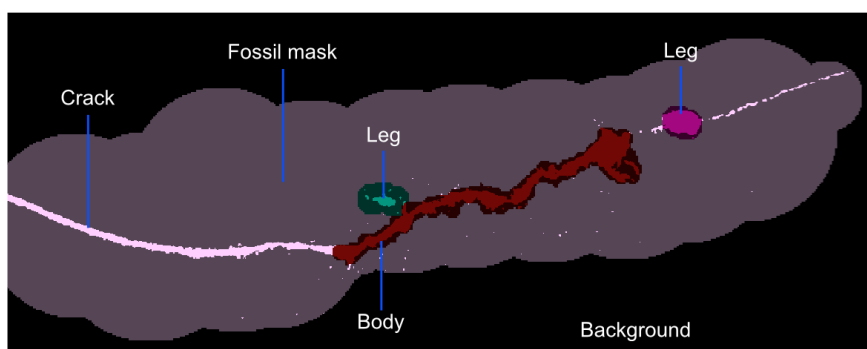


Figure 9. Masking example. Here a rough 'Fossil' mask has been used to pick out a region of interest, facilitating initial inspection of data and removing much of the background (hidden). Three subsequent

masks (two legs and a body) have subsequently been more carefully added; at this point only the crack and noise are left in the 'Fossil' mask.

Multiple masks: Multiple masks can be easily fused into a single object at output, so there is little harm in using multiple masks for a single structure. This often provides flexibility – for instance part of a structure can be masked separately so that in one version of the output it can be hidden to allow users to see inside an object; in another version of the output it can be fused seamlessly with the rest of the object. As another example 'left' and 'right' versions of arthropod appendages can be separately masked, and then either fused together at output or output separately, as desired. In short, it is far easier to join objects at output than it is to split up a single mask later on, so the use of many masks for all potentially separable structures is recommended.

Mask cut-offs: Where structures converge (e.g. a leg meets the body) the user must decide at which point to switch masks; this is an arbitrary decision, but if not made consistently on subsequent slices can result in ragged-looking 'cuts'. One of the best ways to attain consistency is the use of the *Masks from Curves* command (see below).

Masks manipulation

The *Masks* panel lists all masks that exist for the currently open dataset. When a dataset is created a single mask called 'Background' is created, with all pixels assigned to it. Figure 10 shows an undocked *Masks* panel with many masks visible. Most changes to masks are carried out through this panel; a few also use the *Masks* menu.

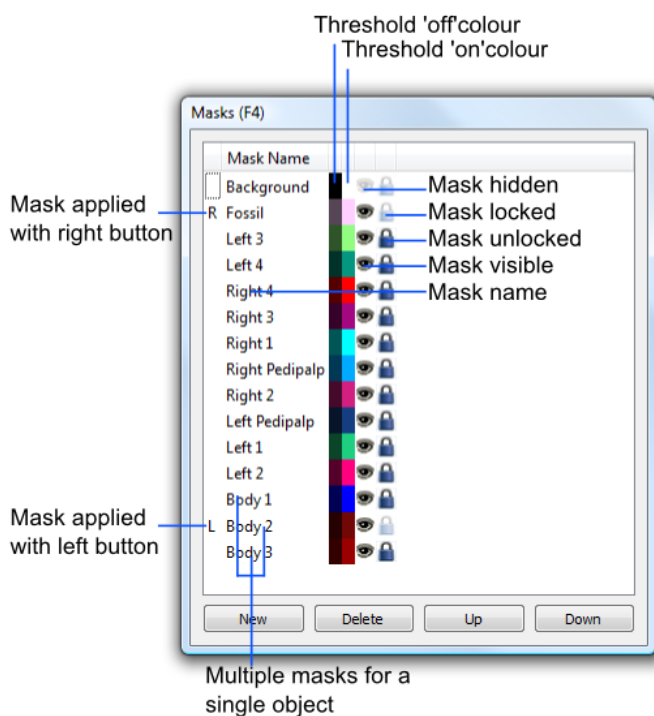


Figure 10. *Masks* panel

Creating masks: The *New* button in the *Masks* panel (or the *New Mask* command in the *Masks* menu) creates a new mask. Up to 255 masks in total can be created. New masks are created with an uninformative name (e.g. Mask 2), and a random colour; it is good practice to at least rename a mask after creation.

Renaming masks: Double click on the Mask Name in the *Masks* panel to edit it.

Changing colours: Double click on the right-hand colour block (the Threshold 'on' colour) to change it. The left-hand colour block (the Threshold 'off' colour) is not set independently, but is a darker version of the 'on' colour. Double-clicking this block will bring up a dialog which enables the contrast between the light and dark versions of the colour to be set.

Selecting masks: One or more masks can be selected by left clicking on any column of the *Masks* panel. To select multiple masks use Ctrl-click or Shift-click. Selection is indicated by an underlined mask name. Note that mask selection and choice of left mouse button mask is not quite the same thing, though often they will coincide – it is possible for the selected mask NOT to be the left mouse button mask for instance, and more than one mask can be selected. Selection of masks is used for bulk locking or hiding, bulk deleting, mask copying, and the creation of output objects.

Mask visibility: Double-clicking a mask's 'eye' icon toggles its visibility; this can be done to masks in bulk by selecting them (see below) and using the *Show Selected Masks* or *Hide Selected Masks* commands on the *Masks* menu.

Mask locking: Double-clicking a mask's 'padlock' icon toggles its lock status; this can be done to masks in bulk by selecting them (see below) and using the *Lock Selected Masks* or *Unlock Selected Masks* commands on the *Masks* menu.

Re-ordering masks list: Masks can be moved up and down the list by selecting a single mask and using the *Up* and *Down* buttons on the *Masks* panel. This reordering only affects how the masks appear in this list; it has no effect on output or images.

Deleting masks: To delete a mask or masks first select them (see above), then click delete or use the *Delete selected mask(s)* command on the *Masks* menu. As all pixels MUST be assigned to a mask, SPIERSedit brings up a dialog to determine which mask you want to assign pixels in the to-be-deleted mask(s) to. The delete operation may take an appreciable amount of time as SPIERSedit goes through all the slices and changes all instances of the deleted mask(s) to the target mask.

Copying Masks

SPIERSedit provides four different *Copy* commands in the *Masks* menu to copy masks from or to the previous, next, current or selected slice(s); in this context current = viewed slice; next = slice after viewed slice; previous = slice before viewed slice; selected = slice(s) selected in the *Slice Selector* panel. These commands copy all masks selected in the *Masks* panel from the source to the target slice(s). Ticking the *Advance one slice after copy* or *Go back one slice after copy* options in the *Masks* menu combines copy operations with a single move operation. Using these options in conjunction it is possible to copy masks one slice at a time quickly through the dataset with a single command for each step. Alternatively, using the *Copy... to selected* commands it is possible to copy mask data from one slice to many in a single command.

Masks and Segments

If the *Segment brush applies mask* option on the *Masks* menu is ticked then drawing segments on using the brush (with either mouse button) will also apply the selected masks to the same pixels.

Masks and Curves

The *Mask from curve* command on the *Masks* menu allows the user to generate masks quickly for large numbers of slices, and is normally the most effective (though not the simplest) way to specify masks over a large number of slices. Its use is described in the section on curves.

Segments

Concepts

Multi-segment datasets: In the Basic Concepts section we assumed that data could be modelled simply by splitting into two 'segments' - background ('off') and fossil ('on'); these are called single-segment datasets in SPIERSedit (background is a special case, so not counted as a segment). In some datasets however the situation is more complex; for instance in Figure 11A two different types of material are present in the fossil (light and dark grey). This is best modelled as a multi-segment dataset – there are two types of material (recall that background does not count), so it they requires two segments in SPIERSedit, rather than the default one. In a multi-segment dataset each source image has more than one corresponding working

image, one for each segment; each segment has independently specified generation rules. The threshold image no longer consists of pixels simply 'on' or 'off'; instead each pixel is assigned to one of the segments (or none, background). The threshold image is calculated from the working images using the following simple rules: if a pixel is below threshold level for all segments, it is off (black); If it is above threshold level in only one segment it is assigned to that segment; if it is above threshold level in more than one working image it is assigned to the segment in with the lightest shade of grey. Figure 11B-D shows two working images and the corresponding combined threshold image.

Segments and Masks: The segment and mask systems are independent and complementary. Each pixel in each slice assigned to one segment (or none, i.e. background, 'off') AND one mask. Segments are primarily used to pick out different types of material, masks to pick out regions. The way in which masks and segments interact to produce renderable objects is discussed in the *Output* section below.

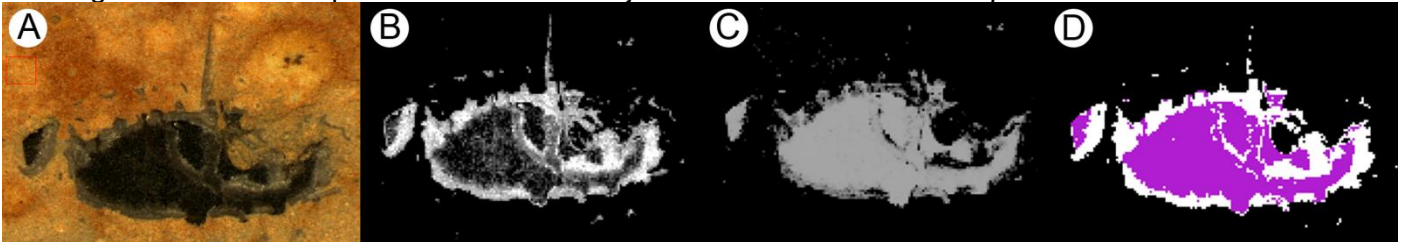


Figure 11. Multi-segment dataset. A; source image (starfish arm section comprising two distinct materials). B; working image for first segment (lighter material round edge of fossil). C; working image for second segment (darker material in core of fossil). D; Threshold image showing pixels identified as light segment (white), dark segment (purple) or neither (black).

Display colour: Segment colour is visible in all modes except mask mode, where mask colour is used instead, If however the *Always Show Segments* option in the *Mode* menu is turned on, you will see segments (in a washed-out form) in mask mode too. As with mask colour, segment colour is simply the colour that segments appear in SPIERSedit, and has no influence on output colour.

Inactive and locked segments: Like masks, segments can be locked or hidden, though showing/hiding is termed activation/inactivation for segments to emphasise the difference in function. Making segments inactive essentially deletes them; SPIERSedit treats inactive segments as though they do not exist. Unlike true segment deletion of course, a hidden segment can be restored with a single operation. Locking a segment stops all segment or brighten mode brush operations from affecting any pixels currently assigned to that segment.

Working with segments

Segment selection: The upper of the two drop-downs in the 'segments' section of the main toolbox is used to choose the segment you are working with at the moment. This affects the operation of slice generation, brighten mode and segmentation mode. The lower of the two drop-downs only affects the right mouse button in segmentation mode. As with masks, these choices are displayed as L and R (or B) in the leftmost column of the *Segments* panel, and left/right clicking in here will select them. Note that the R drop-down also has a 'delete all' option – if set to this, the right mouse will remove pixels from all segments (i.e. turn them off).

Brush in segment mode: In segment mode the brush is used to assign pixels directly to the selected segment (left mouse button) or segment specified in the 'R' drop-down (right mouse button). Note that the R drop-down also has a 'delete all' option – if set to this, the right mouse will remove pixels from all segments (i.e. turn them off).

Brush in brighten mode: The brighten brush only affects the selected segment. Note that brightening in a multi-segment dataset can not only bring pixels over the threshold level so they turn on (as before), but can also make them brighter than the corresponding pixels in other segments so that the pixel changes to the selected segment.

Brush in recalc mode: This only affects the selected segment.

Generation and segments: Generation rules are stored on a per-segment basis. If using linear generation for instance, one segment can use a rule with *Invert* on to pick out dark pixels, while the other uses a rule with *Invert* off to pick out light pixels. Selecting a different segment (see above) will flick the settings of the *Generation* toolbox to those setup for that segment for *Linear* and *Polynomial* generation. *Range* generation works in a different way, specifying a single set of rules applied to all segments (see below).

Advice on use of Segments

Generation method: For datasets based on colour source images, carefully chosen *Linear* generation settings and/or *Polynomial* generation are often the best way to generate working images for multi-segment datasets. For datasets based on monochrome source images however (e.g. CT data), multi-segment datasets are best handled with *Range* generation (see below).

Using segments to fill holes: A common task with fossil datasets is to generate a model of voids in a certain part of the model, for example inside a fossil otherwise reconstructed with a single segment. The segments system allows a simple trick to be used to handle this. First, the normal segment is created and properly edited (if desired). Once prepared, it is locked and a second 'fill' segment created. No slice-based generation is performed on this second segment – slices are left blank. To fill the holes, the first segment is locked, and the 'fill' segment then brushed roughly over the holes using the segment brush.

Number of segments: There is no formal limit on the number of segments, but the speed of many operations in SPIERSedit is inversely proportional to the number of segments in use; furthermore the disk-space used by the dataset is roughly proportional to the number of segments. In most datasets only two or three segments will be required; users are advised to avoid going beyond three unless they are sure they know what they are doing.

Segment Manipulation

Segment manipulation is primarily performed through the *Segments* panel, which is very similar to the *Masks* panel both visually and in operation. Some operations are performed instead through the *Segments* Menu.

Creating segments: The *New* button in the *Segments* panel (or the *New Segment* command in the *Segments* menu) creates a new segment; this will take an appreciable amount of time as new working images will need to be created for each source image. These are initially created blank (as are new working images for a new dataset).

Renaming segments: Double click on the Segment Name in the *Segments* panel to edit it.

Changing colours: Double-click the colour block to change it.

Selecting segments in the panel: One or more segments can be selected by left clicking on any column of the *Segments* panel. To select multiple segments use Ctrl-click or Shift-click. Selection is indicated by an

underlined segment name. Note that segment selection in the panel and selection in the drop-down in the *Main Toolbox* panel are not the same thing. Selection of segments in the panel is used for bulk locking or activating/deactivation, bulk deleting, segment copying, and the creation of output objects.

Segment activation: Double-clicking a mask's 'eye' icon toggles its activation status (visibility); this can be done to segments in bulk by selecting them (see above) and using the *Activate Selected Segments* or *Deactivate Selected Masks* commands on the *Segments* menu.

Segment locking: Double-clicking a segment's 'padlock' icon toggles its lock status; this can be done to segments in bulk by selecting them (see above) and using the *Lock Selected Segments* or *Unlock Selected Segments* commands on the *Segments* menu.

Re-ordering segments list: Segments can be moved up and down the list by selecting a single mask and using the *Up* and *Down* buttons on the *Segments* panel. This reordering only affects how the segments appear in this list; it has no effect on output or images.

Deleting segments: It is not normal to delete a segment unless it has been created in error; normally segments that are not required are simply deactivated. Deleting a segment removes its associated files and cannot be undone. To delete a segment or segments first select them (see above), then click delete or use the *Delete Selected Segment(s)* command from the *Segments* menu.

Range Generation

Concept

Range generation is a generation mode primarily intended for multi-segment monochrome (e.g. CT) datasets, although it can also be used on colour datasets. In range generation the user splits up the range of possible greyscale values (from black to white) in the source image into sections corresponding to different types of material, and assigns segments to each. Figure 12 shows an example of one such dataset, where a fossil is preserved as both void (dark) and pyrite (white), and two segments are defined using the range system to cover the darker pixels ('Default') and the lighter ones ('Segment 2').

To implement range-based generation, the user first creates the required number of segments (see above). With the *Generation* panel in *Range* mode (the *Range* tab selected), the positions of the segments in the black-white gradient can be set using the controls detailed below; in practice it is normally best to start with a *Distribute Over Range* command to 'separate' the segments, which by default will be overlapping and not arranged in any sensible way. In the same way that linear generation for single-segment datasets is normally initiated by finding optimal values on a single 'test' slice, range generation is also best set up in this way, using the *Auto* tick box to apply changes as they are made until optimum settings have been determined, and the Range generation can be applied to the same dataset. Note that segment boundaries are allowed to overlap, but this is rarely a sensible thing to do, and should be avoided unless the user is confident they know what they are doing!

Range Generation (i.e. clicking *Generate* or using *Auto* and making changes) affects all selected slices as does normal generation, but unlike generation in *Linear* or *Polynomial* modes it also affects ALL segments, not just the selected one.

Range Generation Panel

The *Generation Panel* in range mode consists of a shaded strip running from black to white (representing the range of shades to be assigned), together with *Base* and *Top* 'spinboxes' and two tick boxes. Segments are shown on the strip occupying the range currently assigned to them; the currently selected segment (the one in the 'L' drop-down of the *Main Toolbox* panel) is shown in a lighter colour, with a green border, and the *Base* and *Top* values shown apply to this segment, representing the range on a scale covered where 0=black, 255=white.

Selecting Segments: Left-clicking on the coloured block representing the segment on the shaded strip will select it (i.e. have the same effect as selecting it in the L dropdown on the *Main Toolbox* panel).

Dragging boundaries. Using the either mouse button you can pick up and drag the left and right margins of segment blocks in the shaded strip to reposition them (i.e. change the *Base* and *Top* values of the segments). If you use the left mouse button, margins will 'snap' onto adjacent margins to ensure no gap in between, and if you drag a margin already in contact with another, both will move together. If you use the right mouse button instead, this 'intelligent' snapping and combined dragging behaviour is disabled.

Using Base and Top: With a segment selected, *Base* and *Top* values can be altered directly using the spinboxes; no snapping behaviour is implemented.

Selected only: With this box ticked generation affects only the selected segment, not all segments. This option is not normally helpful, and is included only for completeness.

Hard fill: Generates in hard rather than soft-fill mode – see below.

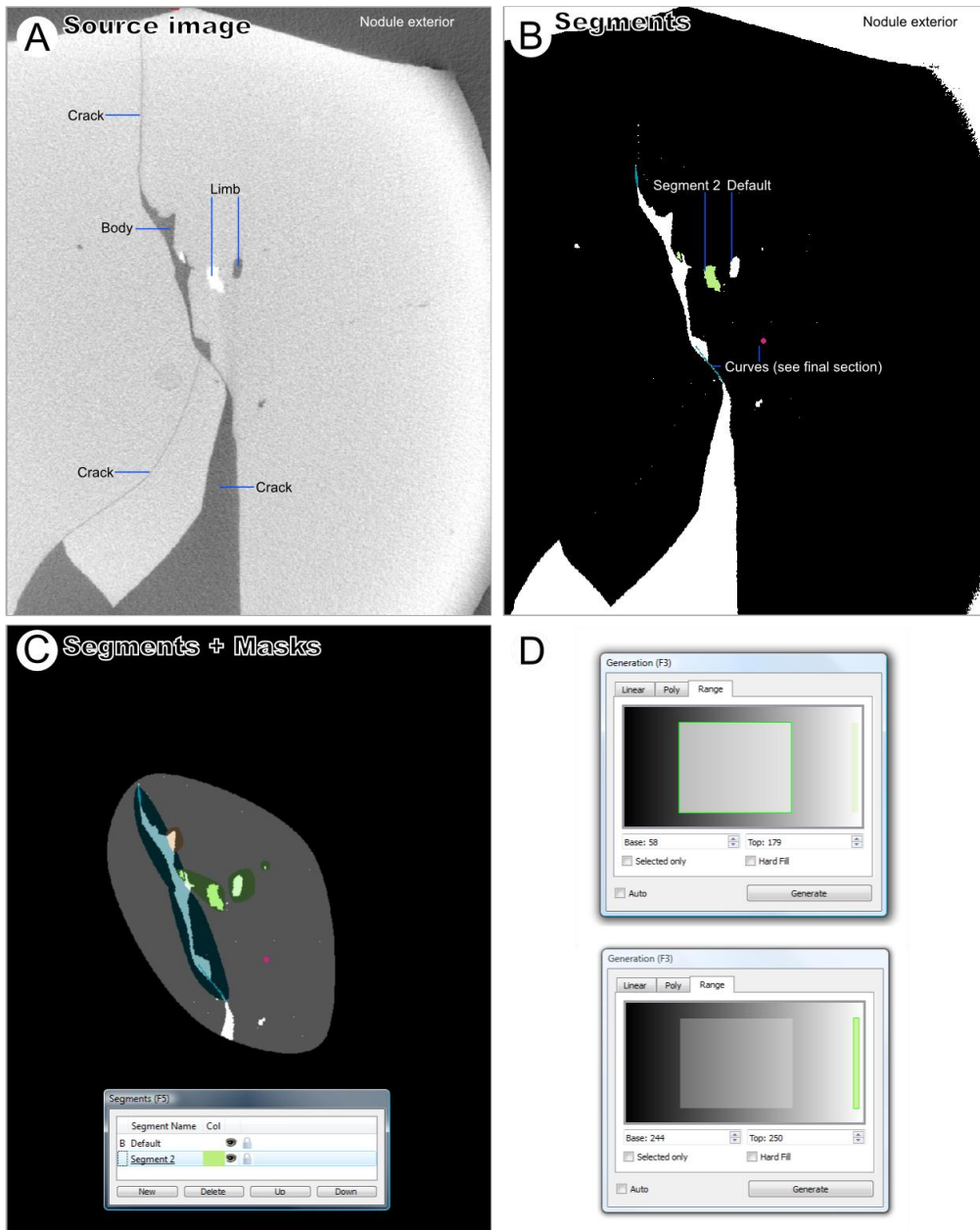


Figure 12. Multi-segment CT dataset handled using range generation. A; Source Image. B; Thresholded image showing multi-segment interpretation. C; Segmented image with masks overlaid; D, Range generation settings used, top shows 'Default' segment selected, bottom shows 'Segment 2' selected. *Distributing segments.* The *Distribute over range* command takes all segments selected in the *Segments* panel and distributes them evenly over a specified range (by default the entire range, 0 to 255). This is primarily useful when setting segments up initially, as a prelude to moving their boundaries to optimal positions.

Hard and Soft Fills

By default, *Range* generation uses a 'soft fill' algorithm to generate each working image (recall that there is one working image per segment). If source image pixels are at the midpoint of the range, the corresponding working image pixels are fully white. As source image pixel values move away from this midpoint, the corresponding generated pixels get darker linearly, crossing the threshold value (of 127) at the correct points (the specified Top and Base values for the segment), and tailing away to 0 beyond that. Figure 13 provides a visual explanation of soft-fill values for a hypothetical four-segment dataset

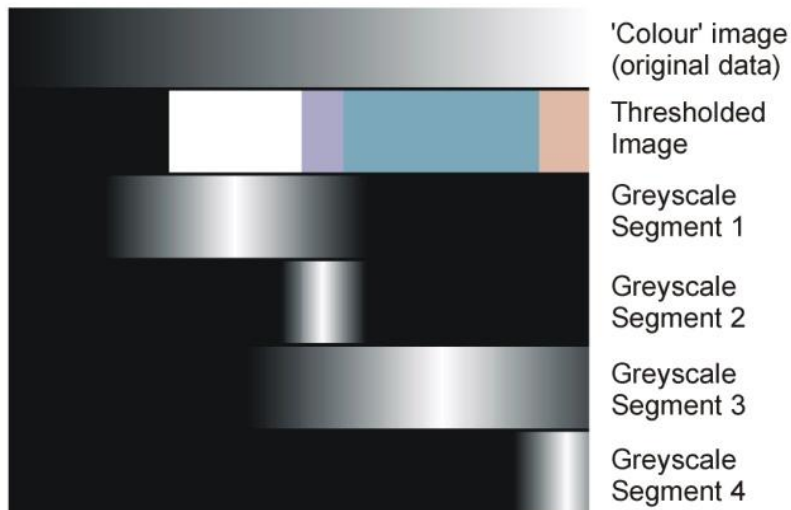


Figure 13. Soft-fill range generation. Top; the grayscale range of the source images. Second row; Segment definitions for four segments. Bottom four rows; working image levels generated for each segment for each source image level

The point of soft-fill range generation is to enable the brighten brush to be on range-generated dataset. If used on working images generated in this way, it will function to 'bring out' a segment, pushing boundaries locally up and down. This allows subtle post-editing after generation.

When generating a single segment, however, soft-fill is not always appropriate; pixels within the specified range might not display as the expected segment if another segment has been artificially brightened at that point. To ensure that all pixels in the designated range are assigned correctly, tick the 'hard fill' box. With hard-fill in operation, SPIERSedit sets all pixels in the designated range to pure white in the segment greyscale file, and all outside it to pure black, ensuring that this problem does not arise.

Polynomial Generation

Concepts

Linear generation is a relatively crude way of turning a colour image into a monochrome working image, and while *Range* generation is an effective alternative for monochrome datasets (e.g. from CT scanning), both approaches potentially miss subtleties of colouration in colour-based datasets (e.g. from serial grinding), where different materials can be picked out with colour differences rather than simply brightness level. *Polynomial* generation mode is a more sophisticated approach to this generation-from-colour problem, which can in some circumstances produce 'better' thresholds, which will need less manual editing afterwards. It is however more complex both in concept and practice, and it is recommended only for advanced users.

Polynomial generation uses the concept of *training*; the user first manually edits a region of the dataset to their satisfaction, so that all pixels in this region (which can span multiple slices) are assigned to their correct segments. SPIERSedit is then asked to create generation rules (based on mathematical polynomial equations) which would work to replicate this manual editing work as effectively as possible (i.e. would correctly predict the segment assignment in the sample for as high a percentage of pixels as possible). Once created, these rules can be applied using the *Generate* button to all selected slices (or with the *Recalc* brush) in the same way as for *Linear* generation.

Mathematical note: Polynomial mode generation rules are polynomials of the form:

$$v = k_1 + k_2r + k_3g + k_4b + k_5r^2 + k_6g^2 + k_7b^2 + \dots + k_{(3n+1)}r^n + k_{(3n+2)}g^n + k_{(3n+3)}b^n$$

where v is the greyscale value, r , g , and b are the red, green and blue values of the colour image, and n (the order of the polynomial) and k_1 to $k_{(3n+3)}$ (constants) are the values that comprise the rule.

The *Poly* tab of the *Generation* Panel provides controls for the creation of polynomial rules; with this tab selected, the *Generation* button (and the *Recalc* brush) will use polynomial mode generation rules.

Polynomial mode can be used for multi-segment datasets, but both Polynomial rule-creation and generation is performed on a per-segment basis.

SPIERSedit uses a genetic algorithm to create polynomial rules; this approach simulates Darwinian evolution to 'breed' effective rules. It starts with a random rule (or the best rule from the last time the genetic algorithm was run, if applicable) and determines what percentage of the pixels in the supplied sample would have come out correctly had that rule been used. It then makes a small random change to the rule, and checks again. If the new rule performs better (is 'fitter') it is retained, otherwise it is discarded. The process is repeated for many 'generations' until the rule reaches an optimal fitness and no further improvements appear to be possible.

Creating a Sample

Polynomial generation requires a sample of edited 'correct' pixels to build its rules. The sample used is the pixels turned 'blue' in Lock/Selection mode (see below) for all slices selected in the *Slice Selection* panel; it can thus consist of as many or as few pixels on as many slices as the user desires. SPIERSedit has no facility for storing samples; whenever 'Find Polynomial' is clicked the current selection is used, and any previous sample is discarded.

Poly Tab Controls

The *Poly* tab on the *Generation* panel (see Fig. 14) provides controls for the polynomial rule creation process.

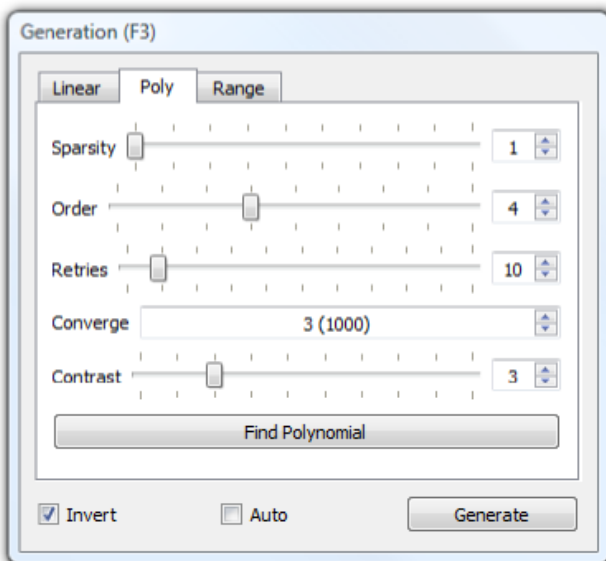


Figure 14. Poly tab of the Generation Panel.

Sparsity: Controls how many of the sample pixels are used in the breeding process. A sparsity of 1 means every pixel is used, 2 means every other pixel, 5 every fifth pixel etc. Use this to speed the breeding process, although this might result in slightly less fit rules.

Order: Strictly speaking this is not part of the breeding process, but like the other settings it should be set before breeding. It controls the 'order' of the polynomial (n in the mathematical note above). Normally you can leave this at the default value of 4. Increasing it increases the time take to breed a solution, but can (in theory at least) result in better rules. Changing this and then generating without first creating breeding new rules with '*Find Polynomial*' can result in strange behaviour!

Retries: The number of times the breeding process is run. Genetic algorithms can become stuck in blind alleys, where the rule has stopped improving, but is still far worse than it could be because a wrong direction was taken early on. To get around this the breeding process is run several times with different starting rules each time; the best solution from all runs is then taken. The catch is that the time taken for the genetic algorithm to run is proportional to the number of retries used. You should normally use at least five retries, unless you are refining an existing rule (see below). The default is 10.

Converge: The number of generations without any improvement before the algorithm gives up trying to make a rule better. High values make for slower breeding but (potentially at least) slightly better rules.

Contrast: This isn't a setting for breeding but for generation. You may find that a rule generates greyscale images with either too little contrast (where most pixels are either just brighter or just darker than the threshold level), or too much contrast (where most pixels are either pure black or pure white). This doesn't affect the threshold image, but does affect the usefulness of the brightness brush – you need a reasonable amount of contrast to play with. Changing the contrast setting to compensate for this – check how much contrast you have by generating and visually inspecting the working image by turning off thresholding (untick *Threshold* on the *Mode* menu).

Find Polynomial button: Starts breeding with the current settings. Progress is reported in a small dialog, which reports the fitness of the current ('Best') rule as a percentage, representing the proportion of pixels in the sample that the current rule predicted correctly (so high is good); values of over 95% are normally attainable. The algorithm can be left to run to completion, or can be stopped at any time with the stop button. Whether the operation is complete or has been manually stopped the best rule found will be used. Once completed or stopped, the dialog should be closed with it's 'Close' button.

Advice on polynomials

Breeding can take a long while, particularly for large samples across multiple images. The best way to speed it up is to perform a two-stage breed. Firstly, set a large number of retries (say 30 or so), but choose a low convergence (say 2000) and a high sparsity (how high depends on the speed of the computer and the size of the sample – anything that makes things go reasonably quickly). Perform a *Find Polynomial* operation with these settings, which will try a lot of different rules in a rough-and-ready way. From this you will end up with one good candidate. Now set up a careful breed without any retries (set retries to 1, sparsity to 1, and converge to 32000 or more) and perform a second *Find Polynomial* operation. This second run of the algorithm will take the best rule from the first run and give it as much chance to improve as possible. Using this two-stage approach, high quality rules can normally be generated in a few minutes at most.

Output

Concepts

SPIERSview: SPIERSedit does not directly visualise models in three dimensions, but instead exports models in compact '.spv' file format, for viewing in SPIERSview (see SPIERSview manual). SPIERSview also provides facilities to export models as geometries to other software.

Output Objects: SPIERSedit exports a series of *Output Objects*, which are listed in the *Objects* tab of the *Output* panel; these are defined by the user as combinations of masks and (for multi-segment datasets) segments. The object will consist of all pixels assigned to any of these segments that fall into any of these masks. For instance, an object might list two masks and a 'void' segment; it would then comprise any void pixels with either of the two masks. In most cases however output objects consist of a single mask and segment, and represent all pixels in that segment and that mask. By default a single object is created, representing all pixels in the default mask and the default segment.

Output Downsampling: In addition to the dataset downsampling described in the Basic Concepts section, SPIERSedit supports a second form of downsampling performed at output time. This *output downsampling* applies only to the model exported to SPIERSview, and provides performance gains in SPIERSview as it reduced the complexity of the model to be displayed. Like dataset downsampling, output downsampling is defined as an XY and a Z setting. While XY downsampling works in the same way for both dataset and output downsampling, Z downsampling does not. A Z-downsampled dataset simply skips files – for

example if Z downsampling is set to 3, only every third file in the sequence is used. Z-downsampling at output instead merges files; if set to 3, each set of 3 files is merged at output to determine whether any individual pixel is 'on' or 'off'. Output downsampling is additive to dataset downsampling – if the dataset is downsampled by a factor of 2 and the output is also a downsampled by a factor of 2, it will in fact be downsampled by a factor of four from the source data.

Output Cache: Outputting models can be time consuming, as SPIERSedit needs to process data for all slices in the dataset. However once an output has been performed, SPIERSedit stores (caches) output information for each slice; for subsequent output operations only slices that have been altered are processed, and hence output is normally far quicker. Certain operations (e.g. creating or deleting a new output object) reset this cache.

Settings Tab

The *Settings* Tab of the *Output* panel (see Fig. 15, left) was introduced under Basic Concepts.

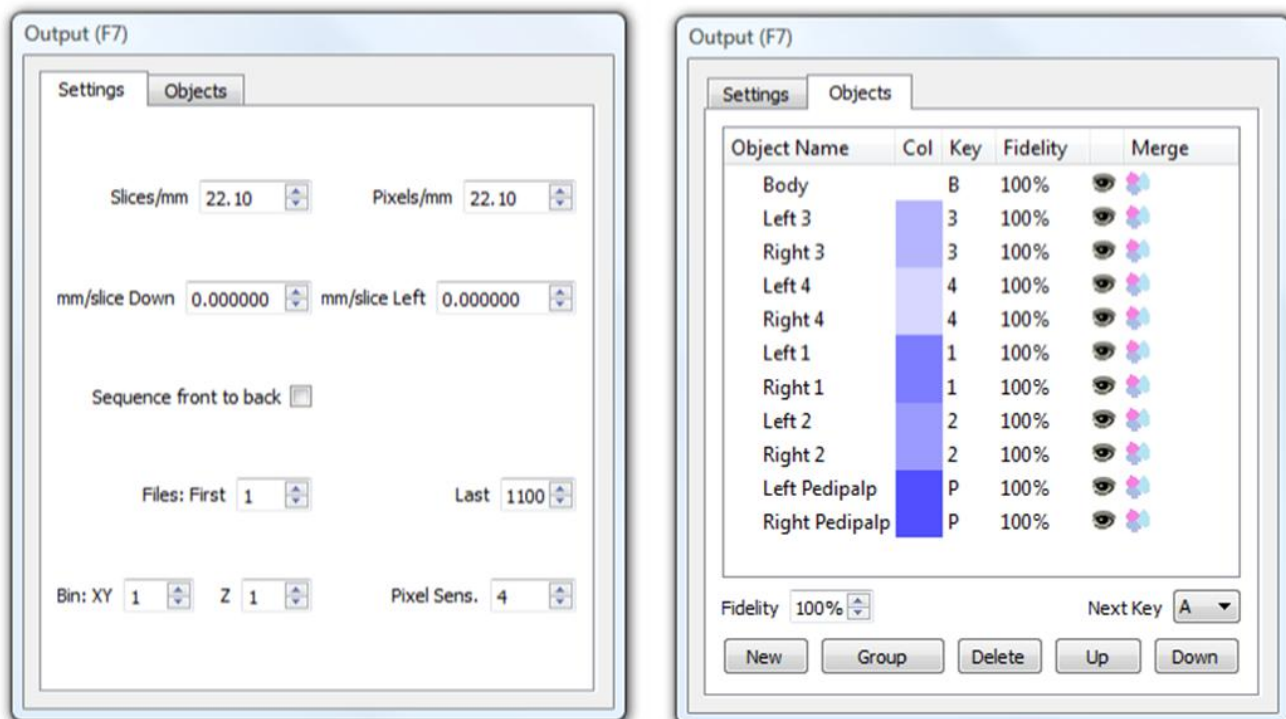


Figure 15. The Output Panel. Left; Settings Tab. Right; Object Tab.

Slices/mm, *Pixels/mm*, *Sequence front to back*: See Basic Concepts (above).

mm/slice Down and *mm/slice Left*: See Deskewing (below).

Files: First and *Last*: restricts output to part of the dataset in terms of position in the Z direction; by default all files are included, but often a restricted portion only need be rendered at any one time, and including fewer files will normally speed up the output and rendering process.

Bin: XY and *Z*: These are the settings for output downsampling within each image (XY) and between images (Z).

Pixel Sens.: This is pixel sensitivity; it controls how output downsampling combines pixels. With output downsampling on (XY or Z values > 1), each voxel (3D pixel) output is assembled from Z x XY x XY

threshold image pixels, each of which may be either 'on' or 'off'. Pixel sensitivity is the number of threshold image pixels that need to be on before the voxel is turned on. For example if XY and Z are each 3, a cube of $3 \times 3 \times 3 = 27$ pixels will be reduced to a single voxel. If Pixel Sens. is set to 1, only one of these 27 pixels need be on to turn the voxel on. If Pixel Sens. is set to 14, over half would need to be turned on. If in doubt, leave Pixel Sens. at 1; using higher values can however help to reduce noise caused by sporadic scattered pixels.

Objects Tab

The *Objects* tab of the *Output* panel (see Fig. 15, Right) provides a list of output objects that works in a manner similar to the *Masks* panel.

Object Name: This column provides a reference name for the object, which will appear in the *Objects* Panel of SPIERSview. Edit by double-clicking. Hovering the mouse over the name of an output object gives a list of the *Masks* and *Segments* that comprise it.

Col: The item colour in SPIERSview; edit by double-clicking. Note that groups can have a colour, though this is only used if they are merged (see below).

Key: The keyboard shortcut key that will be used to show/hide the object in SPIERSview ([-] means no key assigned). The *Next Key* dropdown below the main list is the key that will be assigned to the next object created.

Fidelity: The SPIERSview model fidelity; values of less than 100% will trigger a simplification step in SPIERSview to attempt to reduce the object's complexity. Values less than 50% are not advised at export (see SPIERSview manual). The *Fidelity* spinbox below the main list is the default for new objects.

Visibility: The 'eye' icon column is used to turn items on and off for export purposes; invisible items will not be exported.

Merge: Groups (see below) can be merged into single output objects, which are treated by SPIERSview as single fused objects, rendered with the group colour. This facility allows complex output object to be created. To merge or unmerge a group, just double click the merge item of the group or of one of its component objects (in the latter case you will be asked to confirm that you want the entire group merged).

Creating a new object: To create an object for a single-segment datasets, just select the masks that are to comprise it in the *Masks* panel then click *New* in the *Objects* tab of the *Output* panel (or use the *New Output Object* command on the *Output* menu). For multi-segment datasets the segments that are to comprise the object must also be selected in the *Segment* panel before the object is created. The masks and segments comprising an object cannot be edited after it is created – if need changing the object must be deleted and recreated. Objects are created with name based on the masks they use. The keyboard shortcut is set from the *Next Key* drop-down (which is also incremented after each object is created), and the fidelity is set from the *Fidelity* spin-box. Colour is based on the component masks.

Deleting objects: Select an object or objects in the Object list, then click the *Delete* button or use the *Delete Output Object* command on the *Output* menu. These commands are also used to delete empty groups.

Selecting objects: One or more objects can be selected by left clicking on any column of the *Objects* panel. To select multiple objects use Ctrl-click or Shift-click. Selection is indicated by an underlined object name. Selection of masks is used for bulk deleting and for grouping operations.

Re-ordering objects list: Objects can be moved up and down the list by selecting an object and using the *Up* and *Down* buttons. This reordering affects how the masks appear both in this list and in SPIERSview. Objects within groups can be moved up or down with a group.

Grouping: SPIERSedit (and SPIERSview) support groups of objects; the utility of grouping is discussed in the SPIERSview manual, but SPIERSedit also uses groups as a precursor to the creation of merged objects (see above). To create a group, select objects then click the *Group* button (or use the *Group* command in the *Output* menu). To remove objects from a group, select them and use the *Remove from Group* command in the *Output* menu. Groups can be created within other groups. Note that the grouping facilities in SPIERSedit are a little less sophisticated than those in SPIERSview; normal practice is to export objects ungrouped (merged objects excepted) and perform grouping in SPIERSview.

Output Commands

Three variants of the output command exist in the output menu.

Export SPIERSview: Prompts the user for a filename (with an '.spv' extension), and exports to this file.

Export SPIERSview and Launch: As above, but after Export SPIERSview is launched to view the file.

View in SPIERSview: The file is exported to a standard name and location (as 'temp.spv' in the working images folder); after export SPIERSview is launched to view the file. This is the simplest export option, and the normal one to use, as once the file has been rendered by SPIERSview the latter program can then save a copy elsewhere if this is required.

Old export code: An older version of the Export system is available; this can be activated (for all three of the above commands) by ticking the *Use Old Exporting Code* option in the *Output* menu. With this enabled output will be substantially slower (and the cache system described in Concepts will not be used), but if the user is experiencing crashes with output (which typically relate to restricted available memory) they may find that the old code succeeds where the newer version fails.

Deskewing

SPIERSedit incorporates a system to correct for skew in models caused by 'drift' of fiduciary markings in serial grinding datasets. These corrections are best explained by example – see Figure 16 (overleaf). Figure 16A shows a block in the image that contains a fossil on its left corner; it has edges cut as fiduciary markers, but these are not at exactly 90 degrees to the plane in which the specimen will be serially ground. This means that the edges will 'drift' steadily when the specimen is ground and photographed, and if these edges are used for alignment, the fossil will consequently be skewed when reconstructed.

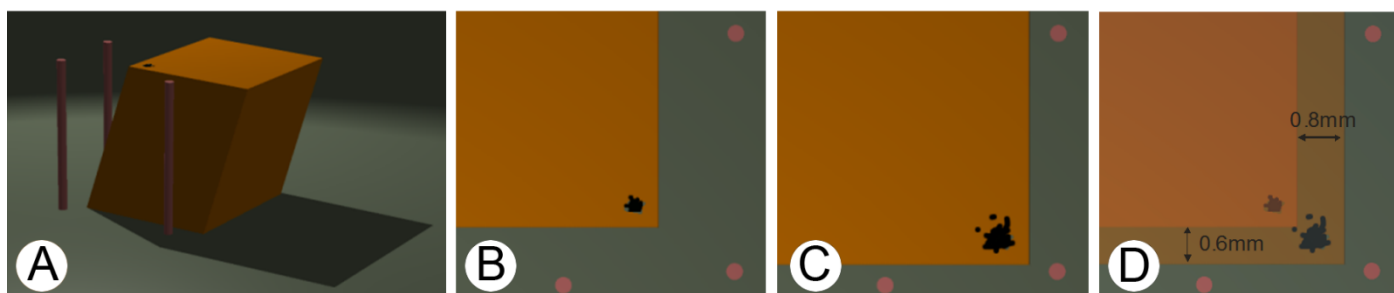


Figure 16. Deskewing example. A, block prior to grinding with fiduciary edges not at 90 degrees to direction of grinding. B. Serial grinding image #10. C. Serial grinding image #70. D. Movement between B and C.

The vertical cylinders represent fixed points of reference. They will not appear in normal photographs of the fossil as they are too far away (normal images will be zoomed in on this corner for maximum resolution). However, suppose that the images B and C were captured at slices 10 and 70 of the grinding run (at some lower magnification). By overlaying these images (and knowing the scale of the image) we can measure how much the edges have moved over these 60 slices – in this case 0.8mm to the right and 0.6mm

downwards. This example is a little artificial, but these sorts of errors do occur in serial grinding datasets, and do require correction.

Deskew values are entered into the *mm/Slice Down* and *mm/Slice Left* boxes on the *Settings* tab of the *Output* panel. In this example 0.01333 (0.8mm / 60 slices) should be entered for mm/Slice down, and -0.01 (0.6mm / 60 slice) for mm/Slice left - note the use of a negative value for left drift as the edge is actually moving right not left.

Measuring volumes

Volumes of objects can be measured using the 'Measure volumes' option in the output menu, which provides voxel counts for each object in the Output Panel rather than performing a normal export.

3D Brush

The SPIERSedit brush normally affects only single slices of the dataset. It can however be used in 3D mode, where it should be thought of as a 3D sphere extending for a number of slices equal to the Brush Size setting on the toolbar. There is no cubic 3D brush implementation. 3D brush mode is toggled on using the *3D Brush* button on the toolbar, or the *3D Brush* command on the *Brush* menu.

The 3D brush works in the same way as the normal brush, and can be used in any mode. The only difference is that it affects multiple slices, and that the Undo system will not work on 3D brush operations.

The 3D brush provides another quick means of rapidly masking large areas over many slices, deleting objects in 3D, and has many other applications.

Curves

Concepts

SPIERSedit supports a form of 'spline' object (technically Catmull-Rom splines) which it calls 'curves'; these are curved lines defined with a series of points ('nodes') through which they pass. Curves can be closed into loops, and these loops can be filled. They can be used to form objects in reconstructions by assigning them to a segment, or kept invisible in final models and used instead to create masks. Curves can be copied between slices in the same way as masks, and an interpolation system allows them to gradually change over a series of slices.

Curves rendered directly as objects (i.e. assigned to a segment) can be used to bring out impersistent or thin linear structures, or as closed and filled loops to define regions that cannot easily be edited out.

Curves used as a basis for masks, particularly when used with interpolation, provide an invaluable tool for rapidly masking datasets. They also greatly aid the consistent definition of arbitrary boundaries, as these can be defined as straight lines or gentle curves, making the nature obvious, and neater than drawing such edges in by hand.

Curves are visible in all modes if assigned to a segment, but unless SPIERSedit is in *Curve* mode they will just appear like normal pixels of whatever segment they are assigned to. When SPIERSedit is in curve mode, all curves present on a slice will be visible (in their specified colour) if assigned to a segment; curves not assigned to a segment are only visible if selected in the *Curves* panel. Points controlling a curve are only visible if the curve is selected.

Each curve can only appear once on any one slice. If more than one curve is needed on one slice then multiple curves must be created. The same curve can be used to model different objects as long as they never don't appear on the same slice, but it is better practice to create a new curve for each new object. If a large number of curves are used it can become difficult to keep track of them; the *Grey-out curves not on current slices* command on the *Curves* menu is provided as an aid to identifying curves active in the part of the dataset under inspection.

Unlike all other data editing on a per-slice basis, curves are not stored in separate files but within the .spe file. If many curves are being used it is hence recommended that the user increases the autosave frequency (see Advanced Prefs) or manually saves more regularly.

There is no maximum number of curves.

Curves Panel

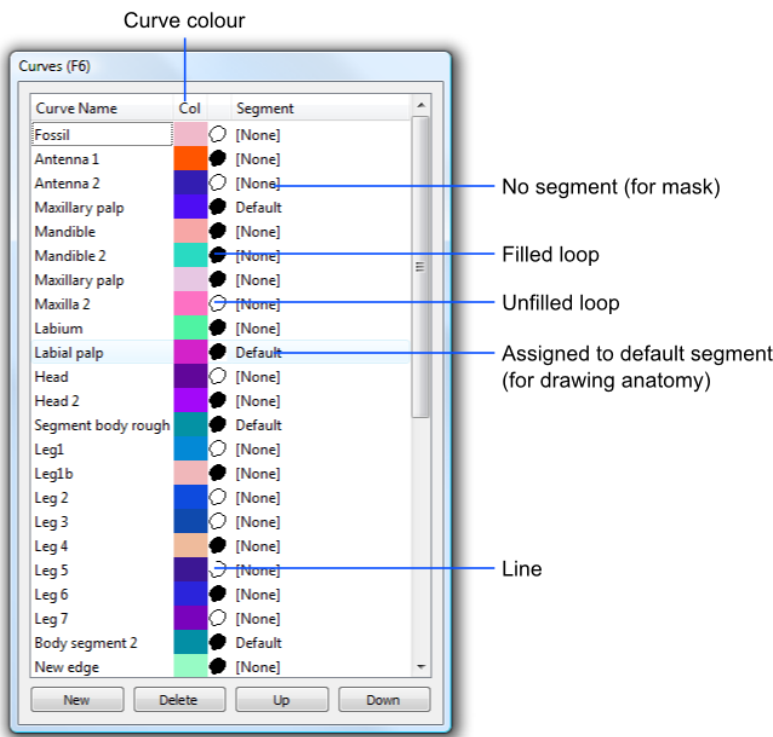


Figure 17. The Curves Panel.

Curves are listed in the *Curves* panel (Fig. 17), which works in a very similar way to the *Masks* panel.

Creating curves: Curves are created using the *New* button on the *Curves* panel. New curves are given a unique name and colour; by default they are open and not assigned to a segment. A newly created curve contains no points; before anything is visible the user needs to create nodes on slices for the curve.

Curve Name: This column provides a reference name for the curve. Edit by double-clicking.

Col: The curve colour (only displayed in curve mode). Edit by double-clicking.

'Loop' icon: This specifies the mode of the curve; curves can be open (a line), closed (an unfilled loop), or filled (a filled loop) (see Fig 18). Double-click to cycle between these three modes.

Segment: The segment the curve is assigned to, or [None] if not assigned. Double click to edit.

Slices: [Added v2.14, not shown in Fig 17] – gives the first and last slice on which the curve is used, or 'Not Used' if not currently active on any slice.

Selecting curves: One or more curves can be selected by left clicking on any column of the *Curves* panel. To select multiple curves use Ctrl-click or Shift-click. Selection is indicated by an underlined curve name. Selection of curves is used for bulk deleting, copy operations, and the mask to curve command. A single selected curve is also a requirement for editing of curves within a slice (see below); nodes will not be visible unless the curve is selected.

Re-ordering curves list: Curves can be moved up and down the list by selecting a curve and using the *Up* and *Down* buttons. This reordering only affects how the curves appear in this list.

Deleting curves. Select curves in the panel then use the *Delete* button to remove them.

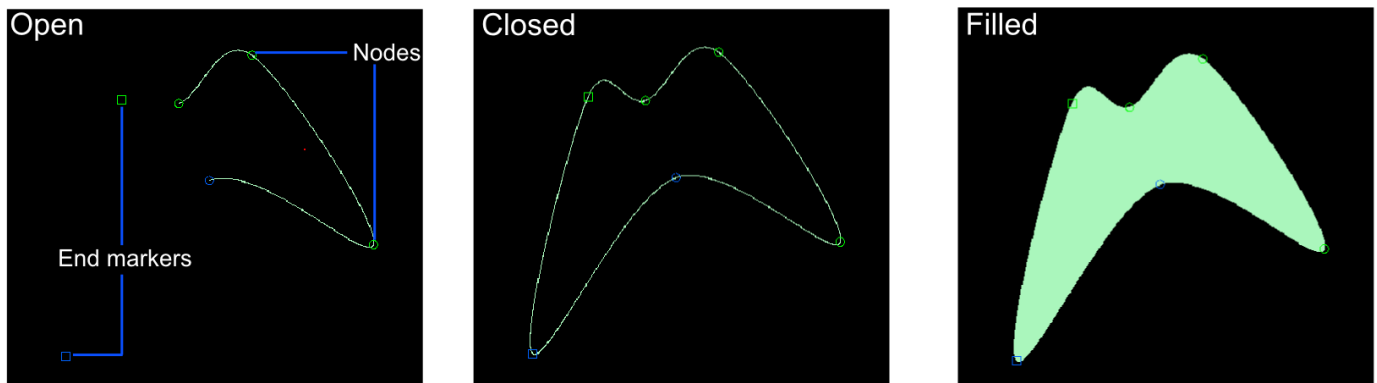


Figure 18. Curve modes. Note that in Open mode curves have two square end markers, which are specialised nodes through which the curves does not pass, but whose position affects the curvature of the final section. The blue square affects the end with the blue circle, the green square the end with the green circle. When the curve is closed these are treated as normal nodes.

Editing Curves

Curves are edited on a per-slice basis by adding, removing or moving nodes. To perform this sort of editing, a single curve must be selected in the *Curves* panel, and SPIERSedit must be in curve mode. Curve editing uses the brush for position, but brush size is ignored.

Creating nodes: The '=' key adds a node to the curve at the current mouse cursor position. If no nodes exist for the curve on the current slice, the minimum of 4 are created near the cursor position.

Removing nodes: The '-' key removes a node from the current mouse cursor position. If this would reduce the number of nodes on the slice below four, the user is asked if they want to entirely remove the curve from the slice or not.

Moving nodes. Nodes can be moved by dragging with the mouse.

Removing curves from slices: curves can be removed from slices by individually removing all their nodes. The *Remove Curves from Selected Slices* command on the *Curves* menu provides a quicker way of doing this in bulk, removing all nodes from selected curves for all slices selected in the *Slice Selector* panel.

Curve markers: By default nodes are indicated with small circles, but the *Curve Markers as Crosses* setting on the *Curves* menu changes these to small crosses, which some users may find clearer.

Locked curves: If the *Lock curve shape* item on the *Curves* menu is ticked, dragging any node will move all nodes of the curve rather than an individual node.

Resizing curves: The *Resize curves on selected slices* command on the *Curves* menu lets the user resize all selected curves on all selected slices by a specified percentage.

Copying curves

Curves can be copied between slices using the copy commands in the *Curves* menu. This works in a way completely analogous to copying masks; the reader is referred to the masks section for more details.

Interpolating Curves

Curves can be interpolated between slices; if, for example, slice 10 and slice 20 have nodes defined for a particular curve, interpolating between these slices will gradually alter the curve in between the two to form a smooth shape change from one to the other. Figure 19 (overleaf) shows a curve interpolated over seven slices.

To interpolate a curve between slices, ensure that the curve is selected in the *Curves* panel, that all slices involved are selected, and that the curve has the same number of nodes on the first and last slice selected. The *Interpolate Over Selected Slices* command on the *Curves* menu performs the interpolation.

There is no need for nodes to already exist on the intervening slices, but if they do they will be over-written by the interpolation operation.

Defining filled curves at regular intervals (e.g. every 50 slices) and then interpolating them is a fast way to rather precisely specify regions to be masked, and is a technique the authors make extensive use of.

Creating Masks from Curves

As intimated in previous sections, an important use of curves, especially interpolated curves, is to draw masks. To create a mask from a curve or curves, select the curve (or curves) in the *Curves* Panel, select the slices involved in the *Slice Selector* panel (it doesn't matter if slices are selected on which the curve does not exist, so often it's fine to use the *Select All* button here), and select the mask which the curve is drawn into in the *Masks* panel. To trigger the operation use the *Mask from curve* command in the *Masks* menu. All pixels from the curves on all selected slices will then be drawn into the selected mask.

While curves in any mode can be converted into masks, in almost all cases the curve should be in filled mode to specify an area rather than just an outline. Typically it is easier to set up the mask in unfilled mode, and then put it into filled mode before performing the *Mask from curve* operation.

Creating a mask from a curve is a one-time copy operation, which does not create any sort of permanent link. If the curve is later modified, the area masked by a *Mask from curve* operation based on it will not update, and the masked area and the curve will no longer be in synch.

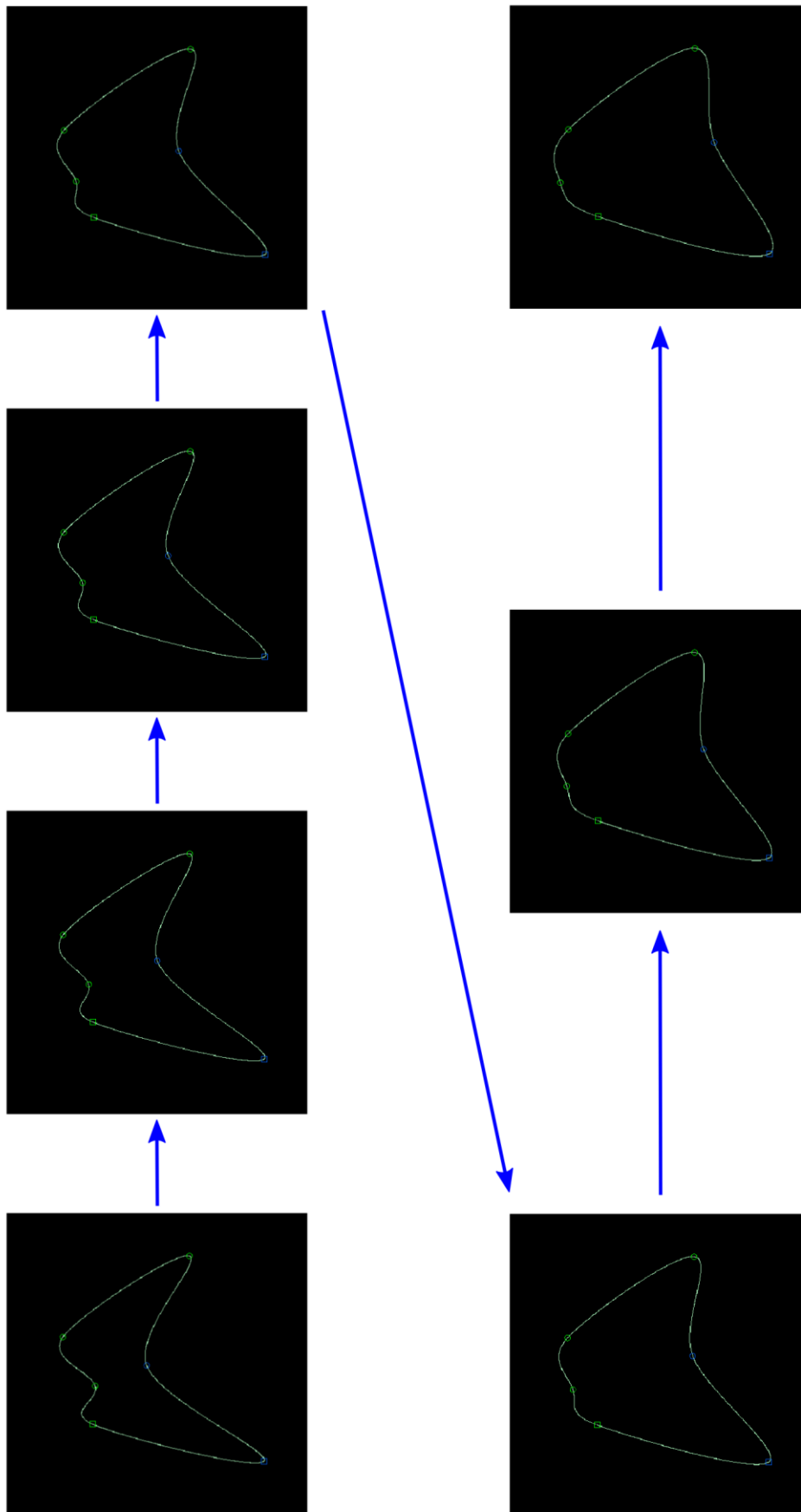


Figure 19. Curve interpolated over seven slices.

Lock/Selection Mode

This is by far the least used of SPIERSedit's six brush modes. It is used to brush a blue 'selection' colour onto images; this is similar to a mask, but stored and handled separately. Each image in the dataset has a separate lock/selection mode 'mask'. The blue colour is applied with the brush and the left mouse button, and removed with the right mouse button. It is only visible in Lock/Selection mode.

Selecting pixels in this way has two applications. Firstly, it allows selection of pixels for sample generation for *Polynomial Generation* (see above). Secondly, selected pixels are locked for generation operations (the *Generate* button on the *Generation* panel, or the use of the *Auto* tick box); it is thus possible to control regions to be affected by generation using this mode; while the *Hidden Masks Locked for Generation* option on the *Masks* menu can provide the same functionality through the masks system, in some cases this is not practical without destroying a complex existing mask structure.

Slice Spacing

For most datasets slice-spacing is to be consistent, and setting the *Slices/mm* value in the *Settings* tab of the *Output* panel is sufficient. In some datasets however slice-spacing is variable, either as a result of errors in data collection, or a change in collection parameters at some point in the sequence. While correct slice spacings are not always known in these circumstances, educated guesses at them can usually be made that will improve model results.

The reconstruction approach used by SPIERS theoretically assumes consistent slice spacing, but is flexible enough to be able to modify models to be tweaked in SPIERSview after construction to allow for some inconsistency in slice spacing. To facilitate this SPIERSedit provides a mechanism for modifying slice spacing. This should not be used as a replacement for a correct *Slices/mm* setting however; to ensure the best reconstruction this value should be set to something close to the average spacing for all slices before exact slice positions are subsequently tweaked.

To alter slice spacing, the *Show position in slice selector* item in the *Slice Spacing* menu should be selected. With this option turned on, the filenames in the *Slice Selector* panel are replaced with positions (in mm) of the slice, relative to a 0 position inferred for notional slice 0 (so if *Slices/mm* is set to 10, slice 1 will be at 0.1mm, slice 2 at 0.2mm etc).

Individual slice positions can be set by selecting the slice and using the *Set slice position* command on the *Slice Spacing* menu. SPIERSedit will not allow a user to set the position in front of the next slice or behind the preceding one; if this is attempted the program will warn the user and move the positions of other slices to maintain slice order.

If several slices share a consistent spacing, this can be set using the *Change slice spacing* command on the *Slice Spacing* menu. Select slices to be affected then use the command – slice spacing can be set as an absolute value or a percentage of standard slice spacing (the *Slices/mm* setting).

We stress again that the mechanism to deal with inconsistent slice spacing in SPIERS is something of a bodge, and while it *will* improve the reconstruction of inconsistently spaced data, the process of reconstruction (using SPIERS or any other software) is eased by the production of consistently spaced data in the first place.

Histogram

The *Histogram* panel shows the histogram of the working image for the selected segment of the current image (see Fig. 20, top). This can be restricted to a smaller area by selecting a region with the lock/selection mode, and then ticking the *Histogram shows selected* on the *Mode* menu.

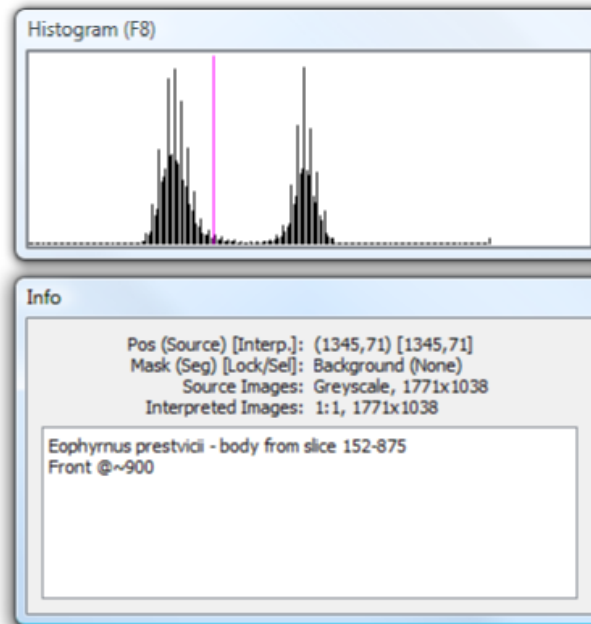


Figure 20. Histogram (top) and Info (bottom) panels

Info

The *Info* panel (see Fig. 20, bottom) shows information such as the position of the brush, which mask and segment it is currently over, and data regarding the resolution of source and working images, together with XY dataset downsampling ratio. It also provides a notes box, which is very useful for recording different generation settings for parts of the data, things to do, and anything else worth remembering regarding the dataset.

Dataset Downsampling

Concept

Dataset downsampling is explained in the Basic Concepts section (above).

Changing dataset downsampling

The current dataset downsampling levels can be changed using the *Change downsampling* command on the *File* menu. This opens a dialog that prompts for a new Z (sparsity) factor, and a new XY factor. Either or both can be changed. If the Z factor is decreased, previously unused slices will be brought into play. The user has the option of whether or not to interpolate masks (and locks/selects) and curves onto these slices; if the appropriate box is selected, masks, locks and curves are created for new slices by copying from one of the existing adjacent slices. If not, these are left blank (or any pre-existing data used, e.g. from before a downsample had taken place). If in doubt, it is best to interpolate.

Changing downsampling on the fly like this is allowed for, but is not to be taken lightly – it is a major operation, which takes some time to complete, and will result in either the discarding of data (downsampling) or potential differences in editing of areas (upsampling with later editing of certain regions).

Dataset downsampling for rough masking

As mentioned in Basic Concepts, an entire dataset can be masked while downsampled to some low resolution, and then the dataset upsampled. This will result in blocky-edged masks, but so long as these edges are in space not on the specimen this will not matter. After doing this all slices will need to be re-generated at the new (higher) resolution, so all threshold corrections done with brightening or segment drawing will be lost. If a user intends to upsample at any point prior to outputting the finished model, they should put off any such editing until AFTER upsampling.

Advanced Prefs

The *Advanced Prefs* option on the *File* menu give access to a dialog with some of SPIERS' more esoteric settings. These are global (rather than per-dataset) settings; some only take effect when datasets are opened (e.g. maximum memory for undo and cache).

Maximum memory for undo: Controls the maximum size of the undo stack. As the stack is wiped every time a multi-slice operation is executed, this is normally limited to the changes to a single slice anyway, and thus the default of 512 Mb is ample. For computers with limited RAM this can safely be reduced, although of course this means that fewer Undo's will be possible.

Undo Timer Interval: Determines how many seconds' worth of actions are clumped into a single undo event. If the amount of memory for the undo stack is reduced, it may be worth increasing this value to allow bigger undo steps and hence to allow stepping further back in time.

Maximum memory for cache: SPIERSedit keeps copies of all files it loads (source images, working images etc) in a 'cache'. Use of this cache greatly speeds subsequent access to files. The maximum size for the cache is set here; it is recommended that it should be set as high as the system reasonably allows; approximately one half of total system RAM might be a good guideline. Be aware that this may make the opening of multiple copies of SPIERSedit problematic.

Tune Cache for Rendering: If ticked, the cache preferentially stores working images, mask data etc rather than source images. If the latter are relatively large (e.g. colour images at twice resolution of working images) this can give substantial performance increases while rendering, as it may enable the entire dataset to fit within the cache. The downside is that it will reduce performance for 'leafing through' the dataset.

File compression: Controls how working images, mask data and lock/selection data are written to disk. Setting to 'off' produces large files, but normally results in faster disk operations. If set to 'low' or 'high' these files are compressed when written back to disk (i.e. the compression only happens if/when they change). 'low' compression offers substantial space savings over 'none' (typically a factor of 5-10), 'high' may increase compression factor a little over 'low'. High settings produce extra processor load (compression/decompressing data), so will normally slow many operations down. The default setting of 'low' will be best in most cases.

(Re)compress source files now: Allows the user to recompress the source images; this is the only exception to SPIERSedit's rule that it never changes source images, so should be used with caution.

Behaviour is controlled by the file compression slider - if this is 'off' then this button converts all source images to '.bmp' format. If slider is set to 'low' images are converted to '.png' format (normally around one half of the size of '.bmp' files. If the slider is set to 'high' images are converted to '.jpg' format, which will be substantially smaller again. '.jpg' images are saved at maximum fidelity, so artefacts should be negligible; repeated conversion back and forward between high and low/off is not recommended though, as this could build up artefacts with each conversion. Compression of source images to 'jpg' format is not normally recommended, but may be an acceptable means of saving disk space once the user is confident that all generation operations have been completed.

(Re)compress working files now: Applies the current file compression level (see above) to all working files.

Cache compression: SPIERSedit can compress files when storing them in its internal cache (see above). This trades off processing speed against memory usage, but it may enable the entire dataset to fit in the cache, which can in itself increase performance substantially. If cache compression is set to low around 4-5 times more files will fit in the cache – if set to high this figure will reach perhaps 7-8 times. Note that if you set cache compression to 'high' then colour images are cached using the lossy JPEG algorithm - there may be slight artefacts which can in theory (though probably not in practice) make their way into your datasets via slice-generation. Cache Compression should normally be turned off, but may be worth trying on machines with limited memory. Note that changing this setting empties the cache, so will not provide an instant performance increase!

Autosave frequency: Controls how often the SPIERSedit settings file (.spe) is written out.

Clear Recent Files List: This button removes everything from the *Open Recent* list on the *File* menu.

Update System

SPIERSedit incorporates an auto-updating system (as do the other two SPIERS applications). If connected to the Internet, it will check for updates every time it is run. If it finds an update it will give the user the option of downloading and installing it. Agreeing to this will close SPIERSedit once the updater has been downloaded, and the proceed to install the update.