IonView 0.91a

Introduction

IonView is a Mac OS X application for viewing, measuring, annotating and exporting data from scanning ion conductance microscope (SICM) images generated by an IonScope system such as the ICnano. It also includes a Quick Look plugin for previewing these files in the Finder and other QL-compatible applications. IonView does not support writing to the SICM image format and probably never will. It does not modify the original files.

These notes relate to IonView 0.91a. The most recent version can be found online at `http://walkytalky.net/ionview/`.

The software was written specifically for use in connection with my own PhD research and its functionality is limited. If IonView does not do what you need, you should also try the excellent open source SPM application Gwyddion, which provides a much more comprehensive set of analyses. IonScope SICM files are supported through an importer module, which I contributed for the same selfish reasons I wrote this.

Bug reports and feature requests for both IonView and the Gwyddion import module should be sent to `mattcaldwell@me.com`. I make absolutely no promises with regard to acting on them.

System Requirements

IonView requires at least Mac OS X 10.6 (Snow Leopard) and also works on 10.7 (Lion). If you encounter problems running under either OS version, please send me a bug report, including as much system info as you can muster.

The application uses some fairly clunky OpenGL for 3D rendering and is not at all optimised, so it needs a reasonable graphics card. Even so, given the graphical demands of OS X, pretty much any machine that will run that should be fine with IonView. But if for some bizarre reason you have managed to kludge Lion onto a creaky ancient Pentium box with 8-bit VGA, please don't come crying to me.

SICM Image Files

IonScope's `ScanIC` control software saves its images in a custom format, using the `.img` file extension. A full specification of the format is beyond the scope of this document, but an overview of its main features is included below.

Pretty much everyone and her dog has used the `.img` suffix for some file format or other over the years, so it is basically useless as a type identifier. Apple themselves claim the extension for NDIF disk images, an obsolete format used for making mountable pseudo-floppies back in the days of Mac OS 8. It is quite unlikely you will run across one of these files.

IonView defines a new file type, IonScope SICM Image, with type identifier `com.ionscope.img`. This type is associated with the file extension `.sicm`. If you change the suffix on your SICM `.img` files to `.sicm`, then everything should work sensibly, with files opening in IonView by default and automatic Quick Look previewing.

IonView also pretends it can open `com.apple.disk-image-ndif`. It *can't*, obviously, but this means the system will let it open files with the `.img` extension. If you don't want to rename your SICM files, you can instead open them by dragging to the IonView icon or by using **Cmd-O** or **File** → **Open...** from within the application. Or you can set IonView as the default application for opening NDIF files like this:

- Select a SICM `.img` file in Finder.
- Press **Cmd-I** or choose **File** → **Get Info**.
- Go to the **Open with:** panel in the info window.
- Select **IonView.app** from the pop-up menu.
- Press the **Change All...** button.

So that you can have something to play with if you don't have ready access to a SICM rig, a small number of SICM images have been included alongside the app, in the `sample data` folder. The files have the `.sicm` extension so they should work without the above faffing.

Data Type and Interpolation

Depending on how they were acquired, the data in SICM files may represent various different kinds of measurement, including topography, current, slope and other arbitrary signals, such as fluorescence intensity, digitised via the system's ADC inputs. The actual recorded value in all cases is a voltage. For many signal types, the relationship of that voltage to the physical quantity it represents is not encoded within the file itself because it depends on the configuration of external hardware.

IonView is primarily concerned with topographic images, for which the necessary information is usually present in the file header. Other kinds of image file will probably not work very well. Viewing such files in 2D should give usable results, but physical measurements will probably be wrong and as a result the 3D rendering may be a mess.

In some scanning modes, notably hopping probe (Novak et al. 2009), data is not always directly sampled for all points in the file. The missing data may be filled in by bilinear interpolation at acquisition time, or the file may contain uninterpolated data. For topographic images, the filename usually has an appended `t` for the former and `nt` for the latter.

IonView implements bicubic interpolation for such images, with overshoot avoidance based on Heckbert (1985) and some fairly simplistic removal of single-point noise spikes. Where the image has already been linearly interpolated, the program attempts to reconstruct the original data points. Doing so requires information that is not always present in the file header, and also

some guesswork. The results may or may not be an improvement on the interpolated image.

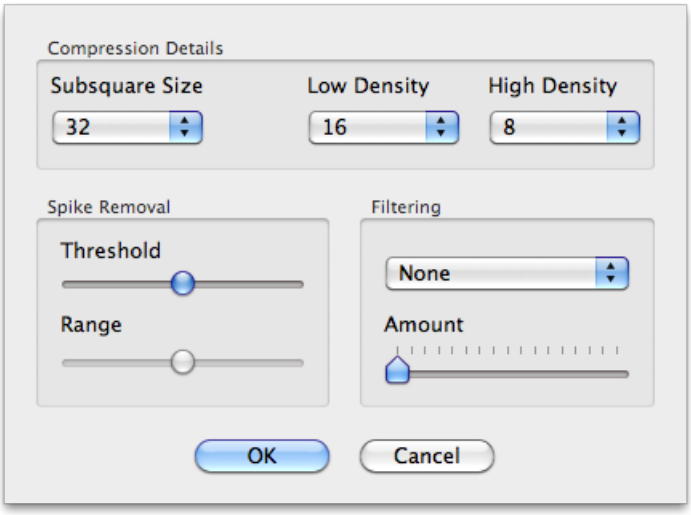


Figure 1: Interpolation Settings

In some cases it can be useful to override the header data and explicitly specify the settings to use for interpolation. This can be done via the configuration sheet accessed with **Cmd-I** or **Model** → **Interpolation Settings...** (Figure 1). This also provides for filtering of the interpolated data, which may occasionally be useful for noise reduction. Only the **Box** and **Median** filters are currently implemented, and the latter is very slow.

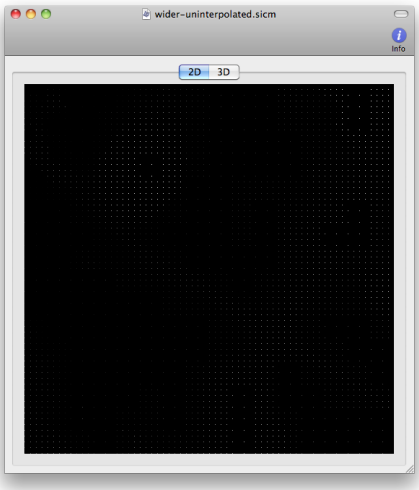
Document Window

When you open a SICM image, it is presented in its own document window. The main panel shows the image itself, with tabs for 2D and 3D renderings. By default the 2D tab is visible (Figure 2).

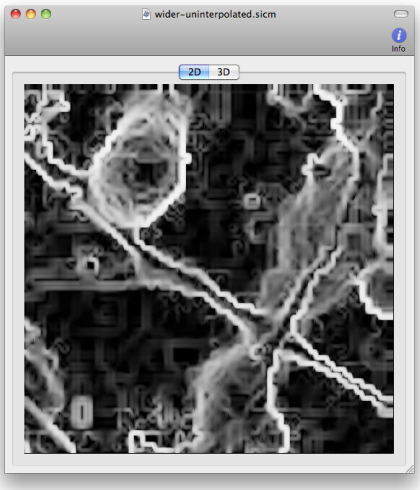
Also visible by default is the toolbar, which provides a single button to open the Info Panel at the side of the window.

Document windows can be resized and support copying, but not pasting. The window is never marked as 'dirty', so it can be closed without warning. Annotations, interpolation, filtering and so on are all ephemeral: nothing is saved when the image is closed.

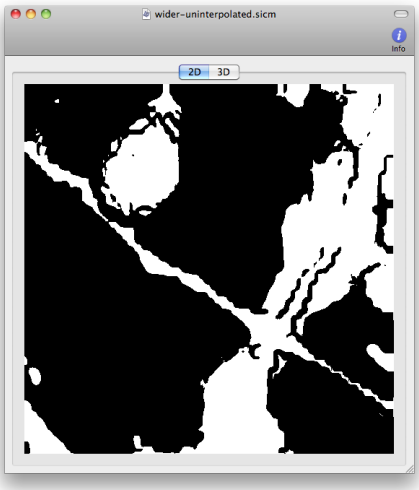
(a) Interpolated Data



(b) Uninterpolated Data



(c) Gradient



(d) Threshold

Figure 2: 2D Image View

You can have as many images open at a time as you like. Most main menu actions such as **File** → **Export**, **Edit** → **Copy** and **Model** → **Interpolation Settings...** apply to the frontmost window. A few features described below operate across multiple windows.

2D Viewing

The 2D view represents the SICM data as a normalised greyscale image, where black is the lowest value in the data and white is the highest. By default the image is shown just as contained in the file, rather than using IonView's interpolation. If the file is a `t`-suffixed hopping topography image, this will include linear interpolation of the image points (Figure 2(a)). If it is an `nt`-suffixed uninterpolated image, only the actual sample points will be drawn, with unsampled points shown in black (Figure 2(b)). You can switch between this default version of the image and the internal bicubic interpolated version using the controls described below.

Clicking and dragging in the image draws a selection rectangle. Holding down the **Shift** key while doing this constrains the selection to be square. Selection capabilities are currently fairly anæmic. If present, the selection is used for some of the information shown in the Info Panel and as an option when exporting data to text. It can also be used to specify a boundary (see below).

View options are accessed through a floating control panel (Figure 3), which is displayed when a 2D view is frontmost.

The radio buttons at the top select between different ways of displaying the image data. **Original** displays the data as contained in the file, which may include bilinear interpolation generated when the image was captured. **Interpolated** displays the IonView's bicubic interpolated version, as described above. **Gradient** displays an image showing the estimated surface slope at each position (Figure 2(c)). **Threshold** displays a threshold region that identifies the parts of the image within a specified topographic range (Figure 2(d)). For non-topographic images, the Gradient and Threshold views are unlikely to make much sense.

Topographic threshold bounds are set with the four sliders below. The upper pair define bottom and top clipping planes. The lower pair specify a range

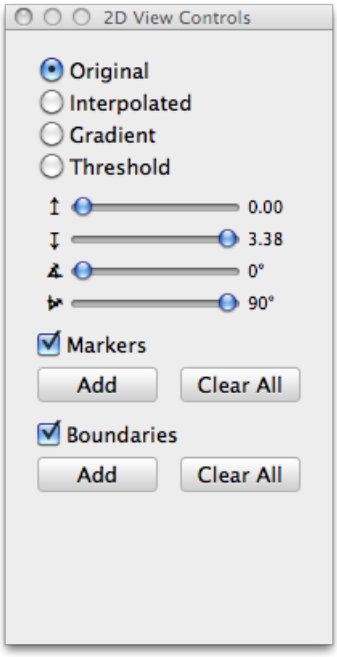


Figure 3: 2D Control Panel

of gradients to be included. In the resulting image, pixels for which both the Z measurement and slope are within bounds are shown in white; all other pixels in black.

The 2D view currently supports two kinds of graphical annotation: markers, which are associated with a single point in the image, and boundaries, which define a rectangular region. Both kinds are managed through the control panel.

Markers are added by double-clicking in the image, or by clicking the **Markers** → **Add** button on the panel. Either action brings up the sheet shown in Figure 4. The marker position is specified in image pixel coordinates. If the sheet was invoked by double-clicking, the click location is set as the initial marker position. (SICM image coordinates have their origin at the bottom left.)

Markers are given a label for internal identification, but at present the labels are not displayed anywhere.

Several different kinds of markers are available, which differ in their graph-

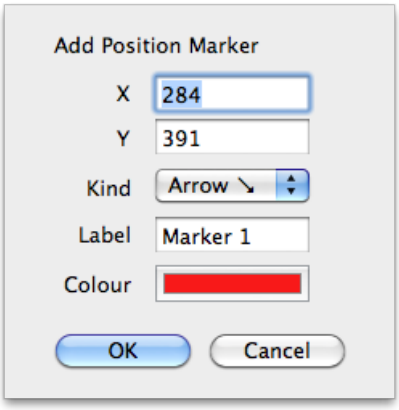


Figure 4: Add Marker Sheet

ical representation: arrows in various directions, crosshairs, etc. At present, marker types other than **Pipette** are drawn in the 2D view only. You can select a different colour for each marker by clicking the colour well at the bottom of the sheet.

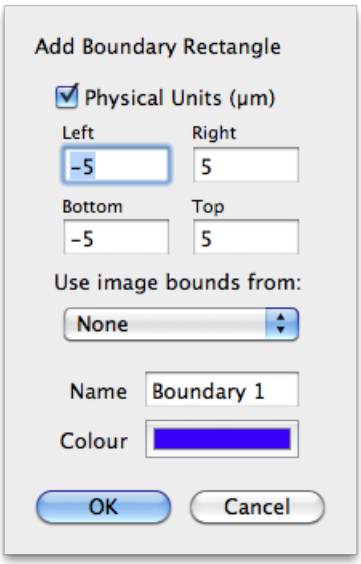


Figure 5: Add Boundary Sheet

Boundaries are added using the **Boundaries** → **Add** button, which brings up the sheet shown in Figure 5. Boundaries can be specified either in image

pixels, where the origin is again at bottom left, or in physical µm with their origin at the centre of the XY piezo stage of the SICM. If a selection rectangle is present on the image, the initial boundary coordinates are set to match that selection. Otherwise, they are set to the outer edge of the image.

Physical units are somewhat unintuitive, but pixel positions are only meaningful within a single image. When dealing with multiple related images physical units are necessary. It is often useful to identify which regions of different images overlap. A common case is when one image is a higher-resolution zoom on a portion of another. The **Use image bounds from:** pop-up menu allows the image bounds from one image to be specified as a boundary on another. The menu lists all open images apart from the one to which the boundary will be added. Simply select the desired image and the boundary coordinates will be set (Figure 6).

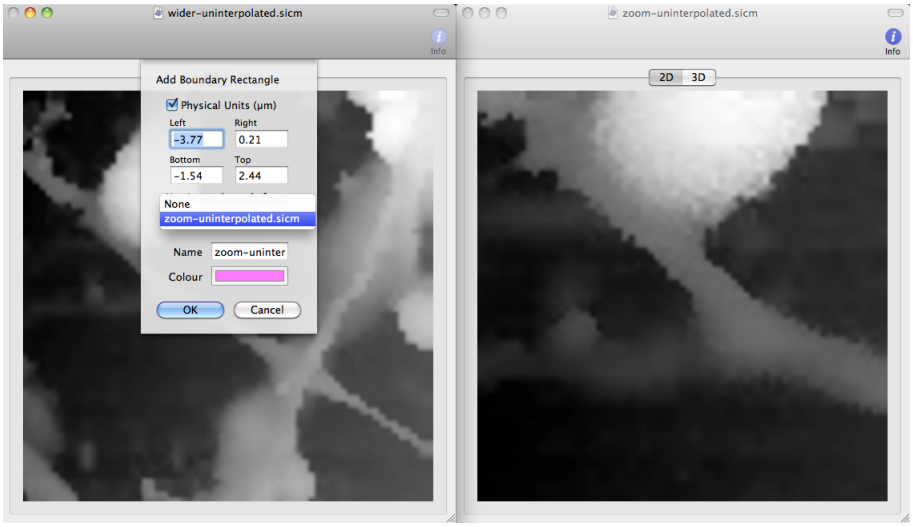
Like markers, boundaries are given labels but these are not displayed.

Markers and boundaries are visible by default, but either set of annotations can be temporarily hidden using the panel checkboxes. You can also remove all annotations of either type using the associated **Clear All** button.

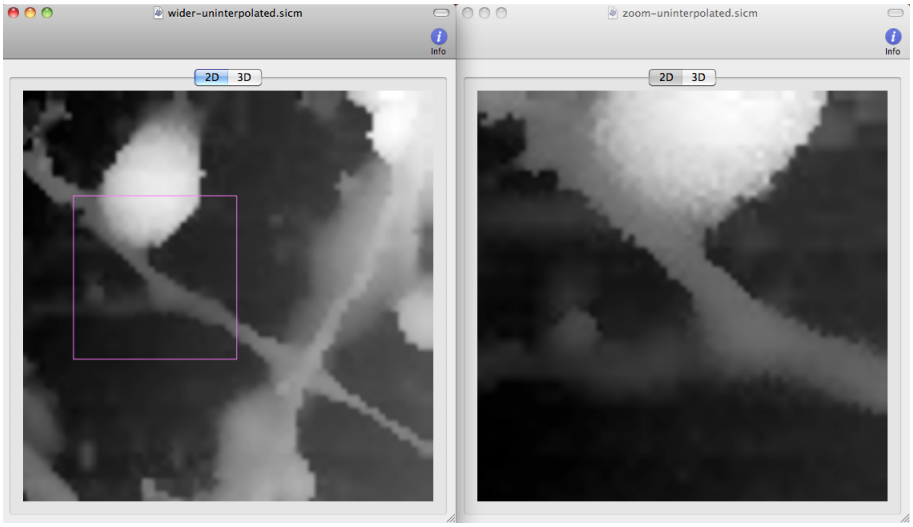
3D Viewing

The 3D view shows a rendering of the image data as a topographic model (Figure 7). For non-topographic SICM images the results are likely to be quite ugly. At present the 3D view always uses the internal bicubic interpolated data. There is no option to use the uninterpolated version. As with the 2D tab, there is a control panel for manipulating various aspects of the rendering (Figure 8). There are also some more direct interactions using the mouse and keyboard.

The orientation of the model with respect to the view position is adjusted by dragging with the mouse. Drag left and right to rotate the object horizontally, up and down to rotate it vertically. Rotation occurs around the centre of the view rather than the centre of the object. These coincide initially, but the object can be moved within the view by dragging with the **Cmd** key pressed down. Once again, dragging up and down moves the object vertically, while dragging left and right moves it horizontally. The interaction of all these movements can easily add up to something quite confusing; use



(a) Select another open image to use as boundary



(b) Boundary marks region of zoom

Figure 6: Using boundaries to show relationship between images.

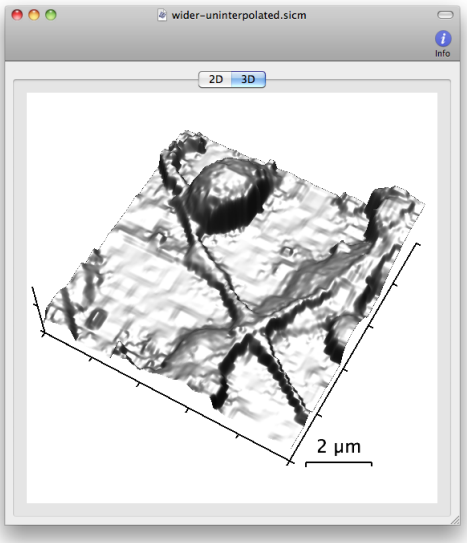


Figure 7: 3D View

the **Reset Position** button in the control panel to restore the default viewing parameters.

You can resize the object within the view in two different ways, changing either the camera position (tracking) or its effective focal length (zooming). These functions are attached to the scroll ball on your mouse or the scroll gesture of your trackpad. (If you're using some other input device, these features may not be available.) Scrolling left and right adjusts the zoom, while scrolling up and down tracks in and out. Used in combination these can be used to exaggerate or reduce the perspective distortion. These parameters are also restored to their defaults by the **Reset Position** button.

If you have multiple image windows open, you can synchronise them all to have the same 3D viewing parameters using the **Sync 3D Views** button (also available via the **View** → **Sync 3D Views** menu item or the keyboard shortcut **Cmd-Return**). The settings from the frontmost window are applied to all other open windows. Doing this only makes sense if the different scans are actually of the same target area.

By default, the 3D model is displayed at its full interpolated resolution. Very occasionally it can be useful to switch to a lower level of detail (LOD), omit-

ting some of the intermediate vertices. Pressing the left arrow key reduces the LOD; pressing the right arrow increases it again.

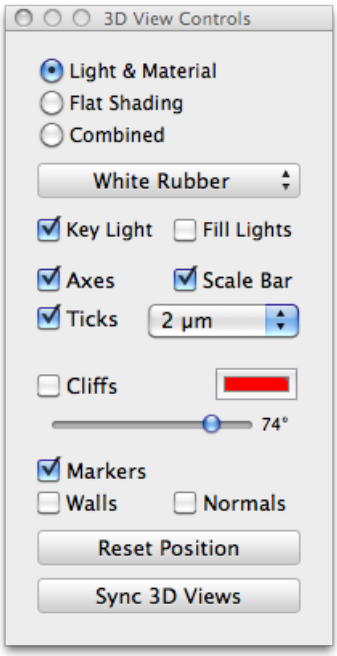


Figure 8: 3D Control Panel

The control panel (Figure 8) provides access to various more discrete view controls.

The radio buttons are the top control the rendering method. The default rendering is **Light & Material**, which uses OpenGL's smooth (Gouraud) shading model to determine the colour at each pixel based on the material properties of the surface and the lighting environment. The **Flat Shading** option instead ignores the material properties and lighting and just uses the same colour for every pixel in a given surface polygon. In this case, the colour is the same as the normalised greyscale used in the 2D view. Flat shading tends to be a bit murky, while smooth shading can be excessively shiny and metallic. The **Combined** mode generates both renderings and blends them together, which can sometimes produce a good compromise. However, the blending is fairly unsophisticated and can lead to strange artefacts, especially if the scan is large or the camera is tracked out a long way. In addition, the blending can introduce unwanted alpha components into

exported images.

The pop-up menu below the rendering buttons allows you to choose from a standard set of OpenGL surface materials. These differ in colour and shininess, but all of them are fairly shiny. This choice has no effect when flat shading is used.

The next pair of checkboxes allows you to switch on or off two sets of lights. The **Key Light** is bright and positioned high overhead. The **Fill Lights** are low and point in from the sides. The latter can be useful for filling in detail on steep shadowed faces, but they also tend to blow out the detail. They are very dark on their own and it's unlikely you'll often want to use them without the key. As with the choice of material, lighting settings have no effect when using flat shading.

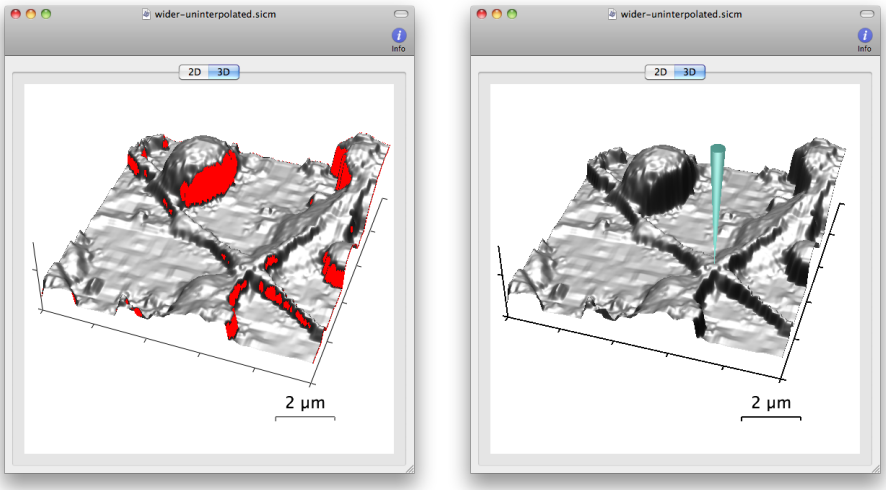
The remaining controls turn on and off various image features. Most of these should be fairly self-explanatory, but a few points should be noted.

The **Scale Bar** is always sized according to the surrounding image frame. It is labelled to represent whatever tick distance is shown on the axes. The default tick spacing is chosen so that the ticks and scale bar are of *vaguely* comparable size, but that's as far as it goes. No attempt is made to make the scale bar the same size as the axis ticks (that turns out to be quite tricky) and if you change the tick spacing things may look very out of proportion. In some circumstances you may need to adjust the bar length later in an image editing program to make it look right.

The **Cliffs** option highlights regions where the surface slope exceeds some specified angle (Figure 9(a)). Because of the nature of SICM imaging, this can reveal regions of probable collision between the probe tip and the sample. The cliff angle is specified using the slider, and the highlight colour can be changed using the adjacent colour well.

The only markers that are currently drawn in 3D are those whose kind is **Pipette**, which are rendered as shown in Figure 9(b). These can be switched off with the **Markers** item, but other kinds cannot be switched on.

The **Normals** option draws the calculated normals for all the polygons in the model. This is almost certainly useless unless you're actually debugging this stuff.



(a) Cliff Highlighting (b) Pipette Marker in 3D

Figure 9: 3D View Features

Info Panel

The **Info** icon in the toolbar shows and hides a panel at the side of the document window. This panel contains information about the scan and, if present, the selection. The various fields are illustrated in Figure 10.

The box at the top left of the panel depicts the area of the scan (in grey) within the overall lateral imaging range of the SICM system (in white). This gives a quick thumbnail sense of the size of the scan and its position relative to other scans. Note, however, that if you open scans made on different SICM rigs, the white boxes may represent different sizes.

Lateral measurements are given in both image pixels and physical units. Note that image pixels may not be the same as screen pixels if the window has been resized. Axial measurements are only shown in µm because there is no meaningful pixel equivalent.

Information relating to the mouse position is shown only for the 2D view, and only when the mouse is actually over the image. The mouse Z value is the scan surface height at the mouse location, measured within the piezo frame of reference.

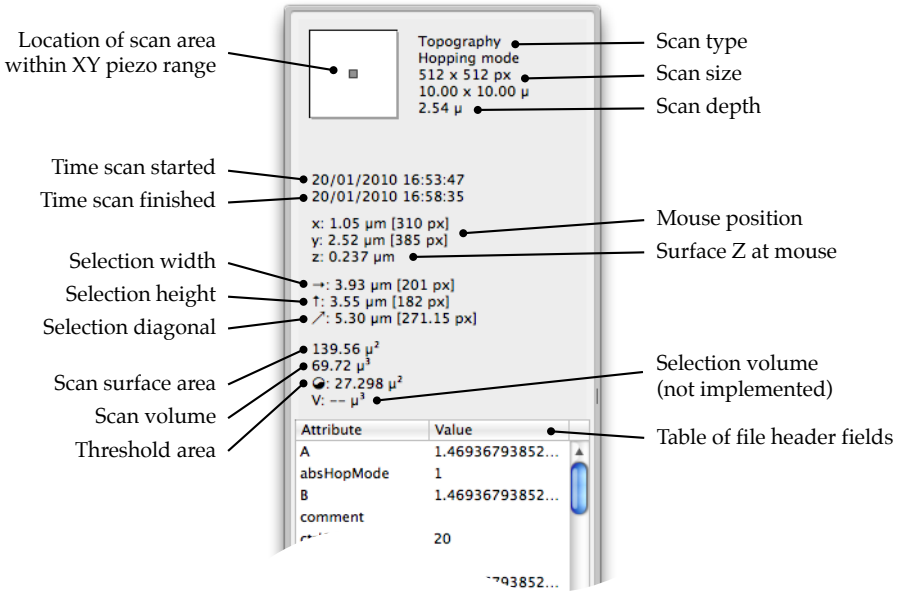


Figure 10: Info Panel

Similarly, selection information is only given when a selection is present. The selection diagonal is provided mainly for measuring lateral distances in the scan.

The scan surface area is calculated by adding up the areas of all the polygons in the model. The scan volume is calculated by adding up the volumes of vertical prisms capped by those polygons and bounded at the bottom by a horizontal plane at the lowest point in the scan. The threshold area is the area of all white pixels in the threshold view, if relevant. This is a *cross-sectional* area, not a surface area. Equivalent calculations for the selected region are not yet implemented but may be added in future.

The lower part of the panel shows a scrollable table of *all* the fields of the file header. Many of these are unused or redundant. The most important ones are presented above in more accessible form, but it can occasionally be useful to look at some of the others.

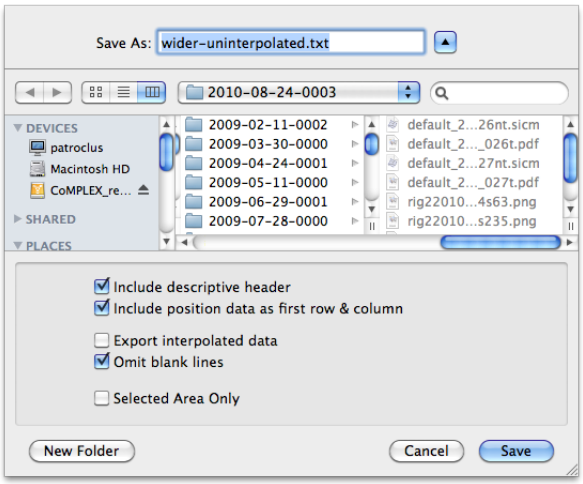


Figure 11: Text Export Options

Data Export

One of the main purposes of IonView is to allow scans to be exported into other, more useful, forms. This is done via the **File** → **Export** submenu.

File → **Export** → **Image...** saves the frontmost image to any of the standard file formats supported by the OS X ImageKit framework. The image is exported more or less as it appears onscreen, including any visible annotations and, if present, the selection rectangle. It is not currently possibly to export only the selected region. For 3D images, there may be some differences in the colour representation depending on the image format. Most formats supporting alpha transparency will show the 3D background as transparent. If alpha blending is used in the rendering, as notably occurs in the Combined mode, then this will also be exported, which can give rise to undesirable artefacts in some image formats.

File → **Export** → **Text...** saves the frontmost image as tab-delimited text, suitable for import into other analysis software such as R, Matlab or Excel. The main data content is just a table of Z position measurements in µm. There are several export options, as illustrated in Figure 11.

The optional descriptive header provides human-readable details of the scan and what has been exported. This may confuse other programs trying to import the data. The end of the header is marked by a line containing only

the following text:

```
-- DATA --
```

The **Include position data as first row & column** option, as the name suggests, adds an extra header row and column containing, respectively, the X and Y positions at which the corresponding columns and rows of Z measurements were taken.

By default, the original raw data is exported. To instead export IonView's bicubic interpolation, select the **Export interpolated data** option.

For uninterpolated or unfinished images, the **Omit blank lines** option skips export of rows and columns that contain nothing but unsampled pixels. Note that there may still be lines that contain a mixture of sampled and unsampled data, and these will not be omitted. Unsampled points are represented in the output by the value `NA`.

Finally, the **Selected Area Only** option exports just the subset of scan pixels that fall inside the selection rectangle.

File → **Export** → **Movie...** exports the 2D versions of all open images as frames of a QuickTime movie. This is basically a very quick hack and provides no configuration options at all. Images are used as currently visible onscreen, including any interpolation, selection and annotations. The image windows must actually have their 2D tabs selected or the resulting movie probably won't work. Frames are placed in the movie in order of filename. This will produce the correct sequence if the images have been named with a timestamp as the SICM control software does by default. If not, you may have to do some renaming to get stuff in the order you want. The movie is generated with a fixed frame rate of 6 fps.

IMG File Format

SICM data generated by the ICnano system is saved in IonScope IMG files, a custom format based loosely on an earlier AFM/STM image format called ECS. The format retains a number of legacy features that are not relevant to SICM images, but is also distinct enough to be incompatible with existing ECS reading code.

IMG files consist of an 830-byte descriptive header, followed by a table of 16-bit unsigned integer samples. The samples represent the voltage applied to the piezo, or the ion current or other signal sampled by the ADC, scaled according to the particular hardware configuration of the SICM. Attributes from the header must be used to convert the data into physical units.

A partial list of the most useful header fields is given in Table 1. Multibyte numeric values are stored in little-endian order. Floating point numbers in the header use the Pascal Real format, a non-IEEE 48-bit floating point format with 1 sign bit, 39 bits of fraction and 8 of exponent.

Offset	Field Name	Type	Notes
0	version	Int16	Always 50
2	xdim	Int16	Number of samples across
4	ydim	Int16	Number of samples down
6	fsdHVA	Real48	Voltage range of the piezo amplifier (V)
12	fsdDAC	Real48	Voltage range of the DAC (V)
18	fsdADC	Real48	Voltage range of the ADC (V)
30	piezoCalX	Real48	X piezo sensitivity
36	piezoCalY	Real48	Y piezo sensitivity
42	piezoCalZ	Real48	Z piezo sensitivity
56	scanSize	Real48	Width of the scan ($\times$10 nm)
666	modeStr	Char[41]	String describing the scan type

Table 1: Important fields in the IMG file header

The nature of the samples may be determined from the first letter of the modeStr field. If it is 'T', the image contains topographic data, if it is 'C' the samples are of the ion current, and if 'A' then it is some other signal sampled from an auxiliary ADC input.

For topographic images, the sample data can be converted to µm using the following formula:

$$\text{microns} = \frac{\text{sample} \times \text{piezoCalZ} \times \text{fsdDAC} \times \text{fsdHVA}}{32767}$$

For current or auxiliary signals there is insufficient data in the file to map the samples to their true physical units, but the values can be converted to the voltage received at the ADC like this:

$$\text{voltage} = \frac{\text{sample} \times \text{fsdADC}}{32767}$$

The full lateral piezo range can be determined from the header information as follows for X:

$$\text{micronsX} = 2 \times \text{piezoCalX} \times \text{fsdDAC} \times \text{fsdHVA}$$

and equivalently for Y with appropriate substitutions.

Samples are stored in the file in bottom-up order: the first xdim samples are the bottom row of the image, the next xdim are the next up from the bottom, and so on.

An unconverted sample value of 0 indicates that the pixel was not sampled, typically because the scan was halted before completion. In images that have been recorded with compressed sampling, an unconverted value of 0x7fff (32767) indicates placeholder pixels that have not been filled in by the default bilinear interpolation.

Mumbo Jumbo

If you use IonView or its supporting materials for a scientific paper, please cite accordingly. An example BibTeX record might be:

```
@manual{IonView,
  title={IonView 0.91a},
  author={Matthew Caldwell},
  year=2012,
  url={http://walkytalky.net/ionview/}}
```

References

Paul Heckbert. Non-overshooting Hermite cubic splines for keyframe interpolation. Technical report, Three Dimensional Animation Systems Group, Computer Graphics Laboratory, New York Institute of Technology, February 1985.

Pavel Novak, Chao Li, Andrew I Shevchuk, Ruben Stepanyan, Matthew Caldwell, Simon Hughes, Trevor G Smart, Julia Gorelik, Victor P Ostanin, Max J Lab, Guy W J Moss, Gregory I Frolenkov, David Klenerman, and Yuri E Korchev. Nanoscale live-cell imaging using hopping probe ion conductance microscopy. *Nature Methods*, 6(4):279–281, April 2009.