Gaussian Fitting Routines for Hinode/EIS data

Ver. 2, 11-Feb-2009, Peter Young

We consider here two situations when you want to fit Gaussians to your EIS spectra:

1. Automatically fit a Gaussian to a line profile at each pixel in an image
2. Average a number of pixels in the image corresponding to a specific structure, and then fit Gaussians to this averaged spectrum.

The present document concerns itself with situation 2. For situation 1 the Solarsoft routine eis_auto_fit can be used.

Situation 2 is complicated by the fact that there are a number of spatial offsets in the EIS instrument, which mean that a pixel mask chosen from a Fe XII 195.12 image (for example) does not exactly correspond to the spatial structures in other wavelengths.

A discussion of the EIS spatial offsets is given on the EIS Wiki at:

http://msslxr.mssl.ucl.ac.uk:8080/eiswiki/Wiki.jsp?page=DataProAnalysis

Described below are a set of IDL routines that account for the spatial offsets to create an averaged spectrum that correctly represents the observed spatial structure. Also described is a widget-based fitting routine (spec_gauss_eis) that allows the user to fit the lines in the spectrum.

Example data set (9-Dec-2006)

We use the following data set as an example:

eis_l0_20061209_113031.fits

which is a 256x256 raster that takes 15 spectral windows with 40s exposures.

Getting started

Firstly you need to calibrate your data with eis_prep. The recommended calling sequence is:

```
IDL> eis_prep, l0name, /save, /default, /retain
```

where l0name is the name of the level-0 FITS file. A level-1 FITS file will be created together with an associated error file.

If you already know which wavelength you want to use to identify the spatial structure that you're interested in, do:

```
IDL> eis_make_image, l1name, 185.21, im185
```

where l1name is the name of the level-1 FITS file, and 185.21 is the wavelength I'm using for this example. The 2D array im185 contains an image derived from the Fe VIII 185.21

emission line (7 wavelength pixels centered on the 185.21 wavelength have been averaged to generate the image).

Now plot the image using

IDL> plot_image,sigrange(im185)

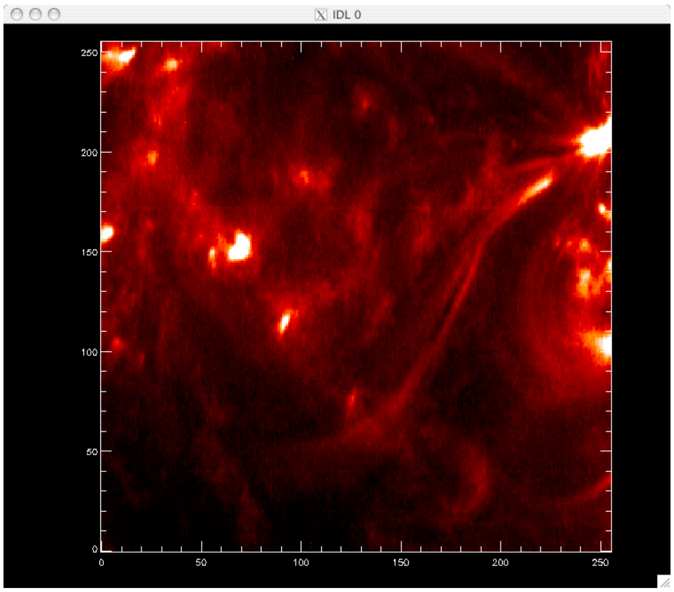

Figure 1. Fe VIII 185.21 image from the 12-Dec-2006 data set.

Since Fe VIII is a cool line (log T = 5.8) then the image consists of a number of fairly small brightenings with a few narrow loop structures.

If you want to browse the data set to choose a different emission line, try using eis_raster_browser to browse your data-set:

IDL> eis_raster_browser, l1name

Creating a pixel mask

A pixel mask is an array of same size as the image described earlier but containing only 1's and 0's. Pixels marked with a 1 indicate that they will be used to generate the averaged spectrum.

To create a pixel mask from your image, do

IDL> eis_pixel_mask, im185, mask185, 185.21, 1

where im185 is the image created in the previous section, mask185 will contain the output mask, 185.21 is the wavelength of the line you've specified, and 1 is the size of the slit (should be either 1 for the 1" slit, or 2 for the 2" slit).

Upon calling eis_pixel_mask, you will see the image plotted again (with color table 5) and in the IDL window you will see a menu:

```
IDL> eis_pixel_mask,im185,mask185,185.21,1

 *** CHOOSE A MODE ***
  left:   polygon mode
  middle: painting mode
  right:  exit
```

The mouse is used to create a pixel mask by clicking on the image. *You need to use a 3 button mouse for the routine to work!* There are two options for creating the mask:

1. Polygon mode. Click on a number of points in the image and, when complete, the routine will join the dots to create a polygon. The polygon will be filled, and the enclosed pixels will be set to 1 in the pixel mask.
2. Painting mode. By holding down the left mouse button, you can 'paint' over the image, selecting pixels as you go. Selected pixels can be removed by clicking with the middle mouse button.

Generally for small spatial features you will use the Painting mode to accurately select the pixels you need, while Polygon mode is used for choosing large spatial regions.

While clicking on the image, make sure to always check back to the IDL window to see what mode you are currently in. Note that the right mouse button is always used to exit out of a mode.

For our example we are going to choose a small brightening in the image so it is necessary to zoom in on the region of interest:

```
IDL> eis_pixel_mask,im185,mask185,185.21,1,xsc=[50,100],ysc=[120,170],
col=255
```

which displays X-pixel range 50 to 100 and Y-pixel range 120 to 170. Note that col=255 is used to set the color of the over-plotted pixels to white.

Use the Painting mode to select the pixels. In Figure 2 the selected pixels are displayed in white.

If you want to reset the mask, then simply exit out of the routine and do:

```
IDL> mask185=0
```

The Appendix to this document explains how the Polygon option in eis_pixel_mask works.

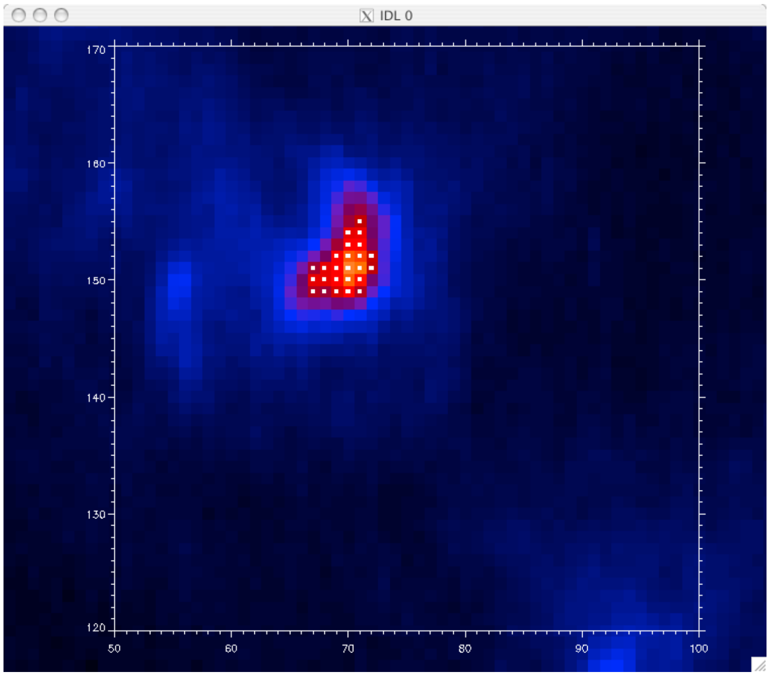

Figure 2. Fe VIII image showing the selected pixels (small white squares).

Viewing the pixel map in different wavelengths

It is useful to check how the pixel mask looks for other wavelengths. To create the mask for, e.g., Si VII 275.35, do

IDL> mask275=eis_adjust_mask(mask185,275.35)

eis_adjust_mask is an important routine as it takes into account the various spatial offsets to move the selected pixels to spatially match the original wavelength.

To give an indication of the shifts involved in the present case, do:

IDL> plot_image,mask185.image+mask275.image,xra=[50,100],yra=[120,170]

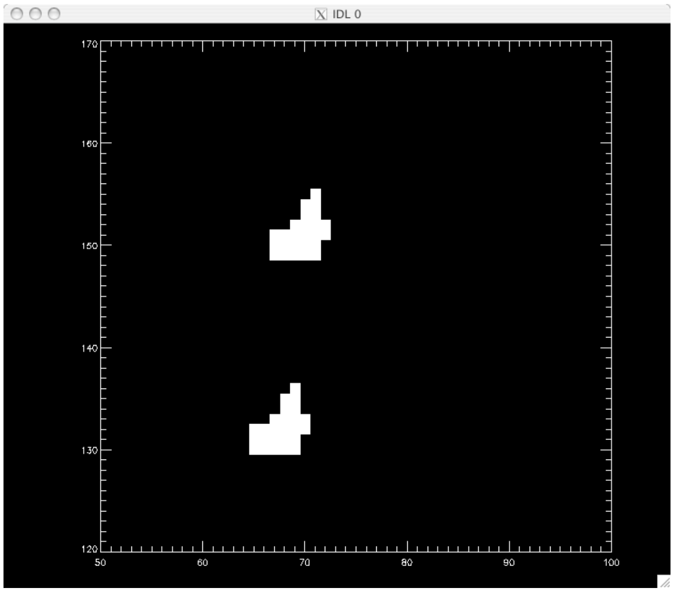

Figure 3. Pixel masks for Fe VIII 185 (upper) and Si VII 275 (lower).

The upper set of pixels are those from the original Fe VIII 185.21 mask, while the lower set of pixels are those from the Si VII 275.35 mask. The latter is seen to be shifted in both X and Y.

Now since Si VII and Fe VIII are formed at the same temperature, then we are able to check how accurately the derived Si VII mask sits on the brightening seen in the Si VII image (remember that mask275 has been derived from the mask chosen in the Fe VIII image).

We first create a 275 image:

```
IDL> eis_make_image, l1name, 275.35, im275
```

and then do:

```
IDL> eis_pixel_mask,im275,mask275,275.35,1,col=255,xsc=[50,100],ysc=[120,170]
```

which gives the image shown in Figure 4. The match is fairly good although seems to be off by about 1 pixel in the Y-direction. Since the shift of the pixel mask is done in integer pixel numbers, then this is within the uncertainty of the method taking into account also the uncertainty in the measured EIS spatial offsets.

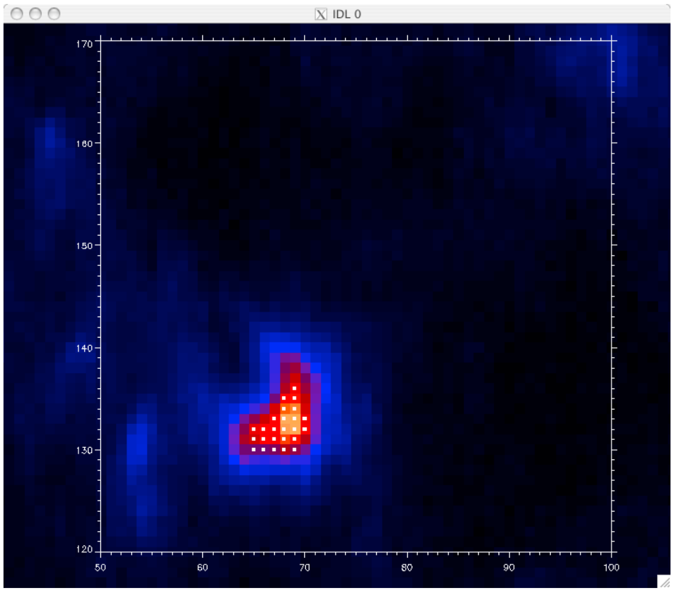

Figure 4. Si VII 275.35 image with pixel mask overplotted.

Deriving the averaged spectrum

Once you are happy with your chosen pixel mask, an EIS spectrum averaged over the spatial region can be derived by doing:

```
IDL> eis_mask_spectrum, l1name, mask185, swspec=swspec, lwspec=lwspec
```

swspec and lwspec are both structures with the following tags:

```
IDL> help,swspec,/str
** Structure <1539eaa4>, 5 tags, length=36868, data length=36866, refs=1:
   WVL          DOUBLE    Array[2048]
   INT          FLOAT     Array[2048]
   ERR          FLOAT     Array[2048]
   QUAL         INT       Array[2048]
   QUAL_MAX     INT       25
```

You will see that the spectrum is defined for the full size of the CCD (2048 pixels). You can view it by doing:

```
IDL> plot,swspec.wvl,swspec.int,psym=10,/xsty
```

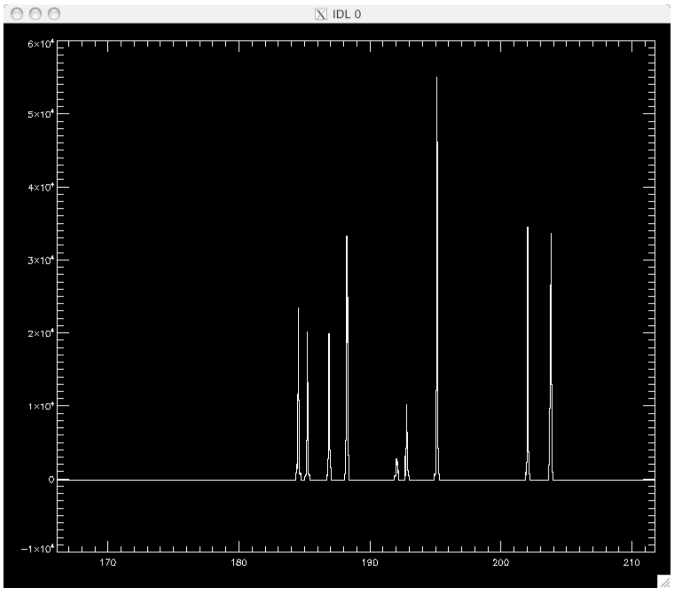

Figure 5. Averaged spectrum derived from pixel mask.

You'll see only a few emission lines that correspond to the wavelength windows of the EIS raster. All other wavelengths are set to an intensity value of -100.

Note that the number of pixels averaged over to generate the spectrum is stored in the tag QUAL_MAX. The significance of the QUAL tag is discussed in the separate tutorial on SPEC_GAUSS_WIDGET, the Gaussian fitting routine.

Fitting Gaussians

Gaussians can be fitted to the output spectrum using the routine SPEC_GAUSS_EIS which is called as:

IDL> spec_gauss_eis, swspec

it is actually a wrapper to a general purpose Gaussian fitting routine called SPEC_GAUSS_WIDGET present in the /gen branch of Solarsoft.

A separate tutorial is available for using SPEC_GAUSS_WIDGET.

Appendix: the Polygon option in eis_pixel_mask

Start up eis_pixel_mask on the whole Fe VIII 185 image:

IDL> eis_pixel_mask, im185, mask185, 185.21, 1, col=255

and click somewhere on the image with the left mouse button – this puts the routine into Polygon mode.

I will choose an approximately rectangular region in the bottom left of the image by clicking three times with the left mouse button. You'll see that after the first button press, subsequent ones will show a line joining the newly added point to the previous. Figure 6 shows the result after 3 button presses.

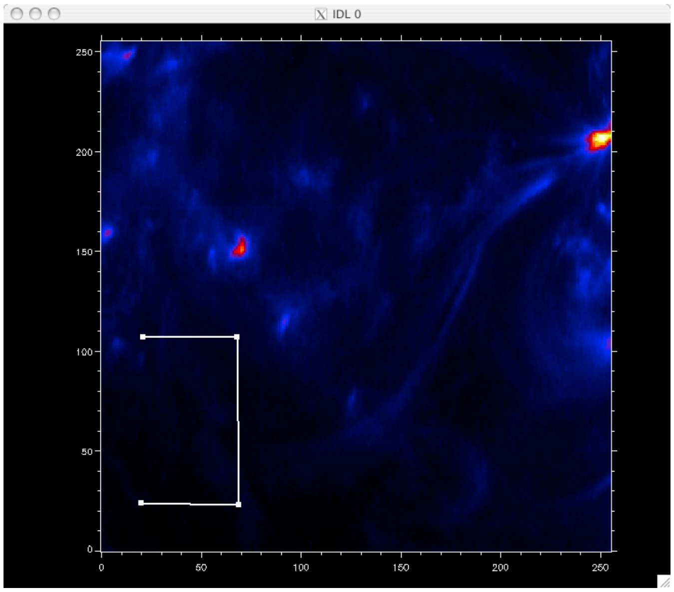

Figure 6. An image from during the process to select a region with the Polygon option in eis_pixel_mask.

To 'close' the rectangle, simply click on the RIGHT mouse button, which exits the Polygon mode and joins up the last two dots. Figure 7 shows the result where the selected pixels form a rectangular block in the image.

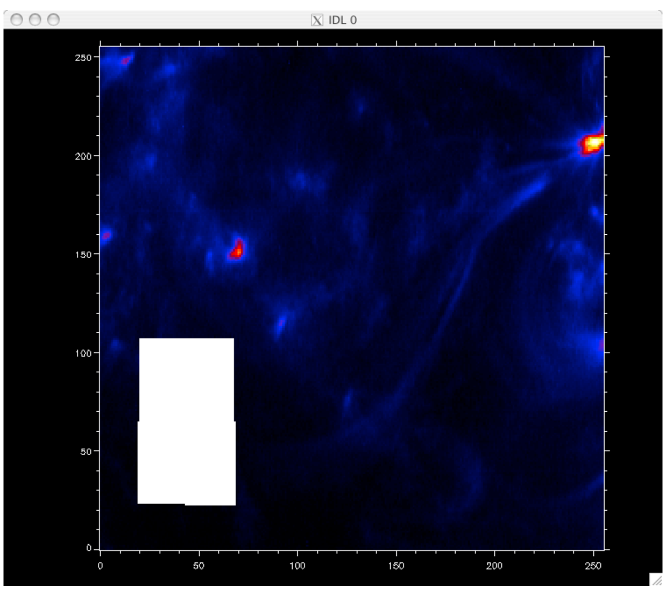

Figure 7. The selected spatial region after using the Polygon option in eis_pixel_mask is shown.

To fine-tune the selected block of pixels, the Painting mode can be used.