Handy Things to Know for Lab 2

Brian Keeney

February 8, 2005

This week we'll go over the IRAF tasks that you'll need to know to complete Lab 2. First, I'll show you how to use a script that I've written to take all of the `.FIT` files in a particular directory and give them `.fits` extensions. Then we'll learn how to blink two (or more) images, and align them if necessary. Finally, we'll learn how to dark subtract and continuum subtract your narrowband images to produce pure Hα and O III images.

FIT2fits Setup

For Lab 1, we had you individually copy all of the `.FIT` files generated at the telescope into the `.fits` files that IRAF expects. As you can imagine, this can become quite tedious if you have lots of images to reduce, so I have written a script to do the conversion for you. The script is called `FIT2fits` and is located in my home directory (`~keeney`). Below you will find instructions for telling UNIX where to find `FIT2fits`, as well as a fix for `ds9` and `ximtool` that should prevent the annoying crashes that we've had to deal with the last couple of weeks.

1. First, change into your home directory.
2. Open a text editor (e.g., `nedit`) to edit the file named `.alias`.
3. Somewhere between the header at the top of the file and the `# Generic aliases` line, add the following lines:

   ```
   alias FIT2fits '~keeney/FIT2fits'
   alias ds9      'ds9 -unix_only'
   alias ximtool  'ximtool -unix_only'
   ```

4. Save your changes and exit the text editor.
5. Type the command `source .cshrc` at the prompt.
6. Now change into the directory where your `.FIT` files are stored.
7. Run the script by typing `FIT2fits`. When the script is finished, all of your `.FIT` files will have been replaced by `.fits` copies. The original files can be found in a new subdirectory named `FIT/`.

You only have to run the setup procedure (the first five steps) once. After that, you'll always be able to run the script by typing `FIT2fits`, and `ds9` and `ximtool` will start with the proper options set.

Blinking Images

Both `ximtool` and `ds9` have capabilities for blinking images. When you use the `display` task within IRAF it asks you which buffer you want to use. If you have two (or more) images that you want to blink, then you should load them in different buffers (1 and 2 for example, although it doesn't really matter what numbers you choose; 2 and 4 would work equally well). By loading the images into two different buffers, you have allowed them to be simultaneously stored in memory.

The exact procedure for blinking the images depends on whether you're using `ds9` or `ximtool`. If you're using `ds9` then, after the images have been displayed in different buffers, go to the "Frame" menu and select "Blink Frames". It might take a few seconds for the images to blink, but you'll know when they start. Select "Single Frame" under the "Frames" menu to stop blinking. If you're using `ximtool` then, after the images have been displayed in different buffers, go to the "Options" menu and select "Blink frames". Alternatively, you could open the `ximtool` Control Panel by clicking on the leftmost button on the right side of the toolbar. A new window should open that looks like the window in Figure 1. There is a section labeled "Blink/Register" in this window. Look for the "Frames List" line in this section. Here you should set the buffers that you want to blink between (leaving any unused space blank). Then click "Blink" and your images will blink, probably after some delay. Click "Blink" again when you want the images to stop blinking.

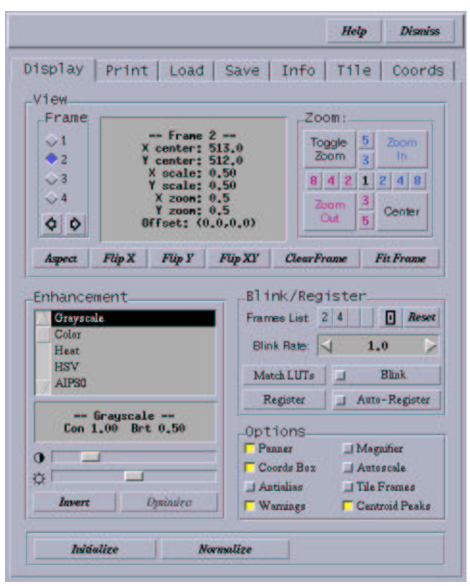

Fig. 1.— The ximtool Control Panel window.

Registering Images

What if your images don't have the same pointing so that all of the stars move when you blink them? This situation would certainly make it more difficult to pick out the asteroid in the field. This problem is solved by registering, or aligning, your asteroid images using IRAF's `imshift` task. You can use the `imshift` task to shift the coordinates of one of the asteroid images so that it matches the coordinates in the other one. The procedure for aligning two images is as follows:

1. Pick a star that is in both of your images, and preferably near the center in one of them. Measure the position of that star in each of your images (e.g., using the Radial Plot option in `imexam`).

2. Calculate the Δx and Δy pixel shifts needed to align the two images to the same coordinate system.

3. Double check your shifts by calculating them based on a second star in your image (just in case you chose the asteroid to begin with!). They should agree within about a pixel across the image.

4. Plug these shifts into `imshift`. This task expects input of the form `imshift input output xshift yshift`. So, when you call `imshift` you will type something like `imshift ast2 ast2.aln -10 10`, which tells `imshift` to start with the image `ast2.fits`, add $\Delta x = -10$ to all of the x-positions and $\Delta y = 10$ to all of the y-positions in this image, and save the output to the file `ast2.aln.fits`.

5. Display your two aligned images in different buffers and blink them to see how well you've done. The stars might still move a little bit, but if you let enough time go by before taking your second asteroid image, it should move a lot more than any of the stars.

So you say you didn't wait long enough between images and still can't tell which object is your asteroid? Don't panic yet. Take a close look at your finding chart. The asteroid will be a fairly bright object that **isn't** on it! If you still can't find it, call me over to have a look.

After you've identified which object is the asteroid in each of your images, you should be able to calculate how far it has moved relative to the background stars. From there, it's pretty straight forward to determine the proper motion. You should also make note of any peculiarities that you encounter (e.g., does the asteroid move in the way that you expected?).

Dark Subtraction

We will use the versatile IRAF task `imarith` for both dark and continuum subtraction. As its name suggests, `imarith` is a task for performing arithmetic manipulations on images. It expects input of the form `imarith operand1 operator operand2 result`, where `operand1` and `operand2` are either input images or constants and `result` is the output image. The operator can be any arithmetic operator (i.e., "+", "−", "*", or "/") or one of the relational operators "min" and "max". If you have a science frame of the Crab Nebula named `m1.fits` and a dark frame named `dark.fits`, you would dark subtract the science frame by typing `imarith m1 - dark m1.ds` from within IRAF, and the resulting image will be stored as `m1.ds.fits`.

Continuum Subtraction

What's the difference between dark subtraction and continuum subtraction? The point of dark subtraction is to subtract off the counts that come from dark current in each of your images. The point of continuum subtraction, on the other hand, is to subtract all of the continuum sources (i.e., stars and sky) out of a narrowband image to leave a pure emission line (Hα or O III) image. Since the offband filters are typically wider than the emission line filters, they will let in more light over the same amount of time. Thus, you will have to scale the continuum images to correct for this effect.

What criterion should you use to scale your images? One method is to scale them so that the average sky counts in the dark-subtracted continuum and Hα/O III images are the same. The first thing that you'll need to do is determine the average sky counts in your dark-subtracted on- and off-band images. (*Hint:* Do you remember the `imstat` task? Ask me how to use it to determine the average sky value in your images. One of the `imexam` options might also be helpful.) Then, if *contsky* is the average sky counts in the continuum image and *emsky* is the average sky counts in the emission line image, you will need to scale the continuum image by a factor of $f = emsky/contsky$ (Why?). This can easily be accomplished by using `imarith cont * f scaledcont`, which multiplies every pixel in your continuum image (`cont.fits`) by the appropriate factor (f) and stores the result in the new image `scaledcont.fits`. Now that you've set the sky level to be the same in your narrowband images, you can proceed to subtract the continuum image from the Hα/O III image to create a pure emission line image.

Now try it for some of your images. Be warned, it may be necessary to register your images so that the stars are found in the same position in both the continuum and Hα/O III images! Do you notice any "holes" in your pure Hα or O III images? What causes these?

The method of continuum subtraction described above is very similar to the method that you will use for flat fielding. You will need to take flats for Lab 3, and I'll hand out a worksheet describing the flat fielding process in the next couple of weeks. Until then, good luck with Lab 2!