Flat Fielding Guide

Brian Keeney

March 1, 2005

This week we'll go over the specifics of flat fielding. We'll be using the IRAF task `imarith` to do this, which was described in "Handy Things to Know for Lab 2". If you haven't looked at that handout yet, you should probably do so before you attempt this one.

Combining Your Flats

The lab writeup asks you to take enough flat fields to obtain $\sim 100,000$ total counts. In order to do this, you'll have to take at least 2 flat fields in each filter (and probably more like 3 if you're being careful). In order to reach the maximum signal-to-noise ratio (S/N), you'll need to combine all of your (dark-subtracted!) flat fields in each filter until you have a single image for that filter which has $\sim 100,000$ counts per pixel (Why does this give you maximum S/N?). The most straightforward way of doing this is to use `imarith`. If you have two flats named `flat1.fits` and `flat2.fits` that you want to add together, then you could easily do so by entering "`imarith flat1 + flat2 combined`" from within IRAF. In this case the image `combined.fits` will contain the sum of `flat1.fits` and `flat2.fits`.

The `imarith` task becomes less useful when you have to add more than two images together. The biggest limitation of `imarith` is that it won't let you manipulate more than two input images at a time (i.e., "`imarith flat1 + flat2 + flat3 combined`" will not work). The easiest way of combining a lot of images is to use the `imcombine` task, which expects input of the format "`imcombine input_images output_image`" and accepts many optional keywords (the one we'll be using is the `combine` keyword). In order to make `imcombine` add three flats together, you will have to type "`imcombine flat1,flat2,flat3 combined combine=sum`". The comma separated list (with no spaces!) tells `imcombine` to use all of those images as the input list, and the `combine` keyword tells `imcombine` exactly how you want the images to be combined. By setting `combine=sum`, we have told `imcombine` to add all of the input images together and make the output image their sum. Other options for the `combine` keyword are `combine=average` and `combine=median`, which work in a similar way.

Since our filenames are usually rather long, you can save yourself some typing on the command line by creating a file that will serve as the list of input images. If all of your flats in the V-band end in "`flat_V.fits`", for example, then you can easily create this file with the command "`ls *flat_V.fits > Vflat.list`", which takes the output of the command "`ls *flat_V.fits`" and places it in the file `Vflat.list` (If your flats aren't so conveniently named, you can manually create the list file using a text editor; just make sure that you have one image name per line). You can then combine all of the images by entering "`imcombine @Vflat.list combined combine=sum`"

at the IRAF prompt. The `@` symbol tells IRAF to look in a file named `Vflat.list` for the input image names.

Normalizing Your Flats

Since we'll eventually be dividing our science images by our flat field in order to get rid of pixel-to-pixel variations in sensitivity, we want to normalize our combined flat field. All this means is that we'll change the average value in the combined flat field to 1 instead of $\sim 100,000$. That way, when we divide the flat into the science images, we won't be reducing the counts in the science images by 5 orders of magnitude. The first order of business is to determine the the average counts in the center of the combined flat field (or whichever part of the combined flat corresponds to the position of the targets in your science images). **Hint:** The `imstat` task can return statistics on a section of an image if you use the format "`imstat image[`x_1`:`x_2`,`y_1`:`y_2`]`" to define a rectangular region with lower-left corner (x_1, y_1) and upper-right corner (x_2, y_2).

After you've done that, divide your combined flat field by this average value to create a normalized flat. Double-check that things have worked as you planned by displaying the normalized flat and maybe even running `imstat` on it. What's the average value in your normalized flat? What about the minimum and the maximum values? How do they compare to the same values in the combined (unnormalized) flat field? Are the results what you expected?

Flat Fielding Your Images

This is the easy part. Once you've obtained normalized flats for each filter, you just need to divide all of the (dark-subtracted!) images taken in a given filter by the appropriate normalized flat field using `imarith`. This will eliminate the pixel-to-pixel variations in sensitivity and allow you to directly compare counts between objects in different parts of your image. Now it's time to calculate magnitudes!

At this point, you've learned all of the IRAF tasks that you'll need to perform the data analysis for Lab 3. Now that you have the required tools, it's just a matter choosing the appropriate tool to get the desired result. By the end of next week's lab (March 8), groups should be distributing magnitudes amongst themselves. Good luck and, as always, if you have any questions, feel free to ask.