Experiment with CCD Camera Images

For an astronomer, an image is not necessarily a pretty picture. Indeed the information in each image usually goes far beyond what you can see at first look. Image processing is the technique by which an image can be transformed to reveal (or suppress) selected features, or to extract quantitative information in other forms.

There are three programs we use often in astronomy to acquire, analyze, and modify images:

- SAOImage ds9
- Aladin
- ImageJ or AstroImageJ

We may use these in different instances because they offer a different set of features. Aladin, for example is excellent for comparing images with one another and with astronomical data bases, and for making quick color images from several images through different filters. SAOImage ds9 is versatile for acquiring images and interacts in real time with our cameras. ImageJ, written and maintained for biomedical imaging, can be modified for special tasks, and is ideal when we are measuring the brightesses of stars.

In this activity we will use ImageJ because it provides for sensitive control of the colors in an image and preserves all of the information of the original data.

The basics

Astronomical images are most often created by placing an electronic detector such as a Charge Coupled Device camera at the focus of a telescope. CCD cameras have an array of sensors exposed to light focused by the telescope. Each sensor records the light from its share of the image. At the end of an exposure, a computer reads them and measures how much light was received point by point in the image. An element of this record is called a *pixel*. Just as in the camera of your cell phone, state of the art large telescopes use millions of pixels per CCD detector, and may have arrays of detectors side-by-side to cover large swaths of the sky. We use a camera ourselves in other activities. First, however, we will see how we can view and measure the data they return.

We will work with four images recorded in short exposures with our CDK20 south telescope at Mt. Kent, in Queensland, Australia. We have selected images of the Tarantula Nebula, a region of gas and young hot stars in the nearby Large Magellanic Cloud galaxy. There are four images in the data set, taken on the night of October 4, 2010, at about 15:50 UTC by Dr. Rhodes Hart. Each one is a 30 second exposure -- short by astronomical standards, but just long enough to get the nebula without completely saturating the bright stars. He took 4 images through 4 different filters, each one seeing sky in a different range of wavelengths or colors:



B:(375-490 nm) V:(490-580 nm) R:(570-690 nm) I:(725-1025 nm)

The filters are chosen for making precision measurements of the light from stars, not necessarily ideal for making color images that match what the eye sees. The eye is most sensitive at 550 nm, and only 10% as sensitive at 475 nm in blue, and 650 nm in the red. The CCD camera can sense wavelengths as long as 1100 nm.

Click the links to download the images to your own computer, or if you are doing this in the astronomy lab, look for them in the directory called "Image Processing".

- ngc2070_I_ij.fits through the I filter ^[1]
- ngc2070_R_ij.fits through the R filter ^[2]
- ngc2070_V_ij.fits through the V filter ^[3]
- ngc2070_B_ij.fits through the B filter ^[4]

We have already done some processing on them for you by subtracting the response of the camera to no light (a so-called "dark frame"), and by aligning the images so that when superimposed the stars and other features register.

Use ImageJ

If your own computer will run Java, then you may load ImageJ or run it from the web site. It is already installed on the computers in the astronomy lab. To access ImageJ from your own computer, use this link to our servers:

```
ImageJ for Astronomy<sup>[5]</sup>
```

Select "Download ImageJ for your own computer". We have a version that has additional features for astronomy, but you won't need those for this activity.

Once you have the software running you'll see a window that looks like this:

File Edit Image Process Analyze Plugins Window He	\times
	lр
$\Box \bigcirc \Box \heartsuit / \checkmark + \land \land \land \And / \land \land$	>>>

Switch to alternate macro tool sets

Click on *File* in menu bar to select an image. Other entries offer different options for display and processing. The best way to learn what they can do is to try them out once you have one or more images on the screen.

Load the images

Load the three images taken in visible light in the order R, G, and B:

- File -> Open -> ngc2070_R_ij.fits "R" is red
- File -> Open -> ngc2070_V_ij.fits "V" is the astronomer's visual or green
- File -> Open -> ngc2070_B_ij.fits "B" is blue

It's important to do them in this order because we will make a color image out of them, and the order determines which file becomes which color.

These will appear as *32-bit* images, that is values stored in them can be from large negative to large positive numbers.

Run the cursor around and look at the values in the ImageJ control window and you'll see something like this

but it will change as you move the cursor around. The *x* and *y* are the coordinates of the pixel, and *image* is the value representing how much light fell on the image at that spot during the exposure. The numbers may be negative! Oddly that may happen because of how they are stored in the file. Just remember that it's all relative to the lowest possible value, which would be black if no light fell in that spot. For these images black reads -32768.

The first thing you should do, then, is to process the images to change all the numbers in them. Since there are about 18,000,000 values in all three images, it's good to know how to make the program do this almost instantly.

Select the "R" image by a mouse click on the top bar of the window.

Process -> Math -> Add -> Value: 32768 -> OK

Repeat for the other two windows.

Make a stack

Notice that we did this operation on each image, one by one. It's simpler to manage several images at once by placing them together in what is called a *stack*.

Convert all the images to one stack with this simple action

Images -> Stack -> Images to Stack -> OK

Now there is a slider at the bottom of the image display, and if you move it from left to right you select one of the three images. Use the lower slider to select the red image of the stack. Notice that as you move the slider to the right there are three positions, and the "R", "V", and "B" image names will appear sequentially in the label of the image window. Select the leftmost position of the slider to have the red image.

If the new image display range after the math operation makes everything white, then the image may seem to be empty. Just run the cursor around and you'll see that the value changes from place to place.

Use

Image -> Adjust -> Window/Level -> Auto

on the red image. You can experiment with the sliders, but for a start simply let the software do its work with "Auto" which will adjust all three layers of the stack. You wil probably see some stars, but not much detail. The detail is in

the low signal levels that are not bright enough on the screen to see.

At this point if you are happy with the result you may save the stack by using

File -> Save as -> Tiff and then enter a name that you will remember, perhaps mystackname.tif

You may load this file and be back at this step instantly should something go wrong later.

Make a color composite

The smallest value in all three images should be around 0, the largest probably less than 32767. A bright star might be 16000 or so, but the nebula that is hidden in this image is only a few 100. If we used a scale of 0 to 255 with the star as 255, then the nebula would be around 255 * 100/16000 = 2. You would not see it. That's what happens on the display you are looking at. It takes a value from 0 to 255 and turns that into a brightness on the screen. At each element you see on the screen there are three numbers from 0 to 255 that tell the display how much red, green, and blue light to show.

Let's make a color composite image so that we can work with this and see the colors.

Image -> Color -> Make composite

The first image will be red, the second green and the third blue. You'll see the color composite of all three on the screen. Each layer has its full range of data, but what you see is limited by the screen's display to a range from to 255 values.

You should also see a "Channels Tool" on the screen that will allow you to turn off the effects of one or more layers. For now, keep them all on. Use the bottom slider to be sure you have the first layer selected.

Image -> Adjust -> Window/Level -> Set

and enter

Window Center: 500 Window Width: 1000

This means that values from 0 to 999 will be in the range from black to fully "on" on the screen. The bright parts of the image will probably look red.

Now select green "V" image and do this again, and then the blue "B" image.

The composite image you see will now probably look yellow. Much of the faint detail will be visible, but many bright stars will be white.

Modify the color

The remaining (difficult and arty) step is to get a color balance that you think is reasonable.

The camera is more sensitive to red light than it is to blue light (as is your eye), and slightly more sensitive to red than green. Your eye is more sensitive to green than to red or blue.

You can compensate for the lower sensitive for blue by increasing its importance in the final image. Select the blue (right) layer with the lower slider and set its image levels to something like this

Image -> Adjust -> Window/Level -> Set

Window Center: 100 Window Width: 200

Then experiment with the green and red to see if there is a combination that makes the nebula look white with a green cast.

As a trial you might try

Red: 350/750 Green: 300/600 Blue: 100/200

If you get something you like, save it as a tif file with a unique name. That way you can recover it later if you need to.

You may also save a jpg image that you can email, display or modify by saving as a jpg:

File -> Save as -> Jpeg

Make fine detail visible

Lastly you can process the composite to create effects to highlight structure. When an image is masked, you replace selected pixels in the image with any value you want. With unsharp masking, an image is intentionally altered to blur it and remove fine detail. This fuzzy image is then subtracted from the original to reduce the influence of coarse structure and leave the detail of the original in an enhanced final image. Let's try this trick with ImageJ and our color image of the Tarantula Nebula.

Start with the last saved tif, because the next step will destroy the intermediate image.

For example,

Process -> Filters -> Unsharp Mask

Radius: 1 pixel

Mask Weight 0.6

will gently highlight the detail. It also makes noise more apparent, so while larger values are useful on some images, this is about as much as tolerable on the Tarantula example.

But there are other tricks! Use a larger unsharp mask radius, and then try a little "Gaussian blur". It gets very interesting. Be creative, and look for a display that seems reasonable and shows features in the nebula.

1. Save the image as a jpg file. You will need to submit it for credit for this activity.

Compare to the sky

Change the orientation of the image

When a camera is attached to a telescope it may be rotated around in almost any direction. We'll see too that as a telescope looks in different directions in the sky, the image may have a different orientation on the camera's detector. Add the complication that either optically or in how the data are stored the image may be inverted top to bottom and/or left to right. Let's see if we can figure out what the orientation of the image you have made actually is, and then modify the image so that north is up and west is to the right, as it would appear in the sky.

Exit ImageJ if it is still running, start it again, and load the processed color jpg image you saved from your work. In the ImageJ menu bar there is an entry Image, and under that Transform, which leads to Flip and Rotate options that you will use. First, however we need to see what this nebula looks like in the Digital Sky Survey that is a reference archive of the whole sky. One way you can access this through Sky-Map with the web site

Sky-Map^[6]

Once you are in that site, enter *Tarantula* in the search box and you'll see an image of the nebula in the Large Magellanic Cloud. Compare that you your image and see if you can figure out how your image must be transformed to match the north up and west to the right standard.

Alternatively, you have more control in the display if you use Aladin. Use our gateway to Aladin

Aladin^[7]

and select the option "Run the Aladin Sky Atlas applet in your browser", or if you already have it installed on your computer you may run it without needing the browser. After it has started, enter *Tarantula* in the search box and Aladin should load a color version of the Digital Sky Survey. This will be roughly on the same scale as your image. The color version is over saturated and may be hard to compare with your short exposure image. Instead, now click the "DSS" button that will load a single gray scale blue image. At this point you will probably start to recognize the features.

2. How must you transform your image (flip and/or rotate) to make it match the image of the DSS with north up and west to the right?

The Tarantula Nebula is about 170,000 light years away, in another galaxy. It's as far as our telescopes can pick out individual stars at birth, and at that mostly the brightest ones. You'll notice some stars in the image are very red. These are most likely red giants, massive stars that have already used their supply of hydrogen in their core, and are evolving toward becoming a supernova.

There is an image of the Tarantula taken with a 1.5 meter telescope of the European Southern Observatory on their website that is also worth comparing to yours.

ESO Tarantula image^[8]

Compare the infrared

The I or infrared image is taken at wavelengths your eye cannot see. It will show cool stars, and will not see hydrogen gas which emits red light at 656 nm and blue-green light at 486 nm, but nothing in the I band. Let's see what the nebula would look like in false color, where the infrared is displayed as red, red light is displayed as green, and green light is displayed as blue. We'll skip the blue image, and load instead these three

- File -> Open -> ngc2070_I_ij.fits "I" the infrared will be red
- File -> Open -> ngc2070_R_ij.fits "R" the red will be green
- File -> Open -> ngc2070_V_ij.fits "V" visible or green will be blue

Go through the same processes you did to change the zero, create a stack and change the mapping of the intensity to the display in a composite color image.

3. Save the resulting color jpg image with I, R, and V band components for credit.

Let's compare the two images you have made.

4. Why is the nebula in the infrared composite green, rather than red?

5. Identify a star that is extremely cool compared to most stars in the image. A cool star emits more infrared light than red light.

References

- [1] http://prancer.physics.louisville.edu/classes/108/topics/image_processing/ngc2070_I_ij.fits
- [2] http://prancer.physics.louisville.edu/classes/108/topics/image_processing/ngc2070_R_ij.fits
- [3] http://prancer.physics.louisville.edu/classes/108/topics/image_processing/ngc2070_V_ij.fits
- $\label{eq:lasses} \end{tabular} \end{tabul$
- [5] http://www.astro.louisville.edu/software/astroimagej/imagej/
- [6] http://www.sky-map.org
- [7] http://www.astro.louisville.edu/software/aladin/
- [8] http://www.eso.org/public/images/tarantula/

Article Sources and Contributors

Experiment with CCD Camera Images Source: http://prancer.physics.louisville.edu/astrowiki/index.php?oldid=300 Contributors: WikiSysop

Image Sources, Licenses and Contributors

File:ubvri_filters.gif Source: http://prancer.physics.louisville.edu/astrowiki/index.php?title=File:Ubvri_filters.gif License: unknown Contributors: WikiSysop File:imagej_window.jpg Source: http://prancer.physics.louisville.edu/astrowiki/index.php?title=File:Imagej_window.jpg License: unknown Contributors: WikiSysop

License

Attribution-Noncommercial-Share Alike 3.0 Unported http://creativecommons.org/licenses/by-nc-sa/3.0/