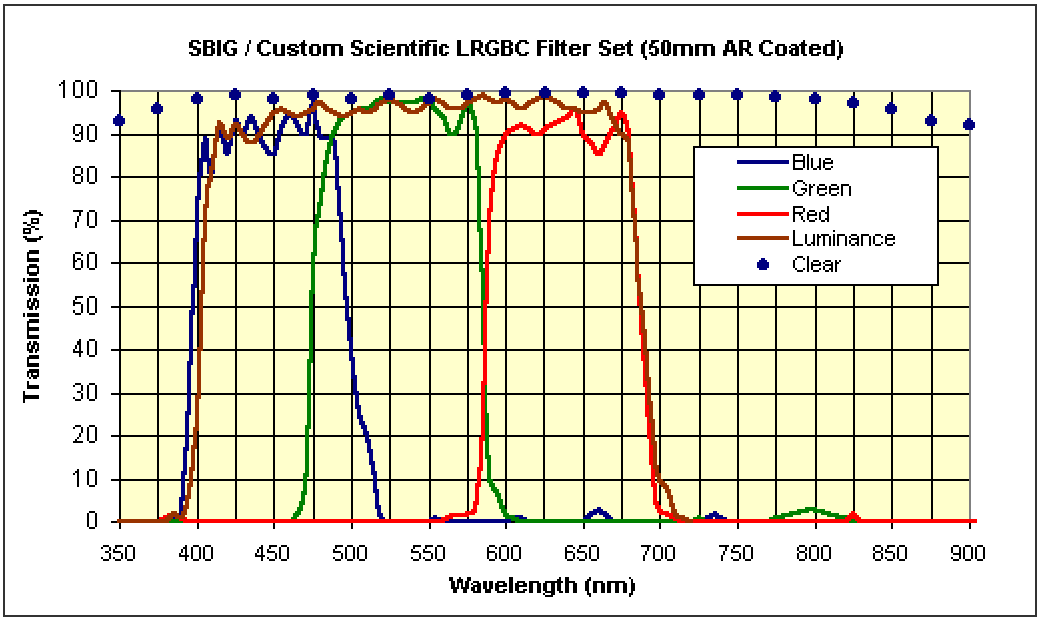
Final Project: Color Deep Sky Imaging

(Due: 2008 May 12)

Background Information on Color Imaging

Everyone has chosen to do true-color deep sky imaging for their final project. You have already learned in Lab 7 how to take a process a deep sky image using one filter (the Luminance filter). This write-up explains how to extend that to processing of color images (images taken with the red, green and blue filters—otherwise known as R, G and B filters). Figure 1 shows a spectrum of the R, G and B filters, and the luminance filter (brown curve). The y-axis is the transmission (the percentage of light that is allowed through). The curves are quite flat, letting about 90% of the light through, but only over a restricted range of wavelengths corresponding to the colors red, green and blue. The luminance curve shows that all of the colors are allowed through that filter, but ultraviolet and infrared are not allowed through.



The idea behind color imaging is to take images in each of these filters and then combine them in a true-color form that computers use. This is a good conceptual match to how computers display an image, since any color can be broken into its “primary” colors of red, green, and blue. Computers can use separate information in red, green and blue to form a color image. Our goal is to make high-quality color images using these filter images. One obvious method is to take the best possible image in each of the colors, with full spatial resolution and good signal-to-noise ratio, and then use the red filter image to drive the red part of your computer display, the green filter image to drive the green part of the display, and the blue filter image to drive the blue part of the display. This is called and RGB image. However, the problem with this is that it requires long time exposures in each of the filters, at full spatial resolution. Instead, astronomers have developed a more effective method called LRGB, which utilizes the RGB images at lower spatial resolution and generally lower total signal to noise, while using a high-quality luminance (L) image to determine the brightness and "shape" of the color. The RGB images are taken at lower resolution for two reasons: (1) Lower-resolution images combine pixels for better signal-to-noise ratio. A 2x2 binned image has four times the signal because it uses four pixels, and 2 times the noise because noise increases as the square-root of time, e.g. Lab 3. Thus, a 2x2 image has twice the signal-to-noise ratio as a 1x1 image of the same exposure time. (2) Color information generally does not change as sharply in the image as brightness. A star-forming cloud is generally red while a galaxy spiral arm may appear blue, and even though the features may change on small spatial scales, the color of the features does not. Therefore, one can take adequate color information with a smaller investment in time, and spend more exposure time on the more critical luminance (brightness) information.

Brief Overview of Steps for Color Imaging

Very briefly, the steps are to take a series of images binned 2x2 in R, G, and B, and then a longer series of images in L. This series of images is then calibrated (you must separately obtain 2x2 binned bias, dark, and flat images, with the flats taken in R, G, and B filters, and 1x1 binned bias, dark, and flat images in L). The calibrated images are then separately combined for each filter, using the same techniques you used in Lab 7 for the L image alone. At this point, you will have three color-filter images (2x2 binned R, G, and B images), and one 1x1 binned L image.

The next step is to align and scale the images so that you have 1x1 binned images in all filters (L, R, G, B) with exactly the same alignment. To do this, first use the Process/Double Size menu item in MaxIm DL on each of the R, G, and B images, then select Process/Align from the menu and align the L, R, G and B images using the L image as the reference. After enlarging the images, the R, G and B images will be noticeably fuzzier, but that is okay. Save the scaled and aligned images as FITS files, making sure to set the Size Format to 16-bit Int in the Save As… dialog window.

The next step you are familiar with from Lab 7—read the images into IDL and use a logarithmic scaling to make the fainter and brighter parts of the image look nice on the 8-bit computer display. However, it is important to first set the scale of the R, G and B images to the same linear brightness scale and then apply the same nonlinear scale identically for all three images. Here is how to do that. Regular font is used for IDL commands, and the equivalent Matlab commands are shown in italics.

Adjust the R, G and B images to the same linear scale

Because the CCD chip responds more strongly to red light than blue light, and the atmosphere also affects color, we have to correct for the low short-wavelength sensitivity by boosting the scale of the green and blue images. The factors (ratios) for R:G:B should be found using the graph in Figure 7 of the Lab 4 instructions. To use the equations there, you need to determine the mean altitude of your object during the time of your observations. Go into Cartes du Ciel for the date and middle time of your data, and determine the altitude, in degrees. Convert to zenith angle (ZA) and then to air mass (X). Enter this value of X into the three equations to get R, G and B magnitudes, then use the magnitude differences G – R and B – R to determine the flux ratios f_R/f_G and f_R/f_B. You will then correct the image scales in the ratio R:G:B = 1: f_R/f_G : f_R/f_B. As a sanity check, your numbers should be around f_R/f_G = 1.15 and f_R/f_B =2.5. Using these as an example (*use your actual values here, not these example values!*), say that you read your images into IDL (*Matlab*) [See the Lab 2 instructions for commands to read in a fits image.] as imgr, imgg, imgb and imgl. Then simply use the lines

```
imgg = imgg*1.15
imgb = imgb*2.50
imgg = imgg*1.15;
imgb = imgb*2.50;
```

This sets the scaling, but in general there is also a difference background level. You need to set the background levels very carefully to the same level. Otherwise your final image will have an overall color cast to it. For example, if the red level is too high, the image will look too reddish. To set the image backgrounds to the same level start with a line plot of a single row of the image, taken from an area of relatively flat background. Ignore any parts of the line plot that cross the object (galaxy, gas cloud, stars) and only look at the flat background.

Here is an example. The row that you plot, and the values you have to add, will be different from this example, but hopefully this shows the general idea. Figure 2*a* shows the result of the following commands to plot row 500 of the three images for one particular example:

```
plot,imgr[500,*]
oplot,imgg[500,*]
oplot,imgb[500,*]
plot(imgr(:,500));
hold on
plot(imgg(:,500));
plot(imgb(:,500));
hold off
```

You can see that the backgrounds are quite different. The R image curve is the top one, the G image curve is the middle one, and the B image curve is the bottom one. Examining the y-axis scale, you can estimate that the G curve needs to move up around 80 units, while the B curve needs to move up about 185 units. Figure 2*b* is the result of the following commands:

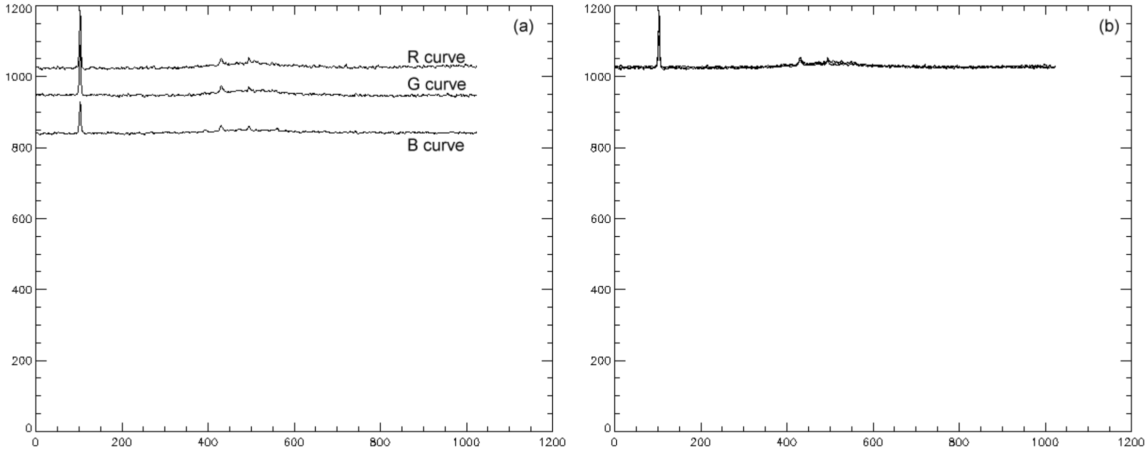
```
plot,imgr[500,*]
oplot,imgg[500,*]+80
oplot,imgb[500,*]+185
plot(imgr(:,500));
hold on
plot(imgg(:,500)+80);
plot(imgb(:,500)+185);
hold off
```

The values of 80 for green and 185 for blue are background “offsets.” Now the three curves lie pretty much on top of one another. However, some even finer adjustment is possible. You can “zoom in” by restricting the y-range of the plot. For example, the background of these images is somewhere between 1000 and 1100, so I issued the commands

```
plot,imgr[500,*],yrange=[1000,1100]
oplot,imgg[500,*]+80
oplot,imgb[500,*]+185
plot(imgr(:,500));
hold on
ylim([1000,1100]);
```

```
plot(imgg(:,500)+80);
plot(imgb(:,500)+185);
hold off
```

to see if further adjustment was necessary, and found that the green offset should really be 78.



Once you determine the background offsets, go ahead and add them, i.e.

```
imgg = imgg+78
imgb = imgb+185
imgg = imgg+78;
imgb = imgb+185;
```

At this point, your R, G and B images should all have the same linear scale.

Adjust the R, G and B images to the same nonlinear (logarithmic) scale

Following the same procedure as in Lab 7, adjust the red image until it looks pleasing to the eye. I ended up with the command (again, your numbers will differ!):

```
tv,bytscl(alog10((imgr-1000)>20<600))
colormap(gray(256));   % only do this once
imagesc(log10(abs(imgr-1000)),log10([20,600]));
```

Once you have it as you like it, plot imgg and imgb using exactly the same scaling:

```
tv,bytscl(alog10((imgg-1000)>20<600))
 imagesc(log10(abs(imgg-1000)),log10([20,600]));
tv,bytscl(alog10((imgb-1000)>20<600))
 imagesc(log10(abs(imgb-1000)),log10([20,600]));
```

Check that they look okay, and make further adjustments if needed. Try to leave the background not-quite black. Remember to check your final numbers on all three images, and make sure that you use the same numbers for each. When you are happy, save the images as BMP files, e.g.

```
   tvlct,/get,r,g,b
   write_bmp,'M74-R.bmp',bytscl(alog10((imgr-1000)>20<600)),r,g,b
   write_bmp,'M74-G.bmp',bytscl(alog10((imgg-1000)>20<600)),r,g,b
   write_bmp,'M74-B.bmp',bytscl(alog10((imgb-1000)>20<600)),r,g,b
map = colormap(gray(256));
imr = (log10(abs(imgr-1000))-log10(20))*(256./(log10(600)-log10(20)));
imwrite(uint8(imr),map,'M74-R.bmp','bmp');
img = (log10(abs(imgg-1000))-log10(20))*(256./(log10(600)-log10(20)));
imwrite(uint8(img),map,'M74-G.bmp','bmp');
imb = (log10(abs(imgb-1000))-log10(20))*(256./(log10(600)-log10(20)));
imwrite(uint8(imr),map,'M74-B.bmp','bmp');
```

Note that both the file names and the image names (imgr, imgg, imgb) are changing in each of these lines. For Matlab, we have to do the scaling explicitly, so we put the scaled images into new variables, imgr → imr, imgg → img, imgb → imb and write out these new variables as bitmap images.

Finally, do the same procedure as for Lab 7 on the L image. For this one, you can and probably should change the scaling numbers for the most pleasing image. They should not be the same as for the color images. Save your L image as a BMP file also.

Combining the Color Images in Photoshop

The final step of combining the images in Photoshop is perhaps the least well defined. The goal is to make a pleasing image, but there are a large number of ways to accomplish that. The basic idea could not be simpler.

1. You read in each of your four BMP images (the R, G, B and L images).
2. Take one of the color filter images (say the B image) and select the Image/Mode/RGB Color menu item.
3. Find the Channels palette (you can select it using the Window/Channels menu command), and click the Green channel so that it is the only one selected.
4. Now find the Green filter image, click on it to select its window, type Ctrl-A to select the entire Green filter image, type Ctrl-C to copy it, and paste it (with Ctrl-V) into the Blue filter image window. Since only the Green channel is selected in the B filter image, the pasted Green filter image will replace the Green channel of the B image.
5. Now select the Red channel of the B image, find the R image, and repeat this copy, paste operation to paste the R image into the Red channel of the B image. The B image is now an RGB image.
6. The next step is to place the L image into this new RGB image as a separate layer. To do that, find the Layers palatte (it is the next tab over from the Channels palette, or select the Window/Layers menu item). Do the same cut and paste operation to paste the L image as a new layer in the RGB image.
7. The final step is to select only the L layer (the new layer that you pasted in the RGB image) and set the overlay property from Normal (in the drop-down list in the Layers palette) to Luminosity (found all the way down at the bottom of the drop-down list). This will use the L image as the brightness, but will use the underlying RGB image to color it.

The above list is the basic technique. However, you can manipulate both the RGB image (and the individual channels using the Channels palette) and the L image to scale the brightness (transfer function), sharpen or smooth different parts of the image, etc. There is really an infinite number of ways to do this. You might find some clues in the following web site, which is an excellent introduction, but using quite sophisticated techniques: http://www.mistisoftware.com/astronomy/Process_m8.htm

Conclusion

The idea of this project, like Lab 7, is to make the nicest pretty picture possible. Experiment with some of the Photoshop tools. I am happy to offer suggestions for your specific case. There is a huge difference between an image done badly and one done well. If you do a careful job, your image will be something to be proud of, and will impress your friends and family.

The Write-Up

When you write up your final project report, include the following:

1. Which object you imaged, what kind of object it is, what its designation (name) is, and what constellation it is found in.
2. How many frames of which filters you took, the frame durations, the camera temperature, etc.
3. The procedure for processing the images, including how many frames you had to throw out, and any special details you think are important in how you processed the images.
4. What image scaling you had to do in the linear part (scale and offsets applied) between the R, G and B images.
5. What nonlinear scaling you did for RGB and L images.
6. What general steps you did in photoshop to combine your images into LRGB, and any special problems you had.
7. A figure showing the raw L image.
8. A figure showing the final image after completion.
9. A short section describing the subject of your image, based on web research. Why is it interesting? Check out www.seds.org, e.g. the description of M74 is at http://www.seds.org/messier/m/m074.html