1 Introduction

The electronics associated with a CCD typically include clocking circuits to move the charge in each pixel over to a shift register, the shift register which reads out the pixels one by one, an analog charge-to-voltage amplifier (both the shift register and the analog amplifier are usually part of the CCD chip itself), and an analog-to-digital converter. The original charge packet consists of one electron per detected incident photon. These are subject to standard counting statistics. The amplifier introduces some additional noise, which may vary from pixel to pixel by a very small amount but which is generally independent of pixel and of signal level. The amplifier and A/D converter also introduce offsets of the zero-point in the signal. Proper use of the CCD requires an understanding of the noise in the data. This project is designed to measure the noise properties of the system.

There are basically two parameters that must be measured. The first is the effective gain, G, of the amplifier plus A/D system. In other words, we want to know the number of A/D output units produced per initial photo-electron. This gain is usually expressed as the reciprocal gain, K, the number of electrons per digital count of output. The other parameter is the amount of read-noise introduced by the output amplifier. This is usually measured in A/D units and then expressed in terms of the number of equivalent electrons.

You will use measurements of a step wedge to provide signals covering a well-defined range of intensities to determine these parameters, using the theory of statistical fluctuations to determine the number of electrons.

2 Preparing the Equipment

After the new Apogee CCD camera was installed on the 20-inch telescope, we moved its old Photometrics CCD camera and electronics to the observatory central bay. We have a crude improvised set-up so that you can carry out this exercise using that old CCD camera.

- Keep a log of everything you do and when you do it.
- Turn on the CCD cooling system in sufficient time to be sure it is cold by the time you begin taking data. Try for a two-hour lead time but one hour may be sufficient. (During the week of this lab, we will probably leave the CCD on all the time, so that you will not have to wait for things to stabilize.)
- If it is not already on, turn on the rest of the CCD system using the separate instructions.
- The step wedge and lamp should be inside the left-hand end of the cardboard tube on the table. The electronics cart will be set up nearby. Locate the battery power-pack and plug in the lamp. Turn it up so you can see the glow of the step wedge.
- The transfer lens is mounted on an aluminum plate, which is fastened by two thumb screws to the face of the CCD camera, which is at the other end of the cardboard tube.
• With the flip-mirror in place, look in the CCD eyepiece and adjust the position of the tube and CCD until you can see the wedge. You should turn off all the room light – there will be enough light from outside to see what you are doing.

• Move the tube back and forth until the wedge is roughly in focus. Rotate the CCD camera until the the image of the wedge is centered – rotating the CCD raises and lowers it as the CCD chip is not on the central axis of the CCD housing. If you look carefully, you will see a rectangle in the eyepiece field of view – this corresponds to the CCD chip. Rotate the cardboard tube until the image of the step wedge fits into this rectangle. Finally, turn the lens on the CCD (it screws in and out) to get the best focus.

• Flip the mirror out of the optical path, turn down the lamp brightness, and take some practice images of the step wedge. Adjust the brightness of the lamp and the exposure time to produce a good image. If necessary, adjust the focus by screwing the transfer lens in or out a small amount. A well focused image is important.

3 Making the Measurements

Be sure the lights are all out. Use the cloth shroud to shield the gap between the tube and the CCD lens.

When you are sure that the CCD has reached equilibrium, you can begin taking data. First readjust the brightness of the lamp and the exposure time for optimum exposures. Ideally this means digital counts of about 10,000 at the brightest portions of the step wedge with a reasonably short exposure, but not less than about 3 seconds. The counts in the background should be several hundred.

Once all these parameters have been adjusted, your data consist of three images, taken in quick succession. The first and last images are of the step wedge and the middle one is a bias image. To take them in quick succession, use different image caches in memory. You can stack the commands for all exposures and cache changes one one line. It is important that nothing changes between the two images of the wedge! If the equipment moves or if the intensity of the light varies, even by a small amount, this will make analysis of the data impossible. Be careful to note the exposure times and start times of each exposure. After taking the three images, record with the comment command the exposure time, the start time, and the content i.e. ‘1st Step Wedge’ or ‘Bias’.

Now you should save your images to the hard disk. After you have written them, you can then exit the CCD program by typing ”quit”. You will next want to transfer your images to the computer back in the Astronomy Department. This can be done by just copying the images onto floppy disks. Just use the DOS command ”copy”:

    copy C:\filename.img A:

Note that CCD images are big: You can only get three of them on one high-density floppy disk. You may want to leave your images on the hard disk at the observatory, just in case you have a problem with the transfer, but we will erase them all at some time in the future. If you are back at the observatory after you are sure that your data are safely on the network in the astronomy department, please delete these images from the hard disk.

You will be working in groups of four but each student must end up with his or her own separate set of images.
4 Analysis

The analysis is done using IDL on the PCs in the computer lab. The first step is to change the filename.img files into filename.fts files. You do this by running a program called img2fits. To convert the file 'filename.img’ type

    img2fits  filename.img

You will find a new file with the name 'filename.fts’ in your directory. Do this for each file you wish to convert.

Make a note of the names of your .fts files: two exposures of the step wedge and the bias frame. Now start up IDL (type “idl”). Read in the images by using the readfits procedure. If your file were called mystep1.fts (with a header file mystep1.hdr), you could type

    s1=readfits('mystep1.fts',head1)

Then, the variable s1 will contain the image and the variable head1 will contain the header (which you can read by typing “print, head1”). To look at the file, you may type

    tvscl,s1

The numbers in the image arrays will be of integer type; to avoid all sorts of possible troubles, you should convert them all to floating point arrays. You can do this with a statement like

    s1 = 1.0*s1.

To display the files with the long axis horizontal (which is my preference), you may rotate the file using “s1=rotate(s1,3)”. (If you rotate one, make sure you rotate all the others!) To see the display in (false) color, you can load a color table (e.g., “loadct,15”) or choose one interactively by typing “xloadct” and going to the pop-up window.

Now, suppose you have the image displayed. You can examine the values of the pixels at any point of the image by typing

    curval,s1

(assuming the image is called “s1”). Now when you move the mouse about on the image, you will see the x and y coordinates of your location on the image, and (under the Value column) the intensity of that point. To record any particular location and value, just click the left mouse button. To exit curval, click the right mouse button. Another useful command is zoom, which lets you look at any part of an image in detail. Type “zoom”, then click on the part of the image you want enlarged.

The mean value of the bias image is the mean offset in the zero point introduced by the amplifier and A/D system. The noise of the amplifier produces random scatter in the value of the bias from one pixel to another. If your bias image is called “bias”, then the mean value of the image can be obtained by typing “print, mean(bias)” – mean is an ILD function that sums all the elements of an an array and then divides the total by the number of elements. You can compute the root mean-square-value of the bias image (the read out noise) by first subtracting the mean bias value from each pixel value (i.e., “del=bias - mean(bias)”) to get the fluctuations about the mean, $\Delta_B$, and then taking the square root of the mean of the square of $\Delta_B$: “print, sqrt(mean(del*del))”.

To investigate the Poisson noise due to the photon statistics of the illuminated CCD, we first correct the two exposures of the step wedge by subtracting off the bias image. Thus

    sc1 = s1 - bias & sc2 = s2 - bias

We then form two new images:(a) the average of the two corrected images sc1 and sc2, and (b) the square of the difference between the two uncorrected images – this is the variance. You should in principle use the raw images s1 and s2 (not sc1 and sc2) to form the difference, since using the
corrected images could introduce extra noise. (Though if the same bias frame is used to correct both, it will cancel exactly.) See the theory write-up for the derivation of the equations you will use. In particular, equation (20) is fundamental.

You will need to define two boxes on each step of the step wedge, each containing several hundred pixels, and located well away from the edges of the steps. Also try to avoid obvious blemishes, etc. (In principle, one box per step is enough, but the second allows an important check on the results.)

The easiest way to get the boxes is with the box cursor procedure. It is called as follows:

```
box_cursor,x0,y0,nx0,ny0
```

When you enter this command, a box will appear on your image. Don’t worry that is is bigger than you want, it can be resized. The left mouse button lets you drag it to where you wish. The middle mouse button, when held down near a box corner, lets you move that corner, reshaping the box. When you have the box in the position you want, click the right mouse button. The box will disappear. Now, the variables x0 and y0 contain the coordinates of the lower left corner of the box, and nx0 and ny0 are the width and height of the box. That’s all you need! For example, if the entire image were \( \text{sc1} \), then the sub-image of pixels within the box is just given by

```
\text{sc1\_box0} = \text{sc1}[x0:(x0+nx0),y0:(y0+ny0)]
```

Furthermore, the exact same box on another image, e.g. the variance \( \text{var} \), will be

```
\text{var\_box0} = \text{var}[x0:(x0+nx0),y0:(y0+ny0)]
```

You can then take the mean of this new sub-image var_box0, etc. If you want to define all the boxes at one go, you just call box_cursor repeatedly, with different names each time for the coordinate variables:

```
box_cursor,x1,y1,nx1,ny1
box_cursor,x2,y2,nx2,ny2
```

You should determine the mean and standard deviation for each box, both in the average image and in the variance image.

I have written a simple procedure called box_vals which will automate the above steps somewhat – see the appendix.

From these data, you are to determine the gain, \( G \) [ADU/electron], the reciprocal gain, \( K=1/G \) [electrons/ADU], the readout noise, \( \sigma_B \) [ADU], and the readout noise in equivalent photoelectrons, \( R = K \sigma_B \). The appropriate equations were derived in class by assuming that the noise in the signal itself is due entirely to counting statistics and that the noise in the readout is a fixed value, whether expressed in digital units [ADU] or in equivalent electrons.

Use the IDL plot command to graph the variance vs. the mean intensity for your measured (box) data. Fit this data to a straight line. Since a straight line is a polynomial of degree one, you can just use the IDL poly_fit procedure

```
coef=poly_fit(xvec,yvec,1)
```

where xvec is a vector of x-values and yvec the corresponding y-values. The result, coef, is the vector of coefficients of the polynomial fit – in this case, just the equation

```
y(x) = a + b*x,
```

where \( a=\text{coef}[0] \) and \( b=\text{coef}[1] \).

There are two ways to determine the read-out noise, \( \sigma_B \). One is from the intercept of the least-square fit. The other is directly from bias image. Compute the mean and the standard deviation
of the whole bias image. Now try cutting out a few boxes from the bias image, and compute the
mean and $\sigma_B$ in those boxes. Is there a large difference? *It is the $\sigma$ from the boxes that gives you a true measure of the read-out noise.* Explain why the value from the boxes is better than the value from the whole image. How does this result compare to the intercept of the least-square fit? Which result do you think is the most reliable? Why?

5 Report

Your write-up should give an overall description of the objectives of the lab exercise, your analysis and your results. To get an idea of how reliable your value of $G$ may be, you should fit 3 least squares lines: One from the first of the pair of boxes you measured, another for the values measured from the second of the pair. The difference between these two fits is a crude measure of the accuracy of $G$. Then, your best value for $G$ will be from a least squares fit to all the data taken together. You should submit the following items:

- Your log recording all aspects of the data-taking session.
- A record of the values measured during the analysis.
- A hard copy of the plot of variance vs. mean intensity, including both the data points and the line from the least squares fit.
- All the basic equations, and a description in words of what they mean.
- Your final values for $G$, $K$, $\sigma_B$, and $R$ and your estimate of their uncertainties.
- Your raw images should be available also; please leave them in your directory on **ursa**. I may want to look at them.

*Never turn in tables or plots where you have not labeled the quantities tabulated or plotted!*

**Due: 3 October 2006**
6 Appendix

The following IDL procedure is in your directory. Read the commands to see how it works. Suppose you have obtained the average of the bias corrected images and have called it "sav". Suppose that the variance image is called "sdif2". First display "sav" and then invoke the procedure with "box_vals, sav, sdif2". A box will appear in the image which you can place where you wish. When you click the right button, the means and sigmas of the box pixels of both "sav" and "sdif2" will be printed on the screen.

```idl
pro box_vals, av, difs
; interactively cut the same sub-images from two images
; called "av" and "difs", then compute means and sigmas
; of the pixels in these sub-images.
print,'Mouse: Left=drag box; middle=resize; right=cut.'
box_cursor, x0, y0, nx, ny
avbox=av[x0:(x0+nx),y0:(y0+ny)]
difbox=difs[x0:(x0+nx),y0:(y0+ny)]
avm=mean(avbox)
dfm=mean(difbox)
sigav=sqrt(mean((avbox - avm)^2))
sigdf=sqrt(mean((difbox - dfm)^2))
print, avm, sigav, dfm, sigdf
return
end
```