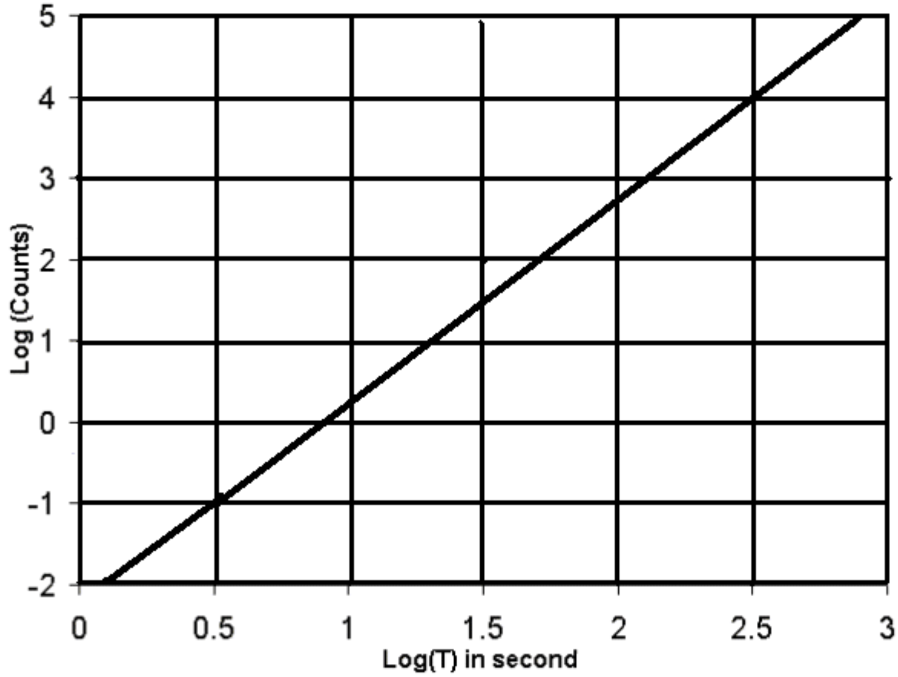
Image Math - Dark Current



A CCD chip is an electrical device in which electrons are counted in each pixel in response to the amount of light applied. However, electrons can be generated inside each pixel even though no light is applied and the camera is in a perfectly dark location with its shutter closed. This happens because the camera CCD is warm and the jiggling of atoms can shake loose electrons that get trapped in each pixel over time. This phenomenon is called 'dark current' and the data counted after each exposure has to be corrected for this effect in order to get an accurate count of the actual scene brightness. If the dark current is too high, and the exposure time is too long, a CCD can easily accumulate so many dark current electrons that there is no room left over to count the actual electrons before the pixel becomes saturated!

Problem 1 - The graph above gives the Log_{10} of the number of dark current electrons generated in three pixels after the amount of time has elapsed, which is also given in Log_{10} units for a selection of pixels in an array. For example, $Log_{10}(T)=1.5$ means 31.6 seconds. The value on the y-axis is $log_{10}(E) = 2$ so E = 100 electrons. A) What is the maximum number of dark current electrons defined by the vertical axis? B) What is the range of exposure times in seconds represented along the horizontal axis?

(A) 100,000 ($log_{10}(E) = 5$; $E = 10^5 = 100,000$)

(B) 1[s] to 1,000[s] ($log_{10}(T) = 0$; $T = 10^0 = 1$; $log_{10}(T) = 3$; $T = 10^3 = 1,000$)

Problem 2 - From the graph above, and rounded to the nearest integer, how many dark current electrons are accumulated by Hot Pixel Number 1 after an exposure time of about A) 10 seconds? B) 316 seconds? C) 10 minutes?

(A) 2.0[electrons] ($log_{10}(10) = 1$; $log_{10}(E) \approx 0.3$; $E = 10^{0.3} = 1.995... \approx 2.0$)

(B) 10,000[electrons] ($log_{10}(316) \approx 2.5$; $log_{10}(E) = 4$; $E = 10^4 = 10,000$)

(C) 100,000[electrons] ($log_{10}(10 * 60) \approx 2.778$; $log_{10}(E) \approx 5$; $E = 10^5 = 100,000$)

m5231153 Yuta Nemoto

Image Math - Flat Fielding

$$A = \begin{pmatrix} 24 & 28 & 37 \\ 25 & 30 & 21 \\ 26 & 38 & 42 \end{pmatrix} \quad B = \begin{pmatrix} 1.33 & 1.14 & 0.86 \\ 1.28 & 1.07 & 1.52 \\ 1.23 & 0.84 & 0.76 \end{pmatrix}$$

Because each pixel is a separate electronic device, no two pixels respond exactly the same way to the same intensity of light that falls on them. One pixel may count 1234 electrons while its neighbor counts 1267 electrons for the same light intensity. The CCD array can be corrected for this effect by photographing a perfectly uniform scene that has the same overall brightness as the scene you want to study.

Astronomers often use a photograph of the twilight sky before stars 'come out' (called a Sky Flat) or the inside of the dome at the observatory (called a Dome Flat). Both of these scenes have very smooth and constant brightness so by photographing them the pixel counts can be corrected for this effect. This process also removes actual geometric distortions in the optics of the camera so it is often called flat-fielding.

Problem 1 - Array **A** above is the raw data for a small section of an image. Array **B** is the Sky Flat array that corresponds to the same pixels in Array **A**. To 'flatten' Array **A**, create Array **C** defined so that $c_{ij} = a_{ij} \times b_{ij}$, where I and j are the row and column numbers for the pixel. (Example $c_{12} = a_{12} \times b_{12}$ so for the above arrays, $c_{12} = 28 \times 1.14$ and $c_{12} = 32$.)

Problem 2 - What can you conclude about the image that was taken by this portion of the CCD camera?

$$R = \begin{pmatrix} 24 & 28 & 37 \\ 25 & 30 & 21 \\ 26 & 38 & 42 \end{pmatrix} \qquad B = \begin{pmatrix} 15 & 15 & 15 \\ 15 & 15 & 15 \\ 15 & 15 & 15 \end{pmatrix}$$

$$D = \begin{pmatrix} 1 & 2 & 1 \\ 3 & 1 & 2 \\ 1 & 3 & 2 \end{pmatrix} \qquad F = \begin{pmatrix} 1.25 & 0.91 & 0.48 \\ 1.43 & 1.28 & 2.50 \\ 1.00 & 0.50 & 0.40 \end{pmatrix}$$

After an image is collected, called reading-out the CCD, the raw pixel counts have to be corrected for dark current counts and for flat fielding. There may also be a constant number of electrons in each well which will 'bias' the numbers up or downwards. Correcting for array bias, dark current and flat-fielding is done mathematically.

If the raw image read-out produces Array **R**, the dark current counts recorded for an identical exposure time is given by Array **D**, the bias counts are defined by Array **B**, and the flat-fielding correction is given by Array **F**, then the final, corrected image will be **C** = (**R - D - B**) x **F**. For example, for pixel (1,3) we would have C_{13} = $(R_{13} - D_{13} - B_{13})xF_{13}$, so that if R_{13} = 37, B_{13} = 15, D_{13} = 1 and F_{13} = 0.48 we would have C_{13} = (37-15-1)x0.48 and so C_{13} = 10.

Problem 1 - Based on the array values given above, what are the values for Array **C** after all of the corrections have been applied?

Problem 2 - The astronomer was hoping to detect a faint star in this image. In what pixels do you think the star is located?

Problem 3 - How much brighter do you think this star is compared to the background sky?

$$A = \begin{pmatrix} 23 & 25 & 28 & 26 \\ 27 & 29 & 26 & 28 \\ 25 & 29 & 21 & 22 \end{pmatrix}$$

A = {23, 25, 28, 26, 27, 29, 26, 28, 25, 29, 21, 22}

Once the data for each pixel have been read-out, this information needs to be transmitted back to Earth from the satellite. To do this, the individual numbers that represent the individual pixel counts have to be transmitted sequentially in a carefully defined stream of data. In the example to the left, a 3x3 image is converted into a string of numbers.

To properly encode and decode the data string, the format of the image must be known (3 x 4) along with the reading sequence {a11, a12, a13, a14, a21, a22, a23, a24, a31, a32, a33, a34}.

Problem 1 - An image is obtained by a satellite sensor and reduced to the data string {11, 14, 12, 12, 18, 15, 21, 16, 17, 25, 19, 17, 4, 8, 13, 16, 5, 9, 20, 32, 12, 7, 19, 13, 11, 13, 14, 21, 16, 8} If the format is a 5x6 image, what will the array look like when it is recovered from the data string?

Problem 2 - During transmission, the 13th data word in the string in Problem 1 was corrupted. Which pixel in the image was damaged during transmission?

Problem 3 - What is the data string that corresponds to the following image array?

$$I = \begin{pmatrix} 23 & 22 & 24 & 23 & 24 \\ 24 & 83 & 24 & 25 & 22 \\ 22 & 22 & 23 & 23 & 24 \\ 22 & 24 & 23 & 24 & 79 \\ 23 & 22 & 22 & 23 & 24 \\ 24 & 23 & 24 & 22 & 25 \\ 23 & 24 & 23 & 65 & 22 \\ 99 & 24 & 23 & 25 & 22 \end{pmatrix}$$

Problem 4 - In which positions in the data stream sequence are the pixels I_{22}, I_{45}, I_{27} and I_{18} found, and what are their values?

Telemetry and Binary Numbers

Data	Binary	Data	Binary
1	0001	10	1010
2	0010	11	1011
3	0011	12	1100
4	0100	13	1101
5	0101	14	1110
6	0110	15	1111
7	0111		
8	1000		
9	1001		

The data for each pixel that describes the number of electrons counted in an image are represented as a string of numbers. To transmit these numbers from the satellite to Earth, they have to be translated into a binary string of data consisting of a pattern of '1' and '0' called binary numbers.

The scheme to the left shows how normal numbers are represented in this way.

Problem 1 - The pattern above shows how the numbers 1-15 are represented in a 4-bit data word. How would you write the same numbers in an 8-bit data word?

Problem 2 - What is the largest number you can write in a 16-bit data word?

Problem 3 - How would you write the number 149 in an 8-bit data word?

Problem 4 - In designing a satellite telemetry set up, you agree to send the pixels in a 2x3 image as a sequence of 8-bit data words. If the image is given by the array of numbers below, what is the A) sequence of pixel numbers in normal form and B) the sequence of numbers rendered as a 4-bit telemetry stream?

$$A = \begin{pmatrix} 5,\ 7,\ 12 \\ 13,\ 6,\ 2 \end{pmatrix}$$

Problem 5- In the 4-bit binary telemetry string for the array in Problem 4, the 10th binary number is changed from a '1' to a '0' by a telemetry error during transmission. Which pixel was affected, and what is its new, incorrect, value?