Homework Assignment #2

Due Date: Friday, 20 September, 2013 at 5:00pm (MDT)

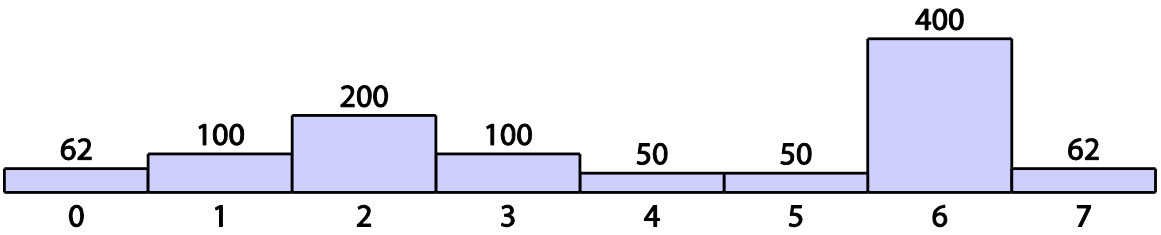
What to Submit

Please place everything you're submitting online into a single .zip or .tgz file. For this homework you should turn in:

1. Written Exercises (either submit with the rest or give separately to me in class on Friday)
2. Programming Exercise Write-up (as described below) and all code that you've written to complete the exercises.

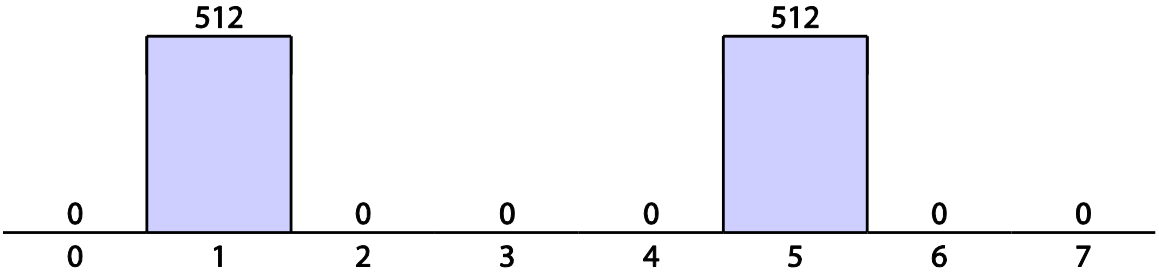
WRITTEN EXERCISES

1) Suppose that a 32x32 image with 8 grey levels has the following histogram:



Calculate the grey-level transformation that performs histogram equalization on this image and sketch what the transformation function looks like.

2) Suppose that you wanted to shape the histogram of the above image to match that of a second image:



Use the histogram specification (matching) technique described in class and your textbook to devise a grey-level transformation that shapes the histogram of the first image to look like the second. Make sure to show all intermediate steps in the calculations. Apply your transformation to the first image and compare the resulting histogram to that of the second image.

Be careful: the continuous math has some discontinuities when you do it with discrete values (especially function inverses), so you'll have to adopt a strategy to handle certain cases. If two input values map to the same output value, what does that imply about the image? How can you use that to handle inverting the mapping? Similarly, what happens if no input values map

onto a particular output value? What does that imply, and how can you use that? Make sure you describe these cases and how you handle them.

3) Suppose that you have a scanned document with text (near black), background (near white), and several grey-level images. After examining the document, you find that the image mostly have values in the range [50,200]. Devise a grey-level windowing approach that will best enhance the contrast of the document. Would histogram equalization be expected to do better?

The following exercises are from the Gonzales and Woods (G&W) textbook:

4) G&W 3.7

5) G&W 3.11

PROGRAMMING EXERCISES

Please complete the following exercises in your programming environment of choice (nominally MATLAB or Python/NumPy):

1) Open and display the image "mystery.png".
2) Calculate and display the **histogram** of this image. Unless you did this by hand on the first homework (most used either `hist()` or `imhist()`) , please write a function to create the histogram by looping over the individual pixels.
3) Use the level operations built into gimp (or a comparable program such as Adobe Photoshop [Image->Adjustments->Equalize], ImageMagick, etc.) to histogram equalize this image and display the result.
4) Write a program to implement **Histogram Equalization** and test it on "mystery.png". Display the histogram-equalized image side-by-side with the original image. (You might also want to compare your results with those from gimp/Photoshop/ImageMagick, etc.)
5) Plot the lookup table used to histogram equalize "mystery.png".
6) Calculate and display the histogram of the equalized image.

Prepare a brief write-up as described below.

Programming Exercise Write-up

Please prepare a brief written write-up (preferably submitted as a PDF) which includes the following:

- Description of your histogram equalization program [4]
- The image mystery.png, both before [1] and after histogram equalization [4].
- The histograms of mystery.png, both before [2] and after histogram equalization[6]. Theoretically, the resulting histogram should be uniform – why is it not?
- The lookup table used to histogram equalize the image [5].