[1]

**DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**

**NORTHEASTERN UNIVERSITY**

**ECE 1467 INTRO TO SUBSURFACE SENSING AND IMAGING Winter 2003**

## Homework Set 1

For each problem please turn in a *complete* answer. In particular you should include any narrative required to understand your method and comment on your results where appropriate, label and place captions on any figures you include, and in general think of your homework as a short report. The intent is *not* to have you do a lot of writing for the sake of writing, and you can certainly assume that we know what the problem statement was, but your homework should be professionally done and we should not have to guess at how to interpret what you hand in.

Also, you will note that there is some use of Matlab required to do this assignment. We realize that the Matlab background of the class may be quite varied, so we expect you all to ask for help as needed. However the responsibility is on each of you to ask, and to ask enough ahead of the homework due date that the request is reasonable and the response might be useful. This reflects what we described in the syllabus as the "graduate class" approach we are emulating: our guess is that the homework contains some challenges. If things are too challenging, you need to figure that out and ask about it ahead of time. So what we all know is the standard undergraduate approach of waiting until the night before to start the homework may be even more dangerous here than in most undergraduate classes.

### Problem 1: Granite Chief

Suppose that I wish to image a mountain 9,050 feet high, above a Valley at 6,200 feet, at a distance of 2 miles. The CCD consists of 640 by 480 pixels, spaced 10 micrometers apart. This is not a particularly good problem for hyperspectral imaging in the visible spectrum, because the mountain in question is almost completely white at this time of year (To be sure that this is correct, Prof. DiMarzio has decided to inspect the area personally.).

(a) What is the angular field of view? In other words, what angle does the target subtend as viewed from the camera?

(b) What magnification is required?

(c) What is the size of the 10–micrometer pixel projected onto the side of the mountain?

[1] 10472-1.tex — 13 January 2004 at 22:06

(d) What focal length lens is needed?

**Problem 2: The Band Picture** For this problem, we will use an example image on the course website. Download the files m10067_1.m and m10067_1.tif . The matlab script shows you how to read in the image file, and I have shown you how to generate some images with different values of $\gamma$, as in the lectures notes, on slide 10471-2-26. This is something I did just for practice.

Most cameras have eight bits of grey scale. Color cameras typically have eight bits per color, although displays often use less. These are sometimes called 24-bit cameras, but do not be misled. They do not produce $2^{(}24)$ different grey levels. Most cameras have something like 640 by 480 pixels, which is the VGA standard. Higher cost cameras have 10 or 12 bits of greyscale, and a few million pixels. A typical digital camera for about $300, now has a bit over 1000 by 1000 pixels. Higher numbers are possible. It is interesting to note that a 640 by 480 image contains about as much information as you can see on your thumbnail at arm's length.

(a) One way to understand the effect of changing the $\gamma$ value is to look at the changes in the histograms of the corresponding images. A histogram is a count of the number of pixels in the image at each possible gray level. (If we think of an image as a realization of a 2-D random variable (also called a "random field"), then the histogram, after being normalized by the total number of pixels, is an way of approximating the probability density function of the image.)

    If an image has poor contrast (for instance, is too dark or too light), then the histogram will tend to be compressed at either end. If the image has good contrast, meaning it makes good use of the available grey levels, then the histogram will tend to be evenly spread out. Since our visual system has difficulty resolving small contrasts in the presence of larger contrasts, it is often useful in enhancing an image to deliberately modify its histogram (by reassigning the pixel values) to increase small contrasts as much as possible. To see the relationship between the $\gamma$ values and the image histograms, look at the code in m10067_1.m and Matlab help for the function "hist" and create histograms for the images at the various $\gamma$ values in this m-file. Include them in your homework and comment on the results. (Note that an easy way to get a histogram of a 2-D matrix or image stored in the variable "data" is to use the command "hist(data(:))". This use of the colon takes a matrix and treats it as a column vector by simply stacking all the columns one after the other.)

(b) Now let's investigate the effect smoothing the picture. This could represent the effect of the point–spread function of a camera with a smaller aperture, for example. Convolving with the two–dimensional point spread function

$$kernel = \frac{1}{4}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix},$$

    will average four neighboring pixels. See the Matlab function *CONV2* to see how to do this. Try this point spread function and see what happens.

Let's try a larger point–spread function. Now, one way to make a point–spread function with a larger size (called the "support") is to repeatedly invoke;

$$kernel = conv2(kernel, kernel);.$$

Keep doing this until the width of the PSF is larger than 8 pixels. What effect does this have on the image?

Now try the following kernel;

$$kernel8 = ones(size(kernel));.$$

Discuss the differences in the images obtained with these two kernels that have the same size.

(c) Next, let's look at the issue of dynamic range. This image has eight bits of data, so values range from 0 to 255, in steps of 1. That is, there are 256 possible grey levels. Simulate fewer levels by dividing by a number $n$, and rounding to an integer. For example, if $n = 2$, we simulate a seven–bit image. I find it useful to then multiply by $n$, and display all the images with the same color axis. Another way is simply to take every odd-valued pixel (or even-valued pixel) and round it to the an adjacent even (odd) value, so that again you only have 1/2 as many possible values. You may well think of your own trick to accomplish the same goal. In your writeup, discuss how you accomplished this and your determination about how many bits are required to produce a reasonable image? What is lost at each step?

(d) The other problem of dynamic range is saturation. Suppose, starting from the original image, we set all the values greater than 128 to 128. See what happens. Then try instead, setting all the values less than 128 to 128. This produces two seven–bit images. How do these compare to the seven–bit image produced in the part above?

**Problem 3: Pupil and Image Functions**

Here we consider some of the issues raised in our class discussion of images and pupil functions. The idea is to work with a toy problem in one dimension, and to begin to understand Fourier Optics. Let's consider the following situation: We have a pupil which is defined by $x_1$ spanning a width of 500 micrometers, which we sample at 1 micrometer intervals. Suppose that the image field of view is defined by $x$, covering the same range, and that the image is 2 centimeters from the pupil. To get started, download the file m10472_1.m from the course website.

We have written the file to set up the arrays $x$ and $x_1$, and to do one simple calculation. Speicifically, assume that $U_1(x_1, 0)$ is non–zero only at the center point. Then the field $U(x, z)$, at the image can be described by a spherical wave,

$$\frac{e^{ikr}}{r},$$

where $r$ is the distance from the origin to the point $(x, z)$ in the image. This field is nearly constant over the image field of view. The program plots the real and imaginary parts of the field. Perhaps more interesting, it plots the square of the magnitude of the field, which is proportional to the irradiance.

Your tasks are to try different fields $U_1$ to see how the sum of all the spherical waves from different points $(x_1, 0)$ add up at different image locations, $(x, z)$.

(a) Now, suppose that we give $U_1$ a curvature toward the point $(0, z)$. Specifically what we want to do is to choose $U_1$ so that all the contributions at $(0, z)$ have the same phase. The point of doing this is that the fields will all add up at this point, giving $U(0, z)$ a large value. At other values of $x$, $U(x, z)$ will be a sum of contributions with different phases, and will thus be much smaller. In this way, we have an "image" of our point source, at $x = 0$.

The easiest way to do this is to make the field contributions all real at the point $(0, z)$, so that $U(0, z)$ will be real. What is the function $U_1$ required to do this? Now, starting from our program, make the correction, and produce the irradiance plots.

(b) Next take the field in the above part, and set it to zero for $|x_1| > 100$ micrometers. This corresponds to reducing the pupil's aperture to a diameter of 200 microns, from the original 500. Produce the irradiance plots and discuss them in comparison to those from the part above.

(c) What happens if we focus to the wrong distance? Go back to part a, and modify the code so that the spherical waves will all be real at $(0, 2z)$ instead of $(0, z)$. Plot and discuss your results.

(d) Finally write $U_1$ as the sum of two contributions, one focused at some position $(x_a, z)$ and the other at $(-x_a, z)$. Try different values of $x_a$, between a few micrometers and a few tens of micrometers. When does it become hard to tell that there are two points? If we had an extended object, hopefully you can begin to see how the image could be described by convolution with the point–spread function.