# Lab/Homework #3

Problem 1 - Raster displays. Start the matlab application on the computer by typing

matlab

at the prompt. Set the default color table to black and white by entering the command:

colormap(gray)

Download the text file 'raster' from the class web site—the file is found under the “Homework” tab in the Homework 1 section. Place the file into your leland disk area. Type out the file and note that it is a sequence of 0's and 1's.

This file is a string of data values that will form a recognizable image once it is displayed with the proper row width. Read this file into a matrix and display it on the screen using the matlab commands *imagesc* and *axis.* Information on each command can be found using the matlab *help* command. Save the output file and submit it using the submission script. Experiment with various lengths to see how the image is distorted if the line length is incorrect. In your writeup illustrate the correct and also one incorrect line length value.

Problem 2 - Resolution.

1. Create an array of zeroes 512 points on a side, and display it on axes that are evenly and equally spaced using the *axis* command as before.

1. Add circles of radius 16 and amplitude 1 to the array with centers at locations (200, 200), (232,200), (200,300), (248, 300), (200,400), and (264,400), and display. You will have to do some matlab programming to do this.

1. Image this array using an instrument with a resolution of 32 pixels by filtering the array with a box 32 x 32 pixels of value 1, using the matlab command *filter2*. Comment on how well the pairs of points are separated in the output image.

1. Plot cuts (lines) through the pairs of points for each of the three separations, and include in your writeup.

Problem 3 - Resolution in an image.

1. Download the binary image 'campusdrive' from the class web site. Read the file in matlab using the command *fread*. The data consist of unsigned characters 8 bits in size, and form an image with line length 580 pixels. The image is 435 lines long. Display the image.

1. Filter this image with boxes 2, 4, 8 and 16 pixels on a side and observe the results. What features do you lose at each resolution size ?

1. Plot cuts through one recognizable feature in the image for each resolution and examine the minimum obvious structure size in each case.

Problem 4 - Quantization.

1. Read in the original image from the previous problem.

1. The initial image consists of eight bits of data for each pixel. Create new images using 5, 4, 3, 2 and 1 bit only for each pixel. How many bits are needed to preserve image quality? Does it change from place to place in the image? Why so?

Problem 5 - An unknown image.

1. Download the byte file lab3prob5data from the class web site.

1. Determine the line length and display the image.

1. Determine the number of actual data bytes in each line, and the size of the header area.

1. Find any line count information in the header and determine if there are any missing lines.

1. Create a new image with no missing lines by duplication of preceding lines to fill in the data.

Hints and tips for first exercise

Here are some example solutions to the first two problems from Lab #3, with some hints on the matlab commands needed for the other problems. Use the help command to learn more about these commands.

Problem 1 - Raster displays.

Sample solution code:

f=fopen('raster');

[pix,count]=fscanf(f,'%d',[18,3]); pix=pix';

pix

imagesc(pix); axis('equal');

axis('image');

Problem 2 - Resolution.

Sample solution code:

pix=zeros(512,512);

circle=ones(33,33);

for i=1:33, for j=1:33,

if(sqrt((i-17)^2+(j-17)^2))> 16 circle(i,j)=0; end; end; end;

for i=1:33, for j=1:33,

pix(i+200,j+200)=circle(i,j); pix(i+232,j+200)=circle(i,j); pix(i+200,j+300)=circle(i,j); pix(i+248,j+300)=circle(i,j); pix(i+200,j+400)=circle(i,j); pix(i+264,j+400)=circle(i,j); end; end;

h=imagesc(pix'); axis ([0 512 0 512]);

axis image;

## The filtering can be done with lines like the following: ##

filter=ones(33,33);

pix2=filter2(filter,pix,'same');

h=imagesc(pix2'); axis ([0 512 0 512]);

axis image;

Problem 3 - Resolution in an image.

Hints:

You will need to read in a file now in binary format. That is, the data are in integer 8-bit format on the disk, rather than as text or ascii representations. The read command for raw data such as these is:

f=fopen('campusdrive.raw');

a=fread(f,[580 inf],'uint8');

where, once again, f is a file pointer and a contains the input array. “fread” replaces fscanf, and we do not use a format description such as %d. The [580 inf] parameter tells the program to place the data in an array of line length 580 and the inf argument says to read until it runs out of data. ‘uint8’ means the dat are stored on disk as unsigned integers of length 8 bits.

You can then filter and display as in the previous problem.

Problem 4 - Quantization.

Hints

In this problem you need to select only the high order bits in the pixels. We can do this with logical and’s using the matlab bitand command:

b=bitand(b,128+64);

which, for example, logically and’s every element of matrix b with a number consisting of 1’s in the 128 and 64 places of a binary number, thus saving the top two bits of an eight-bit number. Open the binary file as in the previous problem, and display as usual using imagesc and axis.

Problem 5 - An unknown image.

Hints: Try this one yourself!