Course: CISC 105: General Computer Science, sections Semester: Fall 2006 Professor: Chandra Kambhamettu Due: Tuesday 12/6/2006, by 1.00PM.

Programming Assignment 3 - Image Processing

1 Academic Dishonesty

Please be sure to re-read the syllabus on Academic Dishonesty. For all programming assignments, students must work INDEPENDENTLY.

2 Grading

80% of the grade depends on your program's correctness and 20% of the grade depends on your program's readability. Your program style should conform with the programs shown in the textbook. Use meaningful variable names and provide indentation to improve readability of your program.

3 Objectives

- Write an algorithm showing the list of steps to take for solving the given problem. In this process, you should identify the functions that are to be used in the program.
- Develop a C program based on the designed algorithm.
- Correct the program for any syntax errors.
- Test the program for any logic errors.

4 Assignment

Write a program to read an image into a 2D array of type unsigned char. Images are grayscale, in jpeg format, of size 256X256. Sample code for this will be provided.

Perform the following:

1) Create an original_image array and working_image array. The working_image array will have the current image processed image, ready to get displayed anytime. You will need 2 arrays each, of size 128X128, 256X256, 512X512, as the image size will be 256X256.

2) Invoke a function **show_instructions** which displays a welcome message and general instructions to the user.

3) Have the following image processing operations available to user, till **q** is typed to quit the program (see next page):

- a Median Filtering of the image
- b Mean Filtering of the image
- c Upsampling of the image (only till 512X512)
- d Downsampling of the image (only till 64X64)
- e Histogram of image. Create image of 512 columns, then for each alternate column, fill the column with frequency of occurrence of that particular intensity value in the image. The filling value can be anything that is visible, say, 50. Rows can be 512 this means, you will scale down the frequency value by a large number that you can determine.
- f Display histogram image (using xv program)
- g Threshold the image. i.e, pixel range such as, 50-80 can be selected and assigned a flat intensity value, say, 255. The resulting image will have all the pixels having 50-80 intensity values look very bright. This is usually done after looking at the histogram.
- h Display Image processed image (same as working_image array)
- i q (quit, but ask the user a file name to write out the final image. User don't need to type .jpeg, but you need to concatenate this extension using strcat)

You can use /usr/tmp/ directory to write intermediate results.

5 Extra Credit (10 points)

Add these extra operations: flip (left to right), invert image (upside down), pick a portion of image of size 256X256 (user defines starting position).

6 What to Hand In

The images will be provided to you. Run your program and make sure all the results are correct. Don't need to submit your results, but, submit a report which explains if all the operations work, or if there is any error that you like to point out. Submit this along with the original code.