CSC442/542 Image Processing – Spring 2017
Programming Assignment #1: Point Processing

For your first image processing assignment, write a program to read in an image, display it, and apply a variety of image enhancement techniques. In addition to the default File menu (Open, Save, Exit), offer the user the following menu options:

1. convert a (color) image to gray scale (use 30% red + 59% green + 11% blue)
2. negate an image
3. binary threshold an image (get threshold from user)
4. posterize an image (get number of levels from user)
5. increase/decrease image brightness (get amount from user)
6. increase/decrease image contrast with a linear ramp (get endpoints from user)
7. adjust image gamma with a power transformation (Equation 3.2-3) (get gamma from user)
8. compress the image dynamic range with a log transformation (Equation 3.2-2)
9. discrete 8-level pseudocolor (e.g., Figure 6.20)
10. continuous pseudocolor (e.g., Figure 6.22 and Problem 6.5)
11. automated contrast stretch (between min/max intensities)
12. modified contrast stretch (get endpoints from user, as percentages of dark and light pixels to ignore)
13. histogram equalization
14. histogram equalization with clipping (get clipping threshold from user, as percentage of total number of pixels)
15. Bit-plane slicing (get plane from user)
16. some other point process of your choosing
17. Help/About message

Apply the selected image processing action to the image window that currently has focus. Your routines should function correctly with color and grayscale images.

Notes

- Do not use any predefined LuaIP functions to perform image modification in your program. You must code the image processing routines yourself. (You may use routines such as RGB2YIQ/YIQ2RGB to convert between color models.)
- You may work in teams of two on this assignment.
- When you are finished writing, testing, and debugging your program, turn in your source code using the MCS website submit program. Splitting your source code into multiple files for separate compilation is highly recommended (i.e., do not submit one monolithic source file). Use zip (or tar) to package multiple files for submission.
- The MCS website submit program is accessed via the MCS Department Web page (http://www.mcs.sdsmt.edu), by selecting the list item on the left entitled "Submit it!". Usage is self-explanatory: enter your name, choose the instructor and click "Select Instructor", choose the appropriate course (CSC442 or CSC542), type in (or browse to) the filename you

wish to submit, and click “Upload”. Submissions will be date- and time-stamped. You must submit your program by the due date (**Tuesday February 7**) in order to receive credit for this assignment. Late programs will not be accepted for partial credit unless prior arrangements have been made with the instructor. If you have any problems with the submit program, report them to your instructor and submit your program by email instead.

- To receive full credit, your code must be readable, modular, and well-documented, as well as correct. Try to make it reasonably efficient in terms of both execution time and space utilization. Your program must compile successfully using Lua and ZeroBrane Studio, on both Linux and Windows. If your program does not run correctly, indicate why. This will make it easier to give you partial credit.
- You may use sample code given in class or found on the course website as a basis for your program, but be sure to give credit for any code that you did not write. (As a general rule, *always* credit the author of any code that you use.)