Image Processing
Spring 2015
Prof. George Wolberg
Homework 1

Due: Wednesday, February 25
Objective: This assignment requires you to implement various point operations on digital images. See the sample problem (in hw1.smpl) and solution (thr.c) to see what is expected of your programs in terms of style and comments.

1) **qntz** *in n out*

Function *qntz* reads the input image from file *in* and uniformly quantizes it into n quantization levels. The output is stored in file *out*. Quantization should be done by subdividing the range of 256 intensities into n uniform intervals. Assign the mid-points of the uniform intervals as the output intensities. For instance, if $n = 4$, then gray values 32, 96, 160, and 224 are used. In general, the intermediate gray values are increments of $256 / n$ with a bias of $128 / n$.

2) **histo_plot** *in out flg*

Function *histo_plot* evalutes the histogram of input image *in* and plots it into an output image *out* of dimensions 256×256. Use black (0) for the background and white (255) for the foreground. The x-axis represents intensity and the y-axis denotes pixel frequency. To denote an entry of n for pixel value v, for instance, set n pixels in column v to white (from bottom-to-top). If $flg = 0$, scale the histogram pixel frequency before plotting so that $Havg = (width * height)/256$ appears at $y = 128$. Clip at $y = 255$. If $flg = 1$, then scale the histogram pixel frequency so that the highest frequency value appears at $y = 255$ (no clipping). What is the advantage/disadvantage of each method?

3) **histo_stretch** *in t1 t2 out*

Function *histo_stretch* stretches the dynamic range of the input image in file *in* to fill the entire [0,255] range. The range that is stretched spans from $t1$ to $t2$. If $t1$ ($t2$) is negative, then the first (last) nonzero histogram value is used to delimit the start (end) of the span. All intensity values below $t1$ or above $t2$ are pulled to 0 or 255, respectively. The output is stored in file *out*. Note: you may want to look at the image histogram before selecting an appropriate $t1$ and $t2$.

4) **histo_match** *in n out*

Function *histo_match* performs a histogram matching operation to input image *in*, saving the result in file *out*. The histogram used is determined by n: 0 for a flat histogram and a positive (negative) number for an exponentially increasing (decreasing) histogram. The absolute value of the number is to be interpreted as the exponent of the exponential histogram. For instance, if the user-specified value is positive, the histogram is given by v^n, where $0 \le v \le 1$ denotes the intensity range (scaled in the same manner as in gamma correction). Otherwise, the histogram is given by $1 - v^{|n|}$. Note that $n = 0$ is used to perform histogram equalization. You will have to make sure that the histogram entries are scaled properly so that their sum is equal to the total number of pixels in the input image. Modify the code supplied in the notes so that the histogram is matched *exactly*, with the possible exception of the very last histogram entry.

5) **threshold_Otsu** *in out*

Function *threshold_Otsu* performs Othsu's adaptive thresholding method to compute a superior thresholded image. The input image is stored in file *in* and the output is saved in *out*. The algorithm uses the input image's histogram to compute an optimal threshold based on minimizing the within class variance (i.e., maximizing the between class variance). Please refer to the original paper by Nobuyuki Otsu entitled "A Threshold Selection Method from Gray-Level Histograms" in IEEE Trans., Sys., Man., Cybernetics, vol. 9, no. 1, pp. 62-66, 1979. Print out the optimal threshold value calculated in *threshold_Otsu*.