**CSC 545/645 Computer Speech, Music and Images**

**Exercise No. 2b, Week 6, due February 21, 2021**

**Display image histograms**

**Goals**

Display the red, green, and blue histograms for an image

**Procedure**

In the first part of this exercise you wrote a program to calculate and print red, green, and blue histograms for an image. In this exercise you will extend that program to display those histograms.

Start with your solution to Ex02a\_histograms; save it as Ex02b\_histogramDisplay. Add global variables for a PFont (PFont f), text x and y positions (int textX = 50, textY = 75), a Boolean variable to determine whether to display an image or a histogram (boolean showHists = false), and a String filename (“charlton vale 3.jpg” is a good image to start with). Also add the following global variables: int posR = 10, posG = 275, posB = 541; these will determine the horizontal start positions on the canvas for the red, green, and blue histograms (rCounts, gCounts, and bCounts).

You should already have a function, calcHists(PImage img) that calculates the histograms (rCounts, gCounts, and bCounts) for the given image; an alternative design would be for calcHists() to calculate the histograms for currentImg, in which case it would not take an argument.

A second function, drawHists(), will display rCounts, gCounts, and bCounts. drawHists() will also display the pixel value (0 to 255) that the mouse is over and its count in the upper left corner of the canvas. This is slightly tricky: if mouseX is between posR and posR + rCount.length, then the pixel value is mouseX – posR; the count is rCounts[mouseX – posR]. The logic is similar for gCounts and bCounts. You can use printHists() to see whether your mouse-over display is correct.

In draw(), if showHists is true, you will call drawHists(); otherwise, you will display currentImg.

A key event handler (keyPressed() or keyReleased()) will determine the display. You probably want to set the background to black immediately on entry to the key handler. If the key is ‘1’, ‘2’, or ‘3’, then you will set showHists to false, set currentImg to img, bimg, or dimg, respectively, and set the canvas size to the size of currentImg. If the key is ‘h’, ‘b’, or ‘d’, then you will call calcHists() on img, bimg, or dimg, set showHists to true, and set the canvas width to posB + the length of bCounts. You can choose a reasonable height – I used the height of currentImg.

You should be able to see the effect of brightening or darkening the image by comparing the histograms of img, bimg, and dimg. Try some of the underexposed or low contrast images – can you tell whether the image is underexposed or low-contrast by looking at its histogram?

**Deliverables**

Submit your .pde file on Blackboard by midnight on the due date.