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

**Exercise No. 1, Week 4**

**Find Colors in an Image**

**Goals**

1. Create an array of images
2. Calculate color distances
3. Iterate over pixels
4. Add images

**Procedure**

Write a Processing program to extract white and yellow lines from an image. Start with *Ex01\_findLines.pde* and use the image *emptyroad.jpg.* Extract the white lines by setting any pixel that is not “close” to white to black; “close” is defined by a threshold which is provided in the skeleton code. You’ll have to write code to calculate the distance between two colors (the threshold assumes the Euclidean distance: sqrt((r1-r2)2 + (g1-g2)2 + (b1-b2)2). Also extract the yellow pixels in the same manner; the reference yellow for comparison is provided in the skeleton code. Allow the user to select the image to display from the keyboard: ‘0’ selects the original image, ‘1’ selects the white line image, ‘2’ selects the yellow line image, and ‘3’ selects the white and yellow image added together. The skeleton code assumes the images are stored in an array (img[]); a global variable, imgIndex determines which image is displayed in the draw() loop.

There are several other images you can experiment with in the Ex01\_findLines data folder but you’ll have to find good reference colors and thresholds. You’ll only need white for the Daytona images but *charlton vale 3.jpg* has white, red, green, blue, and maybe even a bit of yellow if you can find it. Extend your program to print the pixel value at the mouseX, mouseY position on mouse click. (NOTE: mouseX and mouseY refer to positions in the display window. If the display window is larger than the image, or if the image is not displayed at position 0, 0, then you can get array out of bounds errors. To avoid this, display the display window’s pixel value, rather than the image’s pixel value (use the get() function rather than the PImage.get() method).

Processing commands you will probably use (there may be others, too); use Processing.org/reference:

color() //creates a color object

createImage() //creates a blank image

get() and set() //gets or sets a pixel in the display window

image() //displays an image in the display window

PImage.get() //gets a pixel (a color object) from an image

PImage.set() //sets an image pixel to a color

loadImage() //loads an image from a file

size() //Sets the size of the display window

**Deliverables**

Submit ONLY your exercise pde file on Blackboard