CS/ECE 545 Digital Image Processing
Homework 5, Spring 2014 (Due April 23, 2014 by 6PM)

Note: Some of the problems below include questions for you to answer. Put your answers in a separate Word or PDF document and submit it in a zip file that also includes your code.

This homework will be handed out in 2 parts. Part 1 will be handed out on 4/9/2014 and part 2 will be handed out on 4/16/2014. **BOTH PARTS WILL BE DUE ON THE DUE DATE OF 4/23/2014**

Part 1: Programming

1. **Burger & Burge Exercise 11.5 (page 236) (25 points):** Implement a Java class for describing a binary image region using chain codes. It is up to you, whether you want to use an absolute or differential chain code. The implementation should be able to encode closed contours as chain codes and also reconstruct the contours given a chain code. Call your Java class **`Chain_codes`**

2. **Burger & Burge Exercise 11.12 (page 237) (25 points):** Write an ImageJ plugin that (a) finds (labels) all regions in a binary image (b) computes the orientation and eccentricity for each region, and (c) shows the results as a direction vector and the equivalent ellipse on top of each region (as exemplified in Fig 11.19. Hint: Use Eqn. (11.33) to develop a method for drawing ellipses at arbitrary orientations (not available in ImageJ). Call your ImageJ plugin **`Region_labeling.java`**

Submit ImageJ plugin (`Region_labeling.java`)

Part 1: Written Part

3. **(6 points)** Show all your work. By hand convert the following RGB value = (0, 184, 160) to HSV
4. **(6 points)** how all your work. By hand, convert the following HSV values = (0.67, 0.7, 0.7) to RGB
5. **(6 points)** Show all your work. By hand, convert the following RGB value = (124, 220, 0) to YIQ
6. **(6 points)** Show all your work. By hand, convert the following YIQ value = (1, 0.3, 0.3) to RGB

Part 2: Discrete Fourier Transform!

7. **(26 Points) Burger & Burge Exercise 14.1 (page 366):** Implement the two-dimensional DFT using the one-dimensional DFT, as described in Sec. 14.1.2. Apply the 2D DFT to real intensity images of arbitrary size and display the results (by converting to ImageJ **`float`** images). Implement the inverse transform and verify that the back-transformed result is identical to the original image

Important Note: Some of the equations from the book contains errors especially the color Image chapter. Before using any equations, please check the book's errata and use the correct form of each equation. The book's errata is at:
http://imagingbook.files.wordpress.com/2013/06/burgerburge-en1-errata.pdf

Submitting Your Work

Submit (`Chain_codes, Region_labeling`) and your README file in Word or PDF with all your **answers clearly typed up!!!**

Put everything into ONE zip file named *yourfirstname_yourlastname_hw5.zip* using turnin.

DON'T EMAIL ME YOUR HOMEWORK. ALSO, TEST YOUR CODE IN THE ZOOLAB BEFORE SUBMITTING!!