

COMS W3101-2 – Spring 2010

Homework 4

The main goal of the first homework is to have you practice with digital images in MATLAB

For this assignment you should write a principal m-script file called `hw4.m`. At the top of the file you should put your name and ID in a comment. There are 3 exercises in this assignment, for a total of 100 points, plus 20 extra credit points if you do Exercise 3.3. For each exercise you will have to write a single or series of MATLAB commands. Make sure you place a comment before the code solving each of the exercises, in order to specify which part of code solves what exercise and to write the answers to the exercise.

This homework is due on April 6, at beginning of class, no exceptions. Put your code (.m files) and any additional files in a single folder, name it `youruni_hw_4` and zip it. Upload the zipped folder to [CourseWorks](#). Also, bring a printout of your code to class.

Exercises:

You have been asked by a group of veterans to analyze the picture of the flag taken during the World War II's battle of Iwo Jima.

1. **(35 points)** Write a MATLAB script that
 - 1.1. Reads the image `flag.jpg`
 - 1.2. Resizes the image to 256×256 pixels using bilinear interpolation
 - 1.3. Tiles the image to form 4 quadrants, where
 - The top left quadrant is the original image
 - The top right is the red channel of the original image
 - The bottom left is the green channel
 - The bottom right is the blue channel

Your code should produce an image like the one in Figure 1. Do NOT use `subplot()`.

2. **(35 points)** The veterans think that there should be another star on the flag, so symbolize their friends who lost their lives to raise it during the battle. Write a MATLAB function called `'drawStar()'` that:
 - 2.1. Takes an image name as input
 - 2.2. Checks if the input name is correct. The definition of correct is that the suffix of the file name should be `.jpg`, `.png` or `.bmp`. If the input name is not correct, the function should display an error message and close

- 2.3. Displays the grayscale version of the image (NOTE: the input image could be already grayscale, but if it is not, the program should convert it from RGB to grayscale)
 - 2.4. Memorizes the coordinates of 5 points clicked by the user on the displayed image
 - 2.5. Assuming that the user clicks the points in order following an anticlockwise circular trajectory (as shown in Figure 2a), the function draws green lines over the image, connecting the points clicked by the user, so that the final drawing is a star (see the reference example in Figure 2b)
- HINT: look into the keyword 'return' and the function 'line()'



Figure 1 – flag example

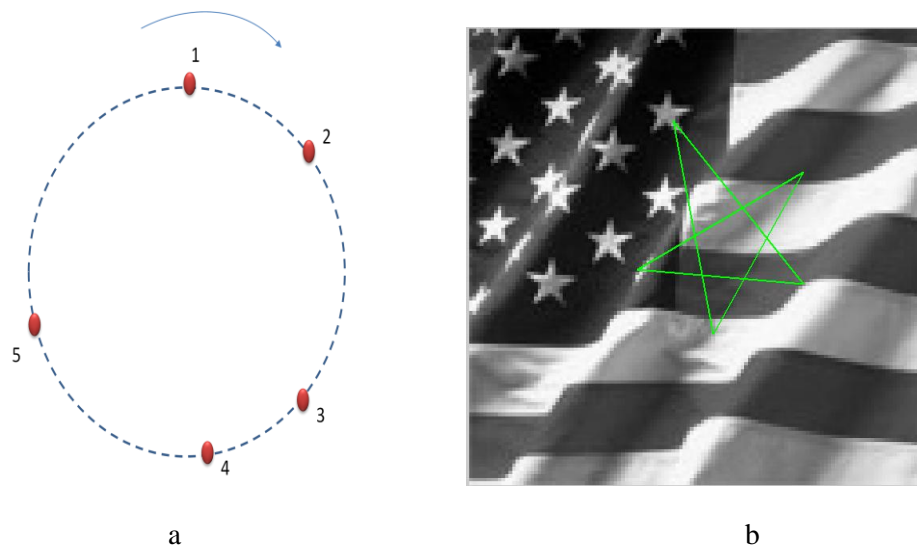


Figure 2 : a – Order of insertion of points by user. b – Example of drawn star

3. (30 + 20 points) Write a function 'histograms()' that takes a color image as input. The function then

3.1. Computes the histogram of the image for each of the R, G and B channels separately

3.2. Shows in a figure 4 images:

- The original image
- The histogram of the red channel with a red colorbar
- The histogram of the green channel with a green colorbar
- The histogram of the blue channel with a blue colorbar

3.3. (for extra credit) Shows in a figure 4 images:

- The original image
- The histogram of the red channel
- The histogram of the green channel
- The histogram of the blue channel

as shown in Figure 3. Note that for point 3.3 each histogram is an image, not a plot! For each channel, you have to create a white $[n \times 256]$ image, with n of your choice, and then fill the pixels with the color of the channel, one column (of 256) at a time, according to the histogram values.

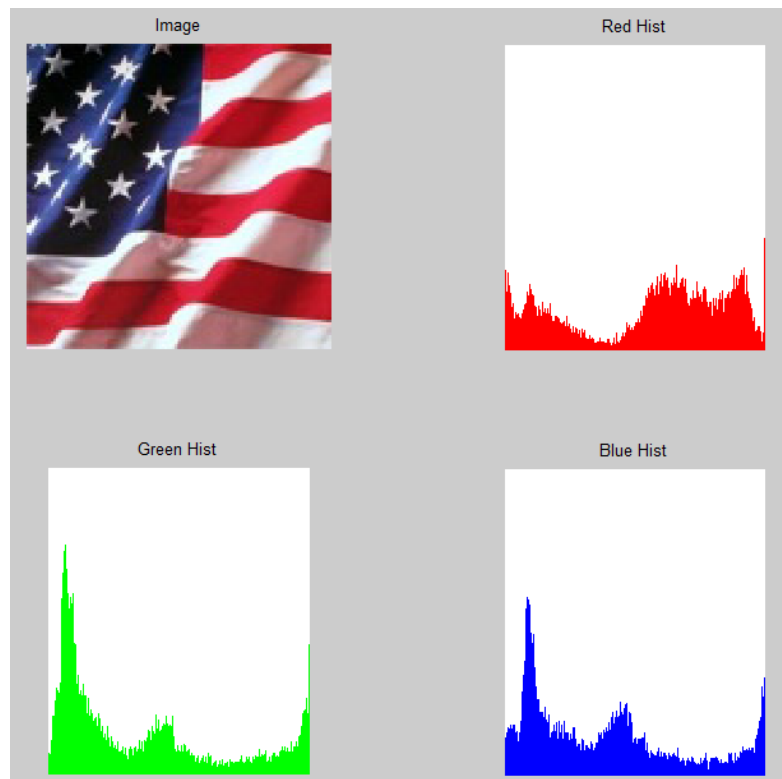


Figure 3