Homework 3

ECE 172A
Introduction to Intelligent Systems

February 17, 2017

Make sure you follow these instructions carefully during submission :

- Homework 3 is due by 11:59 PM, February 27, 2017.
- Submit your homework via gradescope and also by email to arangesh@ucsd.edu with the subject line ECE 172A HW3. The email should have one file attached.

  Name this file: **ECE_172A_hw3_lastname_studentid.zip**.

  The contents of the file should be:

  1. A pdf file with your write-up. This should include your code outputs along with your answer to each question. Include your MATLAB code (only the ones you made changes to) in an Appendix at the end of your write-up.

     Name this file: **ECE_172A_hw3_lastname_studentid.pdf**.
  2. All of your MATLAB code in a folder called **code** (Note that you have to include your code in the write-up in addition to this). This should include all files necessary to run your code out of the box.
- All problems are to be solved using MATLAB unless mentioned otherwise.
- You should avoid using loops in your MATLAB code unless you are explicitly permitted to do so.
- Finally, carefully read and include the following sentences at the top of your report:

  Academic Integrity Policy: Integrity of scholarship is essential for an academic community. The University expects that both faculty and students will honor this principle and in so doing protect the validity of University intellectual work. For students, this means that all academic work will be done by the individual to whom it is assigned, without unauthorized aid of any kind.

  By including this in my report, I agree to abide by the Academic Integrity Policy mentioned above.

Problem 1. Sampling & Quantization (10 points)

In this problem, we intend to study the effects of *sampling* and *quantization* on digital images. Your job is to write a function with the following specifications (you may use loops if necessary):

(i) The function takes one input: the image file name.

(ii) The input image is assumed to be *grayscale*.

(iii) Sample the image in spatial domain with a sampling rate of 10 (your image should be approximately 10 times smaller along width and height, do not use *any* MATLAB functions).

(iv) Do a 5-level uniform quantization of the sampled image so that the bins cover the whole range of grayscale values (0 to 255). You should not use any MATLAB functions for this.

(v) The function returns one output: the sampled and quantized image.

Evaluate your function on the image *peppers.png*.
Things to turn in:

- The original image, the image after sampling, and the image after *both* sampling and quantization.
- Response for the following question : How would the above function be of use in image compression?
- MATLAB Code

Problem 2. Hough Transform (20 points)

(The first two parts of this problem is borrowed from Professor Belongie's past CSE 166 class.)

(i) Implement the Hough Transform (HT) using the (ρ, θ) parameterization as described in GW Third Edition p. 733-738 (see 'HoughTransform.pdf' provided in the data folder). Use accumulator cells with a resolution of 1 degree in θ and 1 pixel in ρ.

(ii) Produce a simple 11×11 test image made up of zeros with 5 ones in it, arranged like the 5 points in GW Third Edition Figure 10.33(a). Compute and display its HT; the result should look like GW Third Edition Figure 10.33(b). Threshold the HT by looking for any (ρ, θ) cells that contains more than 2 votes then plot the corresponding lines in (x,y)-space on top of the original image.

(iii) Load in the image 'lane.png'. Compute and display its edges using the Sobel operator with default threshold settings, i.e.,

```
E = edge(I,'sobel')
```

Now compute and display the HT of the binary edge image E. As before, threshold the HT and plot the corresponding lines atop the original image; this time, use a threshold of 75% maximum accumulator count over the entire HT, i.e. `0.75*max(HT(:))`.

(iv) We would like to only show line detections in the driver's lane and ignore any other line detections such as the lines resulting from the neighboring lane closest to the bus, light pole, and sidewalks. Using the thresholded HT from the 'lanes.png' image in the previous part, show only the lines corresponding to the line detections from the driver's lane by thresholding the HT again using a specified range of θ this time. What are the approximate θ values for the two lines in the driver's lane?

Things to turn in:

- HT images should have colorbars next to them
- Line overlays should be clearly visible (adjust line width if needed)
- HT image axes should be properly labeled with name and values (see Figure 10.33(b) for example)
- 3 images from 2(ii): original image, HT, original image with lines
- 4 images from 2(iii): original image, binary edge image, HT, original image with lines
- 1 image from 2(iv): original image with lines
- θ values from 2(iv)
- Code for 2(i), 2(ii), 2(iii), 2(iv)