

# Problem Set 3 CMSC 426

## Due: Thursday, April 3

**Overview:** This problem set will work a little different. There will be a standard part, and an advanced part. Students may choose to do either part. Alternately, they may choose to complete the standard part, and hand in any portions of the advanced part for extra credit. Not that the standard part totals 100 points, while the advanced part totals 120. A student may receive no more than 140 points total. Grad students may feel that they should focus on the advanced part.

Remember, everything you do should be turned in as a hardcopy. All code should be emailed to the TA. Write the date that you hand in assignments on the hardcopy, especially if you hand them in late. If you leave an assignment under my door on in my mailbox, always include the date.

### Standard Assignment

- 1) Programming RANSAC. **70 points total.**
  - a. Write a function of the form:  $[a, b, c] = \text{gen\_line}(pt1, pt2)$ . This function should take two points as input, and return the parameters of a line that fit these points. The line should be  $ax + by + c = 0$ , where  $\sqrt{a^2 + b^2} = 1$ . Test this function using the pairs of points: i) (1,1) (2,1) ii) (1,1) (1,2); iii) (1,1) (3,2). Turn in your code, and show calls to the routine and responses for these inputs. **5 points**
  - b. Write a function of the form:  $d = \text{line\_dist}(L, p)$ .  $L$  represents a line (perhaps as a vector of  $[a,b,c]$ ).  $p$  represents a point.  $d$  is the distance from the point to the line. Test your code with i) The line formed by the points: (0,1) (2,1) and the point (5,2). ii) The line formed by the points (1,1) and (3,2) and the point (7,4). Turn in your code and show calls to the routine and responses for these inputs. **5 points**
  - c. Compute the correct answers to the problems in (b) by hand. Show your work and the results. **5 points**
  - d. Write a function of the form:  $k = \text{num\_samples}(n, m, p)$ . If there are  $n$  points in an image, and  $m$  of them come from a single line, then if we randomly sample  $k$  pairs of points we should have a probability greater than  $p$  of picking a pair of points that come from the line.  $k$  should be the smallest integer for which this is true. Test this with  $n = 20$ ,  $m = 8$ ,  $p = .9$ . Turn in your code, and show calls to the routine and responses for these inputs. **10 points**
  - e. Write a function of the form:  $\text{pts\_cor} = \text{verify}(p1, p2, \text{pts}, \text{eps})$ .  $p1$  and  $p2$  are points that will be used to form a line.  $\text{pts}$  is a list of different pts.  $\text{pts\_cor}$  should contain a list of all points in  $\text{pts}$  whose distance to the line formed by  $p1$  and  $p2$  is less than  $\text{eps}$ . Test this routine with  $p1 = (1,1)$ ,  $p2 = (2,2)$ ,  $\text{pts} = [(3,4), (5,7), (3,8)]$ ,  $\text{eps} = 2$ . Turn in your code, and show calls to the routine and responses for these inputs. **10 points**

- f. Write a function to generate test data, of the form: `pts = test(n)`. This should generate points in the square  $[0,50], [0, 50]$ . The first 8 points should come from the horizontal line  $y = 25$ , with unit variance Gaussian noise added to them (use `randn`).  $n$  noise points should come from a uniform random distribution across the whole image (use the function `rand`). Test this by running it with  $n = 30$  and plotting the results. Plot points from the line in one color, and noise points in a different color. Turn in the plot and code. **10 points.**
- g. Write the function `[L, pts, c] = ransac(n)`. Generate  $n + 8$  points using `test`, from 1f. Sample enough random pairs of points so there is probability greater than .95 that two will come from the line, using `num_samples` from 1d. For each pair of points, use `verify` from 1e) to find all points that are near the line, using `eps = 2`. Select the line that is close to the largest number of points. Return this line (L), the list of points (pts), and the number of these that actually came from the line (c, ie, the number that were among the first 8 returned by `test`). Run your code for  $n = [0, 10, 20, 50, 100, 200]$ . Plot c versus n for these results. Turn in your code and the plot. **25 points.**

2) Pencil and Paper Exercises for EM **15 points total**

- a) Consider two Gaussian distributions with means of 80 and 160, and standard deviations of 40 for each. Compute the weights that the Expectation Step of EM will compute for data with the values: 80, 90, 100, 110. **5 points**
- b) Consider points at 80 90 100 110 120 130 140 with weights of: 0.78 0.61 0.40 0.21 0.10 0.04 0.02 (these are  $w_1$ , the weights relative to the first distribution.  $w_2 = 1-w_1$ ). Compute the new weighted means of the two Gaussian distributions. (Part of Maximization step). **5 points.**
- c) Compute the weighted standard deviation of the Gaussian distributions, assuming they have the same standard deviation. **5 points.**

Hints on 2. If unsure about these steps, consult the Matlab code from the lecture, or a standard text on probability, such as the one by Drake, or the Yair Weiss handout.

3) Pencil and Paper Exercises for Hough Transform **15 points**

Recall (Forsyth and Ponce p. 330) that we can write the equation for a line as:  $x\cos(\theta) + y\sin(\theta) + r = 0$  with  $r \geq 0$ ,  $0 \leq \theta < 2\pi$ . Draw a 10 x 10 grid, which divides  $\theta$  into 10 equally spaced ranges, and divides  $r$  into equally spaced ranges from 0 to 10. That is, the square in the lower left of your grid represents  $\theta$  between 0 and  $(2\pi/10)$ , and  $r$  between 0 and 1.

- a) In red, lightly shade in all squares that contain the parameters of lines passing through the origin (0,0). **5 points**
- b) In blue, lightly shade in all squares that contain the parameters of lines passing through the point (4,4). **5 points**
- c) In green, lightly shade in all squares that contain the parameters of points passing through the origin (6,7). **5 points**

### Advanced Assignment

1. E-M Programming Assignment. Check Yair Weiss' tutorial. <http://www.cs.huji.ac.il/~yweiss/emTutorial.pdf>, Forsyth and Ponce, and the class Matlab code for tips. **75 points total**

a) Write a function that fits a line to data. It should get as input vectors  $x, y$  and return  $a, b$  the parameters of the best fitting lines. Test your function with

(i)  $x=0:0.05:1; y=2*x+1$

(ii)  $x=0:0.05:1; y=2*x+1+0.1*randn(size(x))$ .

(iii)  $x=0:0.05:1; y=(abs(x-0.5) < 0.25).*(x+1)+(abs(x-0.5) >=0.25).*(-x)$ ;

In all cases plot the data and the best fitting lines. **15 points**

b) Write a function that estimates the parameters of two lines using EM. It should get as input vectors  $x, y$  and return  $(a_1, b_1, c_1), (a_2, b_2, c_2)$  the parameters of the two lines as well as the weight vectors  $w_1$  and  $w_2$ . (Set the free parameter  $\sigma^2=0.1$ .)

(i) test your function on the data in part (iii) of the previous question:  $x=0:0.05:1; y=(abs(x-0.5) < 0.25).*(x+1)+(abs(x-0.5) >=0.25).*(-x)$ ; Plot the data and the two fitted lines as estimated after each of the first five iterations. Also, show in separate plots the membership vectors after every iteration. **40 points**

(ii) experiment with adding Gaussian noise to the  $y$  coordinates. How much noise can you add before the algorithm breaks. **20 points**

c) Challenge problem (optional): Implement a method for choosing the number of lines E-M uses to fit to the data. Test this by generating random points selected from lines, varying the number of lines. How many different lines can you successfully separate using E-M? **10 points**.

2. RANSAC (Pencil and Paper).

We will consider a  $50 \times 50$  image,  $I$ .  $L$  is a horizontal line at  $y = 25$ . Suppose there are  $k$  points on  $L$ ,  $p_1, \dots, p_k$ .  $p_1$  and  $p_2$  have no noise,  $p_3 \dots p_k$  are perturbed by circularly symmetric Gaussian noise with a sigma of 1. Throughout we will assume that RANSAC matches to a line all points with a distance of less than 2 (eg.,  $2 \sigma$ ) to the line. **50 points total**

- a) We use  $p_1$  and  $p_2$  to hypothesize a line (which is identical to  $L$  in this case, since  $p_1$  and  $p_2$  are noise-free). Find the expected number of points in  $p_3 \dots p_k$  that will lie within 2 of  $L$ . **5 points**
- b) Consider the incorrect, vertical line  $V$  at  $x = 25$ . Suppose there are  $n$  noise points drawn from a uniform distribution in the image. Compute the probability distribution on the number of points that will lie within 2 of  $V$ . Hint: it may become useful to approximate this distribution with a Gaussian. **10 points**
- c) Compute the probability that the number of incorrect noise points that lie within 2 of  $V$  will be greater than the expected number of correct points from  $p_1 \dots p_k$  that lie within 2 of  $L$ . The answer will be a function of  $k$  and  $n$ . **5 points**
- d) Now, put all this together to estimate the probability that RANSAC will produce the right answer. Suppose that there are  $k$  points from a line, and  $n$  noise points from a uniform distribution. For  $k = 10$ , and  $n = 10, 30, 50$  compute the number of random pairs of points we must sample to ensure at least a 95% probability that one pair comes from the line. Use the simplified assumptions of (a) to estimate the expected number of points that will be matched to that line. Then, simplify by assuming that all incorrect lines match noise points with the distribution you computed in (c). What is the probability that one of the incorrect lines will match more points than the expected number of points that the correct line will match? **25 points.**

3) Optional Challenge problem. Note, I think this problem is pretty hard. I'm not sure I see how to solve it. Feel free to try to solve a simpler problem of your choosing for partial credit. **20 points.**

Consider three collinear points,  $p_1 = (0,0)$ ,  $p_2=(10,0)$ ,  $p_3=(20,0)$ . Suppose each point is perturbed by independent, identically distributed 2D Gaussian noise with sigma 1. Call the noise vectors  $n_1, n_2, n_3$ , so that the perturbed points are:  $q_1 = p_1+n_1$ ;  $q_2 = p_2+n_2$ ;  $q_3 = p_3+n_3$ . Let  $L$  be the line passing through  $q_1$  and  $q_2$ . Let  $d$  be the distance from  $L$  to  $q_3$ . Compute the probability distribution of  $d$ .

This document was created with Win2PDF available at <http://www.daneprairie.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.