

Revised Due Date: 11:59pm Monday Oct 1st, 2018.

General Guidelines:

For this assignment you will be submit via Moodle a carefully formatted assignment report as well as 3 MATLAB .m functions, as described below. Your report must be submitted as an Adobe Acrobat (.pdf) file. Your report should be well organized and address all questions below. (There is one sub-question that is a bonus question for 4422 but required for 5323 - see below.)

Points will be awarded in part for the clarity and formatting of your report. All plots must be clearly labelled and follow standard scientific conventions. Ensure that x and y axes are clearly labelled and units are specified. Please ensure that all plots are imported as *vector graphics*, not raster images.

1. Fitting a point to lines and a line to points.

(4422: 20 marks + 10 bonus marks, 5323: 30 marks)

In Lecture 2.1, we proved that in the normalized form of the 2D line $\mathbf{l} = (\hat{\mathbf{n}}, -d)$, $\hat{\mathbf{n}}$ is the unit normal to the line directed toward the line from the origin, and d is the distance of the line from the origin (Fig. 1). (Note correction from textbook, which incorrectly uses the representation $\mathbf{l} = (\hat{\mathbf{n}}, d)$.)

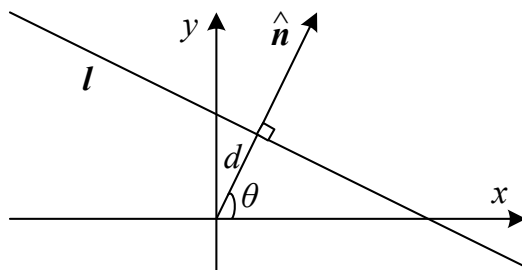


Figure 1: Normalized form of 2D line.

- (a) (10 marks) Modify Fig. 1 and combine with a mathematical argument to prove that the distance of a point $\tilde{\mathbf{x}}$ from a line $\tilde{\mathbf{l}}$ is given by $\tilde{\mathbf{x}} \cdot \mathbf{l}$.
- (b) (10 marks - bonus for 4422). Now suppose you have multiple lines and wish to find the point $\tilde{\mathbf{x}}$ minimizing the sum D of squared distances to each line:

$$D = \sum_i (\tilde{\mathbf{x}} \cdot \mathbf{l}_i)^2. \quad (1)$$

The solution is:

$$\mathbf{x} = \left(\sum_i \hat{\mathbf{n}}_i \hat{\mathbf{n}}_i^\top \right)^{-1} \sum_i d_i \hat{\mathbf{n}}_i. \quad (2)$$

Prove Eqn. 2. Hint: Write the dot product as $\bar{\mathbf{x}}^\top \mathbf{l}_i$ and turn the squared quantity into a quadratic form, $\bar{\mathbf{x}}^\top \mathbf{A} \bar{\mathbf{x}}$.

- (c) (10 marks) Using Eqn. 2, write a short MATLAB function

```
x = LSPointLines( $\tilde{\mathbf{L}}$ )
```

that accepts as input a matrix $\tilde{\mathbf{L}} = [\tilde{\mathbf{l}}_1, \dots, \tilde{\mathbf{l}}_n]$ containing n lines in \mathcal{P}^2 (homogenous format) and returns the point $\mathbf{x} \in \mathcal{R}^2$ (Euclidean, i.e., inhomogenous format) minimizing the sum of squared distances to each line.

Some useful MATLAB functions: sqrt, sum, repmat, zeros.

2. Sensor noise. (40 marks)

This question assesses two methods for estimating the sensor noise in your camera. Select a scene that has as little structure as possible (e.g., a blank wall). Place the camera on a tripod or prop it up on a stable surface and put the camera in timer (burst) mode to take a series of frames in rapid succession without touching or moving the camera between frames. Repeat the experiment under two different lighting conditions (lights on and lights off). Save the images in the highest quality format, preferably uncompressed TIFF.

- (a) **Method 1.** (18 marks)

Identify a region of one of the images that appears to be of constant brightness and then measure the variation in luminance recorded by the camera in that region.

- i. Specifically, write a MATLAB function

```
noise = noiseAnalCrop(im,top,bottom,left,right)
```

that accepts an RGB image `im` and the bounds of the rectangular region (in pixels) that you have identified and returns an estimate of the normalized sensor noise (see below). **Your function should first average the RGB values at each pixel to produce a luminance image.** Repeat the experiment for images taken under high and low illumination conditions, and answer the following questions for both experiments.

Some useful MATLAB functions: figure, imagesc, title, set, xlabel, ylabel, axis, mean, std, double, histogram.

- ii. Show the cropped image. How constant does the luminance appear to be?

- iii. Plot a histogram of the luminance values.
 - iv. How would you describe the distribution?
 - v. As a measure of noise level, report the standard deviation of this luminance image, normalized (divided) by the mean luminance.
 - vi. How does the normalized noise level seem to depend upon the amount of light in the room?
 - vii. What explanations could you give for this dependence or lack of dependence?
 - viii. Identify any potential flaws with this method for estimating sensor noise.
- (b) **Method 2.** (20 marks)
- Select two images taken back-to-back and difference the luminance images.
- i. Specifically, write a MATLAB function

```
noise = noiseAnalDiff(im1,im2)
```

that accepts two successive RGB images im1 and im2 and returns an estimate of the normalized sensor noise. Your function should first average the RGB values at each pixel before differencing the two images. Repeat the experiment for images taken under high and low illumination conditions, and answer the following questions for both experiments.

Some useful MATLAB functions: figure, imagesc, title, set, xlabel, ylabel, axis, mean, std, double, histogram.
 - ii. Show the luminance difference image. Do you think there was any movement in the camera or the scene?
 - iii. Plot a histogram of the luminance difference values.
 - iv. How would you describe the distribution?
 - v. How does this histogram compare to the histograms determined using Method 1?
 - vi. Under what assumptions would you expect them to be similar?
 - vii. As a measure of noise level, report the standard deviation of this luminance image, normalized (divided) by the mean luminance.
 - viii. How does the normalized noise level seem to depend upon the amount of light in the room?

- ix. What explanations could you give for this dependence or lack of dependence?
- x. Identify any potential flaws with this method for estimating sensor noise.
- (c) (2 marks) Which of these two methods do you think provides a more accurate estimate of sensor noise? Why?

3. **White balance.** (30 marks)

This question assesses a method for adjusting the white balance of your camera. Use an unlined piece of white paper as a standard. Take two pictures: one of a complex indoor scene and one of a complex outdoor scene. In both cases, embed your standard in the scene so that it is highly visible and in a region of constant illumination.

- (a) (15 marks) Write a MATLAB function

`im = whiteBalance(im,top,bottom,left,right)`

that accepts an RGB image `im` and the bounds of a maximal rectangular region (in pixels) lying within your standard and returns a new image of the same data type as the input image but corrected for white balance. Specifically, your function should equalize the RGB values at each pixel so that the average red, green and blue values are all the same over the specified rectangular region and the mean luminance for the entire image is unaltered. Answer the following questions for both indoor and outdoor scenes.

Some useful MATLAB functions: `figure`, `subplot`, `title`, `set`, `imagesc`, `axis`, `size`, `reshape`, `repmat`, `mean`, `double`, `cast`, `class`.

- (b) (5 marks) Show side-by-side plots your images before and after white balancing.
- (c) (5 marks) In which image does your standard look more white? Why do you think this is?
- (d) (5 marks) Which image looks better? Why?