ECE 472 Robotic and Computer Vision

Prof. K. Dana

Homework 1: Linear Algebra Review

1. Write a Python program to determine if there are only two independent columns in the following matrix:

$$\begin{bmatrix} -4.22 & 1.83 & -2.22 \\ -0.78 & -3.44 & -1.92 \\ 1.26 & -0.71 & 0.60 \end{bmatrix}$$

2. Given the following coordinates for measurement points, solve the problem of fitting a line to these points using linear least square estimation. You are to solve the problem **two** ways: (1) by hand, (2) with a Python program. Be sure to set up the problem in the form $Aq = b$ by defining A and b. Then compute the parameter vector q using least squares estimation. *Note: Fill in the sections of the Problem_2.ipynb file and submit.*

Points

x	y
-1.1	8.6
2.3	2.9
5.8	-5.8
-3.5	13.0
3.0	0.1
8.5	-9.9

3. As we discussed in class the following are basis vectors for $\mathbb{R}^3$ (3D Cartesian coordinate system).

$$w_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, w_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, w_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Are the following vectors basis vectors for $\mathbb{R}^3$? Why or why not?

$$w_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, w_2 = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, w_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

4. Do the following vectors form a basis for $\mathbb{R}^3$? Why or why not?

$$t_1 = \begin{bmatrix} 0.707 \\ -0.707 \\ 0 \end{bmatrix}, t_2 = \begin{bmatrix} 0.707 \\ 0.707 \\ 0 \end{bmatrix}, t_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

5. Do the following vectors span $\mathbb{R}^3$? Why or why not? Do these vectors form a basis?

$$w_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, w_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, w_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, w_4 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

6. The image in the file *cguitar.tif* is corrupted by a global background shading across the image. Assume this background shading is an image gain that is an affine function of the x, y image coordinates. That is, the corrupted image $e_c(x, y)$ is the true image $e(x, y)$ multiplied by $f(x, y)$ where $f(x, y)$ is modeled by an affine function of x and y. (That is, $f(x, y) = ax + by + c$). *Note: Use the Python file Problem_6.ipynb to get you started.*

$$e_c(x, y) = f(x, y) * e(x, y) \tag{1}$$

You may assume that the upper left corner in the true image should be uniformly white, i.e. $e(x, y) = 255$ for $x < 50$ and $y < 250$. (Note: x is the horizontal coordinate and y is the vertical coordinate. Be careful in Python matrices are given in row, column format so $e(x, y)$ is accessed using $e(y, x)$).

In the Python program do the following tasks:

(a) Use linear least squares estimation to find $f(x, y)$. Clearly state $f(x, y)$ and the estimated parameters obtained from your program.

(b) Use least squares solution to undo the corruption in the *cguitar.tif* image. Please display the uncorrupted image.

Hint: We discussed how a model leads to a system of linear equations. Equation 1 provides a model. For each datapoint, (all the points of the upper left corner), the model gives one equation. You may not need all the points, so a subset of this region may be sufficient. Consider Equation 1. What are the knowns? What are the unknown? Can you put this into $Aq = b$ form as in the line fitting problem? Then can you use Equation 1 to remove the effects of the shading everywhere in the image.

7. Generate points that fall on a rotated 2D ellipse (not aligned with horizontal or vertical axis). Show how SVD can be used to find the orientation of the ellipse. Demonstrate your results with two different rotation angles.

Note: Use the Python file Problem_7.ipynb to generate x and y coordinates for the ellipse.

Hint: Add code which uses *rpts* as input points and estimates the orientation of the rotated ellipse using SVD. Every point in the ellipse can be written as a linear combination of 2 basis vectors. SVD reveals the basis as the major and minor axes of the ellipse (in the first and second columns of U). Once you have the major axis of the ellipse, its orientation can be found by arctan(ycomponent/xcomponent). Next, demonstrate results with a different rotation angle. **Hint:** Pseudocode for generating the points on a non-rotated ellipse uses the parametric form of an ellipse angles=0:(2*pi/99):100;a=10;b=5;x=a*cos(angles);y=b*sin(angles);