Perception for Autonomous Robots
Homework 1

Sri Sai Charan Velisetti UID:117509755 $M.EngRobotics$

I. Problem 1

A. Problem 1 Part 1

As, we already have a predefined relationship between the field of view (ϕ), focal length (f) and the sensor size (d)

Formulae used,

$$\phi = 2 \times \tan^{-1}\left(\frac{d}{2f}\right) \tag{1}$$

We have, ϕ as the field of view, f as the focal length and d is the sensor size.

$$f = 25 \text{ mm}$$
$$d = 14 \text{ mm}$$

The horizontal and vertical FOV will be the same. As the sensor shape is square the formulae will also be the same.We can now substitute the values,

$$\phi = 2 \times \tan^{-1}\left(\frac{14}{2 \times 25}\right)$$
$$\phi = 31.28449292°$$

B. Problem 1 Part 2

To find the area covered by the image on the camera sensor we first have to calculate the height h
Given by:

$$\frac{\frac{H}{2}}{h_i} = \frac{f}{d}$$

Where,
(H) is actual height of the object,
(d_o) is camera to object distant
(h_i) is the image height
(f) is the focal length of camera.
Substituting the values(in mm), we get

$$\frac{25}{20 \times 10^3} = \frac{\frac{H}{2}}{h_i}$$
$$h_i = 5600 \text{ mm}$$

As the area of FOV is square shaped, Area of FOV will be $(2h^2)$

I would like to Express my Thanks to
Dr. Samer Charifa my professor, Mr Gokul Hari, Ms Sakshi kakde and Mr. Jayesh Jayashankar the teaching assiatants and the grader for ENPM673: Perception of Autonomous Robots

$$[2 * 5600]^2 = 125440000 \text{ mm}$$

To, find the number of Pixels covered by the image on the Camera Sensor we have to first find the pixel density per mm. Given by,

$$= \frac{Area(FOV)}{Resolution}$$
$$= \frac{125440000}{5000000}$$
$$25.08$$

Now, To find the number of pixels occupied by the object on the image sensor, Given by,

$$\frac{Area(Object)}{25.08}$$
$$= 99.6810 \;\; = 100 pixels$$

The minimum number of pixels occupied by the Object is 100 pixels

II. Problem 2

Here, three curve fitting techniques: Least Square Method, Image filtering techniques were implemented. To read the video file from the folder, convert it to gray scale, setting thresholds to get a binary output, OpenCV's inbuilt functions were used. The reason being working with BRG format for the given video is not needed and reduces the complexity of the code. We first extract the black pixels which denotes the ball. Find the top and bottom most pixel and save the coordinates to plot. These coordinates are used to fit a curve, which can be assumed as a parabola.

We then convert the collected coordinate points from the NumPy format which positions the orgin at the top left of the image to a Cartesian plane.

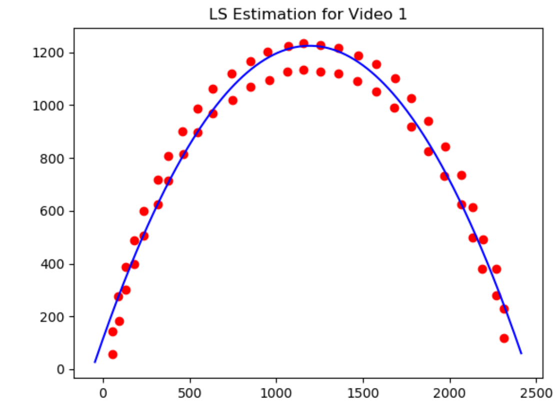
The equation of the parabola is given by:

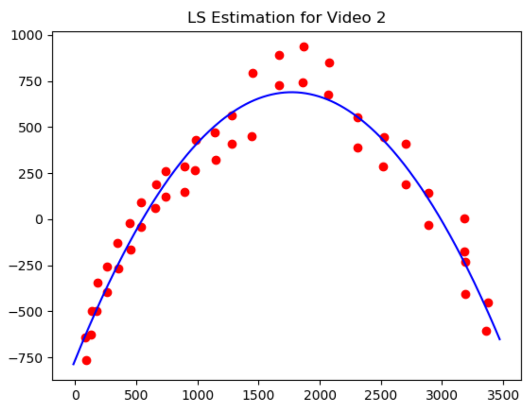
$$y = ax^2 + bx + c$$

The Error function is given by :

$$E = \Sigma_n \left(y - ax^2 - bx - c\right)^2$$

Results:

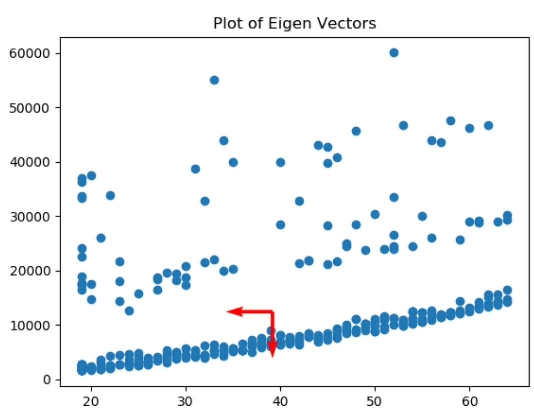




III. Problem 3

We are given a data frame consisting of multiple attributes but we are only interested in using the 'age' and 'Charges' attributes. We can use pandas to import the given .csv file. Then extract the required data.

A. Problem 3 Part 1 : Co-variance Matrix Eigen Vectors



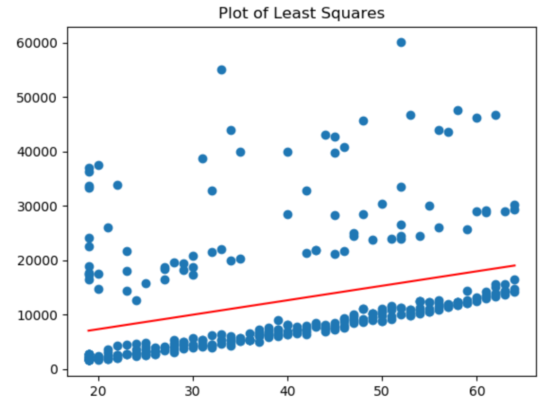
B. Problem 3 Part 2 : Least Square Method

The equation of the line is given by

$$y = mx + c$$

The error function can be written as,

$$E = \Sigma_n \left(y - mx - c\right)^2$$



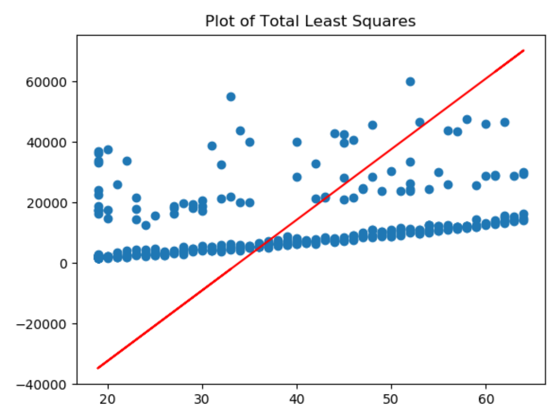
C. Problem 3 Part 3 : Total Least Square Method

The equation of the line is given by

$$y = mx + c$$

The error function can be written as,

$$E = \Sigma_n \left(y - mx - c\right)^2$$



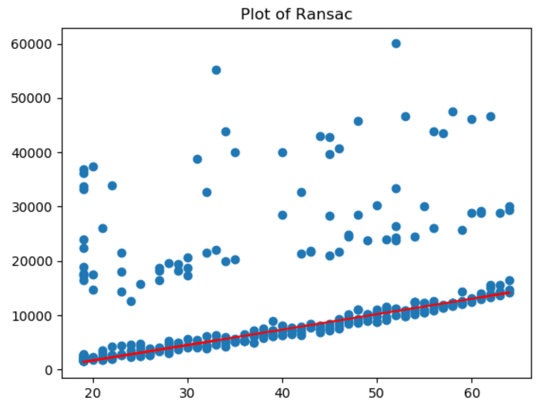
D. Problem 3 Part 4 : RANSAC

We select random points and check for the best fit on a TLS model and then select the coefficients with the least amount of error. Formulae used to calculate error terms:

$$E = \left(ax + by + c\right)^2$$

The number of iteration can be calculated using:

$$N = \frac{\log(1-p)}{\log\left(1-(1-e)^s\right)}$$



E. Reflection on Methods Used:

As we can observe from the methods implemented the most accurate method is the RANSAC method which is able to account for a large number of outliers. As ransac uses both TLS and caluclate the lowest error term for every point within a given number of iterations. We can see why it is more accurate.

IV. Problem 4

The 'A' matrix represent the orthogonal eigen vectors. Therefor, to find eigen vectors and eigen values of A . We arrange the eigen values in descending order and rearrange the eigen vector matrix as per the corresponding eigen value for each eigen vector.

A. SVD

We first Calculate:

$$AA^T$$

Then we can calculate U matrix represent the orthogonal eigen vectors of AA^T. Therefor, find eigen vectors and eigen values of AA^T

Arrange the eigen values in descending order and rearrange the eigen vector matrix as per the corresponding eigen value for each eigen vector.

As Σ^2 is the eigen value of AA^T, Σ (singular value) shall be the square roots of the eigen values. Find square roots of the eigen values on AA^T and arrange them as a diagonal matrix. We then calculate $A^T A$ as Matrix V.

As V matrix represent the orthogonal eigen vectors of $A^T A$. Therefor, find eigen vectors and eigen values of $A^T A$.

We then again, arrange the eigen values in descending order and rearrange the eigen vector matrix as per the respective eigen value for each eigen vector.

Formulae Used,

$$A = U\Sigma V^T$$
$$AA^T = U\Sigma V^T V \Sigma^T U^T$$
$$AA^T = U\Sigma\Sigma^T U^T$$

V. Conclusions

This homework has taught me how to use multiple python libraries to their fullest. By, using OpenCV

Acknowledgment

Author would like to thank, Dr. Samer Charifa my professor, Mr Gokul Hari, Ms Sakshi kakde and Mr. Jayesh Jayashankar the teaching assiatants and the grader for ENPM673: Perception of Autonomous Robots for their constant encouragement towards the realization of this work.