ENPM 673

Perception for Autonomous Robots

Homework 1

Student:
Pulkit Mehta
UID: 117551693

Instructor:
S. Charifa

Semester:
Spring 2022

February 11, 2022

Contents

1 Problem 1

1.1 Field of View

The field of view of a camera is given by

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

where, d is the camera sensor dimension and f is the focal length.

A function is created which takes d and f as arguments and returns calculated field of view. As the camera sensor is square shaped the values of horizontal and vertical field of view are identical. Substituting the given values in (1), we get:

$$Field\ of\ View = 31.28^o$$

1.2 Minimum number of pixels

The height of object in image created is given by:

$$height_{image} = \frac{f * height_{object}}{distance_{object}} \tag{2}$$

Similarly the width of object in image is also calculated. Thus, we obtain the area occupied by the object in the image. As the 5MP sensor covers area of 14 x 14 mm^2, by unitary method the number of pixels occupied by object is calculated. Hence,

$$Number\ of\ pixels = 99$$

2 Problem 2

2.1 Approach

The given video has a red coloured ball for which the trajectory has to be plotted. The frames from the video were extracted and the RGB image converted to HSV as a particular colour needs to be detected. The hsv image is then converted into a binary image using a mask with range of hue, saturation and value pertaining to red colour.

There are two possible ways of extracting the center of the ball:

- Detecting the topmost and the bottom most pixel with value of 1 from the binary(masked) image of the ball and then averaging both the values to get the center.
- Finding the first order moments of the detected contour and finding center coordinates from it.

The first method is not computationally efficient as it requires the use of nested loops for finding the extreme pixels. Thus, the data points were extracted using the second method and appended into an array for further operations.

2.2 Standard Least Squares

Since the trajectory of the ball follows the equation of a parabola, it can be expressed as:

$$y = ax^2 + bx + c \tag{3}$$

where, a,b and c are model parameters. Thus the equation [3] can be written in matrix form as,

$$AX = Y \tag{4}$$

where, $A = \begin{bmatrix} {x_1}^2 & x_1 & 1 \\ {x_2}^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ {x_n}^2 & x_n & 1 \end{bmatrix}$ $X = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ $Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$

The equation is solved to get the optimum values of X:

$$X = (A^T A)^{-1}(A^T Y) \tag{5}$$

The least squares solution is given by,

$$A \cdot X = A \cdot (A^T A)^{-1}(A^T Y) \tag{6}$$

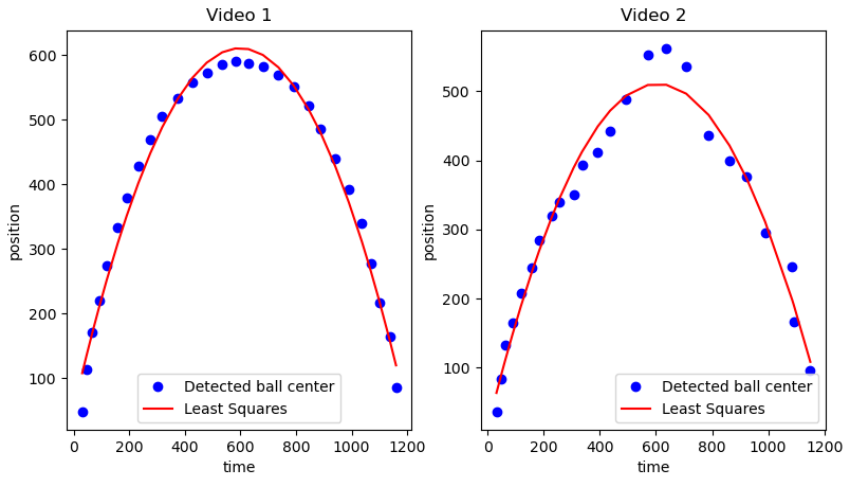


Figure 1: Standard Least Squares fit on ball trajectory in Video 1 and Video 2

The problem of creating the A matrix was resolved by referring to [3].

3 Problem 3

The given data is in the form of a csv file. A pandas dataframe is created of the given data and the required columns of data are extracted from the dataframe,

i.e., age and insurance charges.

3.1 Covariance Matrix

The covariance matrix,S is defined for the given data as,

$$S = \sum_{i=1}^{n} \frac{(X - \bar{X})(Y - \bar{Y})}{n - 1} \tag{7}$$

where, X and Y are the column vectors from the given data, namely, age and cost respectively. $\bar{X}$ and $\bar{Y}$ being the mean of the column. A new array A is created after subtracting the column-wise mean of data from the original data. Now, the covariance matrix S is given by,

$$S = \frac{A^T A}{n} \tag{8}$$

where n is the number of data points in the given data. The eigenvectors of S are plotted as shown in Fig.2.

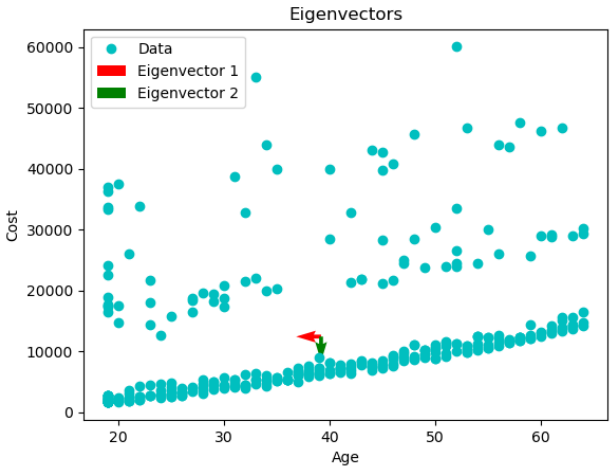


Figure 2: Plot depicting eigenvectors of covariance matrix.

3.2 Linear Least Squares

The method is similar to the one used in Problem 2, the only difference being the model to be fitted is a line. Thus,

$$y = ax + b \tag{9}$$

where a and b are model parameters. The matrices in [4] being,

$$A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \qquad X = \begin{bmatrix} a \\ b \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Now [6] can be used to get the optimum fit.

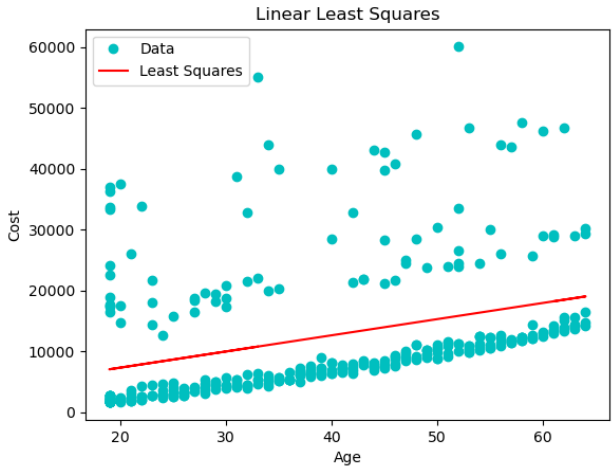


Figure 3: Linear Least Squares fit

3.3 Total Least Squares

The matrix U is defined as[1]:

$$U = [X_i - \bar{X} \quad Y_i - \bar{Y}] \tag{10}$$

where, X and Y are column vectors. Here, they constitute data of age and charges respectively.

The solution of total least squares method is given by,

$$y_{tls} = \frac{d - a * x_i}{b} \tag{11}$$

where, a and b are the components of the eigenvector corresponding the smallest eigenvalue of the matrix $(U^T U)^T \cdot U^T U$ and

$$d = a(x_{mean}) + b \tag{12}$$

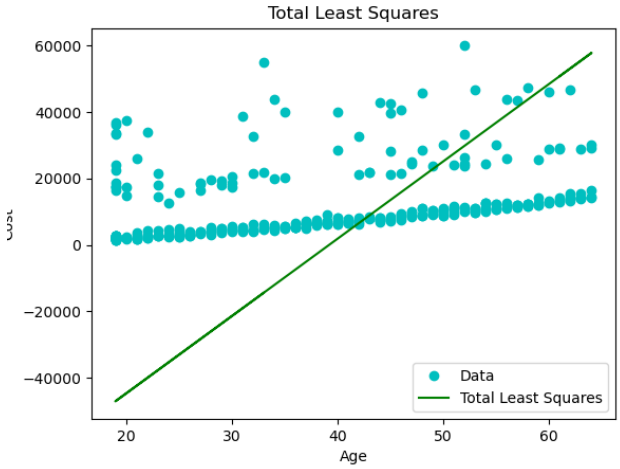


Figure 4: Total Least Squares fit

3.4 Random Sample Consensus(RANSAC)

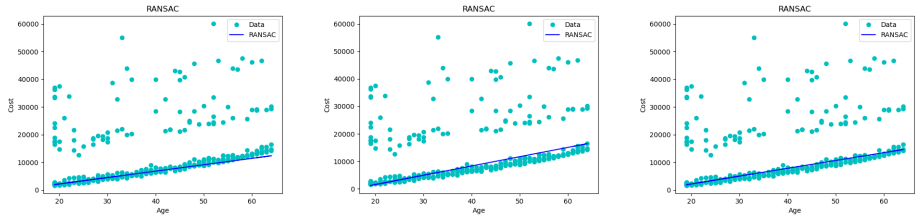
The minimum number of points needed to fit the model is taken to be 2. The maximum number of iterations(N) is initially assigned as infinity and the sample count as zero. The number of inliers are counted according to the the distance threshold after a least squares fit on the data. The error is calculated as,

$$error = 1 - \frac{\text{number of inliers}}{\text{total number of points}} \tag{13}$$

The values of N are updated adaptively using,

$$N = \frac{log(1-p)}{log(1-(1-e)^s)} \tag{14}$$

where, p is the desired probability of inlier and is taken as 0.95 in this case.



(a) $threshold = \sigma$ (b) $threshold = \sigma/3$ (c) $threshold = \sigma/5$

Figure 5: TLS fit for different threshold values.

The algorithm performed better for threshold of $\sigma/5$ as compared to σ and $\sigma/3$, where σ is the standard deviation of the given data.

Occasionally the algorithm threw an exception of 'Singular Matrix' during calculation of model parameters. The main cause for it being the lack of presence of inliers within the specified lower values of threshold for some randomly selected pair of points. However, the increase in threshold value after a certain extent may result in reducing the efficiency of the outlier rejection technique.

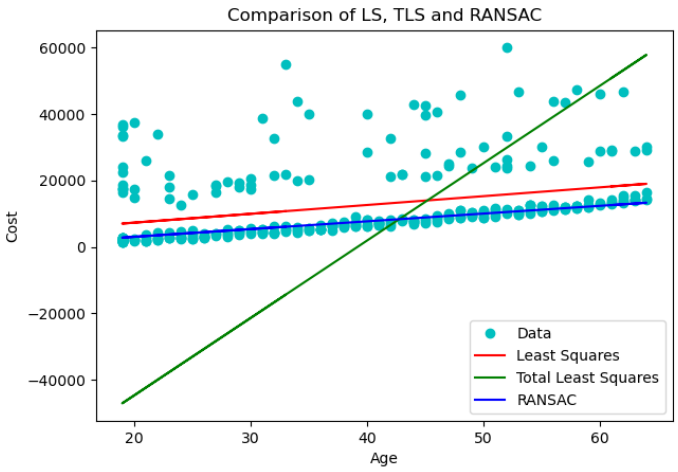


Figure 6: Plot depicting all the methods

The method of Least squares severely penalizes the presence of outliers. Thus, RANSAC is a better alternative as due to its iterative procedure, with minimum number of sample points, an optimum model fit is obtained. However, using RANSAC can be computationally expensive where the outliers are less or the given data points are plotted in higher dimensions. The method of total least squares has an advantage over least squares by being rotation invariant. However, it also has the same shortcoming as of least squares of poorly performing when there are significant number of outliers. If the data was not scattered and the error was minimal, then TLS would have been an optimum choice as it has an edge over RANSAC in terms of efficiency and computational power required.

Hence, for the given data RANSAC performs the best followed by LS and then TLS.

4 Problem 4

The given point correspondences being,

	x	y	xp	yp
1	5	5	100	100
2	150	5	200	80
3	150	150	220	80
4	5	150	100	200

The Singular Value Decomposition of a matrix is given by:

$$SVD = U\Sigma V^T \tag{15}$$

where,

- U is the matrix with columns as eigenvectors of AA^T. Thus, the characteristic equation can be written as,

  $$(AA^T - \lambda I) = 0 \tag{16}$$

  The obtained eigenvalues from [16] can be substituted into [17]

  $$(AA^T - \lambda I)x = 0 \tag{17}$$

  and the null space is determined. Similarly all the eigenvalues and eigenvectors are determined and the matrix U is constructed.

- V is the matrix with columns as eigenvectors of A^TA and the matrix can be contructed following the same steps as in the case of U.

- Σ is a diagonal matrix with the diagonal values as square root of the eigenvalues of AA^T or A^TA, eigenvalues of both being equivalent.

The eigenvectors had to be sorted in decreasing order of eigenvectors so that the common eigenvalues and corresponding eigenvectors were aligned together. The issue of creating a the diagonal matrix Σ was resolved by using the inbuilt *numpy.zeros_like* method and then assigning the diagonal values.

The eigenvector corresponding to zero eigenvalue for A^TA is our solution. As the matrix V was sorted in descending order of eigenvalues, we need to extract the last column from matrix V corresponding to eigenvalue closest to zero[2].

Now the column is reshaped into a 3x3 matrix, which is our Homography matrix.

The Homography Matrix computed is:
$$\begin{bmatrix} 0.053105 & -0.004917 & 0.614685 \\ 0.017701 & -0.003933 & 0.786750 \\ 0.000236 & -0.000049 & 0.007621 \end{bmatrix}$$

References

[1] Duke University CEE 629, System Idenfication. Total least squares. Fall 2017.

[2] CSE 252A Winter 2007 David Kriegman. Computer vision 1: Homography estimation.

[3] Dmitrii Khizbullin URL:. https://hackernoon.com/ransac-ols-pca-3-ways-to-draw-a-straight-line-across-a-set-of-points-828q34ds.
[3] [1] [2]