Umang Rastogi - 116773232
Sayani Roy - 116818766

Assignment - 1

Problem 1

Part 1

Given:
Shape of camera sensor: Square
Side of camera sensor: d = 14mm
Focal length: f = 15mm

Now, since the camera sensor is square in shape the vertical and horizontal Field of View (FOV) will be the same. It can computed using the equation:

$$FOV = 2tan^{-1}(\frac{d}{2f}) \tag{1}$$
$$\implies FOV = 2tan^{-1}(\frac{14}{2 \times 15}) \tag{2}$$
$$\implies FOV = 2tan^{-1}(\frac{7}{15}) \tag{3}$$
$$\implies FOV = 50.033° \tag{4}$$

Equation 4 gives the field of view of the camera in both, horizontal as well as vertical directions.

Part 2

Assuming the lens is thin to apply thin lens formula for calculation of height of the image. The thin lens formula are given below:

$$\frac{1}{f} = \frac{1}{d_o} + \frac{1}{d_i} \tag{5}$$
$$\frac{h_i}{h_o} = -\frac{d_i}{d_o} = m \tag{6}$$

where,
f: focal length of the camera
d_o: distance of object from the lens
d_i: distance of image from the lens
h_o: height of the object
h_i: height of the image
m: magnification of the lens

Using Equation 5, we get,

$$\frac{1}{15 \times 10^{-3}} = \frac{1}{20} + \frac{1}{d_i}$$
$$\implies \frac{1}{d_i} = \frac{1000}{15} - \frac{1}{20}$$
$$\implies \frac{1}{d_i} = \frac{4000 - 3}{60}$$
$$\implies d_i = \frac{60}{3997} \approx 15mm$$

Using Equation 6 and value of d_i, we get,

$$\frac{h_i}{50} = -\frac{15 \times 10^{-3}}{20}$$
$$\implies h_i = -\frac{15}{20 \times 1000} \times 50$$
$$\implies h_i = 0.0375mm$$

Now, assuming that the object is square, we can calculate the area occupied by the object in the image. We can map the area to resolution using unitary method to get,

$$(14)^2 \simeq 5MP$$
$$\implies (0.0375)^2 \simeq \frac{0.0375^2}{14^2} \times 5MP$$
$$\implies (0.0375)^2 \simeq 36pixels$$

Hence, the object will occupy a minimum of 36 pixels in the image.

Problem 2

The following steps were used to fit a parabolic curve to the given datasets:

- A loop is ran to go through each dataset.
- Data is read and split into separate dataframes: one for x-data and the other for y-data.
- Two methods are applied on each dataset for curve-fitting: LS (Least Squares) Fitting and RANSAC (RANdom SAmple Consensus) with LS Fitting.
- The dataset and the fitted-curve are plotted for both the methods to get 2 plots for each dataset, one for each method.
- Solutions yielded by both methods are compared visually as well as using coefficients (a, b, c) of the general quadratic equation: $y = ax^2 + bx + c$

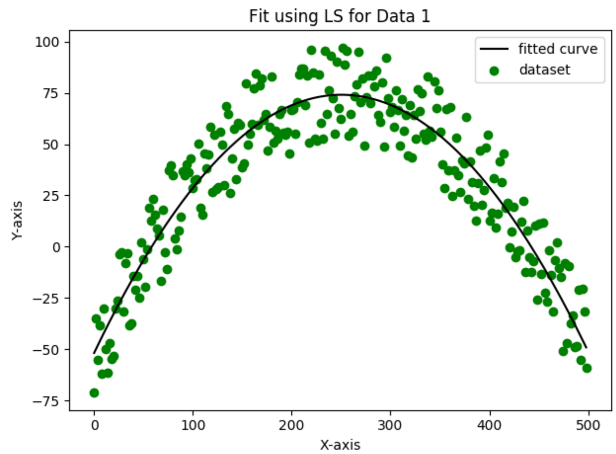

Figure 1: Least Squares Fitting

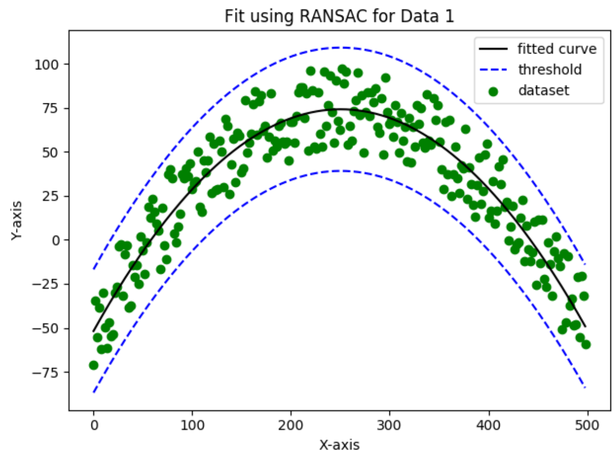

Figure 2: RANSAC with threshold = 35

- To apply the LS Fitting method, we have used the standard formula: $B = (X^T X)^{-1}(X^T Y)$ and X has been transformed to make it a (3xn) matrix. X is of the form: X = $[x^2 \ x \ 1]$.
- Please note that RANSAC involves setting an error threshold to cast out noisy points in the data. The setting of the threshold required a bit of playing around with values from 20 to 50.
- We had to play around with the value of inlier ratio to get the best possible fitting using RANSAC. A higher inlier ratio could lead to no solution at all.

Based on visual inspection of the data using a scatter plot, it seems that the second dataset has outliers while the first one does not. Therefore, we will have to apply an outlier rejection technique for the second dataset.

We ran 2 methods on both the datasets, Least Squares (LS) Fitting and RANSAC with LS Fitting. For the first dataset, both the methods yielded the same coefficients for the fitting-parabola. The plots for the first dataset are given in Figure 1 and Figure 2 above. We considered the general equation of the parabola: $y = ax^2 + bx + c$ and coefficients to fit the dataset yielded by both the methods were:

$$\begin{bmatrix} a & b & c \end{bmatrix} = \begin{bmatrix} -0.00201 & 1.00598 & -51.83883 \end{bmatrix} \tag{7}$$

For the second dataset, a significant difference can be seen between Figure 3 and Figure 4 below, to reject outliers using RANSAC. The LS fitting method yields coefficients:

$$\begin{bmatrix} a & b & c \end{bmatrix} = \begin{bmatrix} -2.45101 & 1.23731 & -51.92573 \end{bmatrix} \tag{8}$$

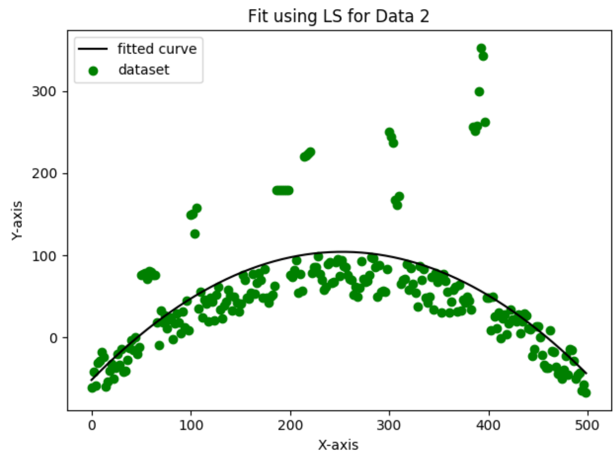

Figure 3: Least Squares Fitting

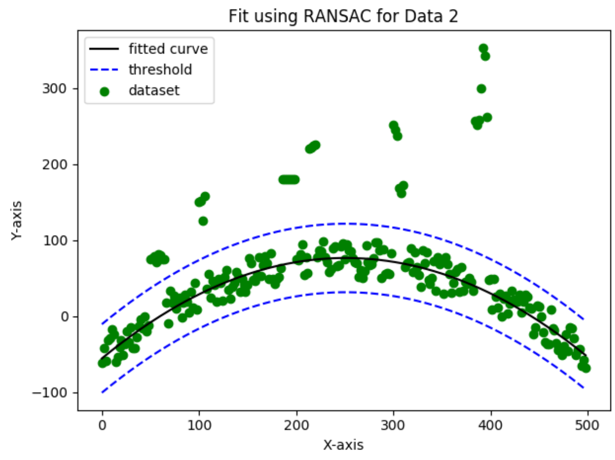

Figure 4: RANSAC with threshold = 45

while RANSAC yields:

$$\begin{bmatrix} a & b & c \end{bmatrix} = \begin{bmatrix} -2.09519 & 1.05188 & -55.42841 \end{bmatrix} \tag{9}$$

with a threshold of 45.

Please note that the coefficients have rounded-off to include only up to 5 decimal places.

In the first dataset, the least square fitting method seems to be better choice over RANSAC. Since RANSAC is computationally heavier than LS fitting, the fact that it provides with the same result as LS fitting makes RANSAC an overkill for first dataset. Hence, LS Fitting is the best method for the first dataset.

On the other hand, RANSAC performs way better than LS Fitting on the second dataset to reject outliers making it the best method to fit a curve to the dataset.

In general scenarios such as the real-world ones, visual inspection of the data would not be always possible. In real-world applications, there would always be noise due to the environment or the system. If we consider that both the datasets were from the same system, RANSAC would overall be the best method to be employed on the system as irrespective of the presence of outliers RANSAC would ensure the delivery of the best fitting-curve for that data.

Problem 3

The homography between 4 corrresponding points on two differnt planes is computed using the following system of equations Ax=0, where:

$$A = \begin{bmatrix} -x1 & -y1 & -1 & 0 & 0 & 0 & x1*xp1 & y1*xp1 & xp1 \\ 0 & 0 & 0 & -x1 & -y1 & -1 & x1*yp1 & y1*yp1 & yp1 \\ -x2 & -y2 & -1 & 0 & 0 & 0 & x2*xp2 & y2*xp2 & xp2 \\ 0 & 0 & 0 & -x2 & -y2 & -1 & x2*yp2 & y2*yp2 & yp2 \\ -x3 & -y3 & -1 & 0 & 0 & 0 & x3*xp3 & y3*xp3 & xp3 \\ 0 & 0 & 0 & -x3 & -y3 & -1 & x3*yp3 & y3*yp3 & yp3 \\ -x4 & -y4 & -1 & 0 & 0 & 0 & x4*xp4 & y4*xp4 & xp4 \\ 0 & 0 & 0 & -x4 & -y4 & -1 & x4*yp4 & y4*yp4 & yp4 \end{bmatrix} \text{ and } x = \begin{bmatrix} H11 \\ H12 \\ H13 \\ H21 \\ H22 \\ H23 \\ H31 \\ H32 \\ H33 \end{bmatrix} \tag{10}$$

. For the given point correspodences:

x	y	xp	yp
5	5	100	100
150	5	200	80
150	150	220	80
5	150	100	200

(11)

we can find the homography matrix

$$H = \begin{bmatrix} H11 & H12 & H13 \\ H21 & H22 & H23 \\ H31 & H32 & H33 \end{bmatrix} \tag{12}$$

using Singular Value Decomposition method.
Singular Value Decomposition or SVD is an expansion of the original data in a coordinate system where the covariance matrix is diagonal. Any given matrix A_{mxn} can be represented as

$$A_{mxn} = U_{mxm} * \Sigma_{mxn} * V^T_{nxn} \tag{13}$$

The first step of calculating the SVD is to find the eigenvalues and eigenvectors of AA^T and A^TA. The eigenvectors of A^TA make up the columns of V and the eigenvectors of AA^T make up the columns of U. The matrix Σ consists of singular values of A which are square roots of eigenvalues from AA^T or A^TA.

From the given points, the matrix A can be written as

$$A = \begin{bmatrix} -5 & -5 & -1 & 0 & 0 & 0 & 500 & 500 & 100 \\ 0 & 0 & 0 & -5 & -5 & -1 & 500 & 500 & 100 \\ -150 & -5 & -1 & 0 & 0 & 0 & 30000 & 1000 & 200 \\ 0 & 0 & 0 & -150 & -5 & -1 & 12000 & 400 & 80 \\ -150 & -150 & -1 & 0 & 0 & 0 & 33000 & 33000 & 220 \\ 0 & 0 & 0 & -150 & -150 & -1 & 12000 & 12000 & 80 \\ -5 & -150 & -1 & 0 & 0 & 0 & 500 & 15000 & 100 \\ 0 & 0 & 0 & -5 & -150 & -1 & 1000 & 30000 & 200 \end{bmatrix} \tag{14}$$

Since, A is a 8x9 matrix, A^TA will give us a 9x9 square matrix. The eigenvalues of A^TA can be computed from $det|A^TA - \lambda I| = 0$. For each eigenvalue, there exists an eigenvector which will satisfy the characteristic polynomial equation of $det|A^TA - \lambda I| = 0$. These eigenvectors are the columns of the V matrix.
The U matrix is similarly computed using AA^T which will give a 8x8 square matrix. The eigenvalues of AA^T is determined from $det|AA^T - \lambda I| = 0$ and the eigenvectors corresponding to each eigenvalue make up the U matrix.
The value of the elements of homography matrix H is the last column of the V matrix.
The Σ matrix is a 8x9 matrix with non zero diagonal elements. The square root of the non zero eigenvalues of A^TA make up the diagonal elements of Σ matrix. The diagonal entries of the Σ are arranged in descending order.