Ball Detection and Regression Analysis -Project 1

Vinay Krishna Bukka (118176680)

February 2023

Contents

1 Problem 1

1.1 Image processing of Video

The video given is read using the OpenCV library and is processed to achieve the goal of getting the center coordinates of ball in the video

- **Approach**

  1. The Video is read by creating a video writer object which is used to read each frame in the video as a image.
  2. Each frame of the image is converted to HSV as HSV has more colour strength for particular colour of interest than RGB.
  3. A trial and error approach is implemented to get the range of red colour ball in each frame. These ranges are used in cv2.inRange function which is used as a masking resulting a grayscale image showing only ball in white of each frame.
  4. Bitwise operations are then performed to apply the mask image onto the original frame.
  5. The center of the ball in each frame is calculated by first extracting the pixel coordinates of the ball in the frame(using np.where function with a condition of pixel value not equal to zero) and applying the mean.
  6. Finally the center of the ball in each frame is highlighted and shown to the console.
  7. Additionally, a 2D plot is shown with centers of ball coordinates.

- **Problems & Solutions**

  1. The first major challenge faced in this problem is to find the correct range of red pixels corresponding to only the ball in the frame. Initially, a low range $[0, 175, 110]$ and high range of $[180, 255, 255]$ is used which is able to capture the full image of ball in each frame. But, each frame had some background noise resulting in irregularities of curve fitting done further. So, to remove the noise from each frame a low range of $[0, 175, 110]$ and high range of $[4, 255, 255]$ along with a blurring the background noise. Through this, each frame is now processed with little or no background noise with giving exact center coordinates of ball though the ball image in some frames is not fully masked.
  2. The next challenge was to get the center coordinates of ball in each frame. This has been achieved using np.where function easily instead of iterating the pixels of each frame and performing extraction

- **Results** The HSV, masked and original images of certain frame is shown in Fig [1]. Also, a 2D plot of ball center is shown in Fig [2].

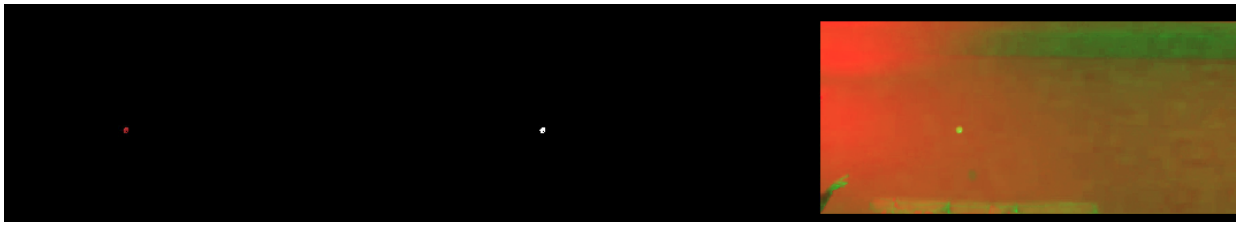

Figure 1: Original, Masked, HSV Image of a frame in video

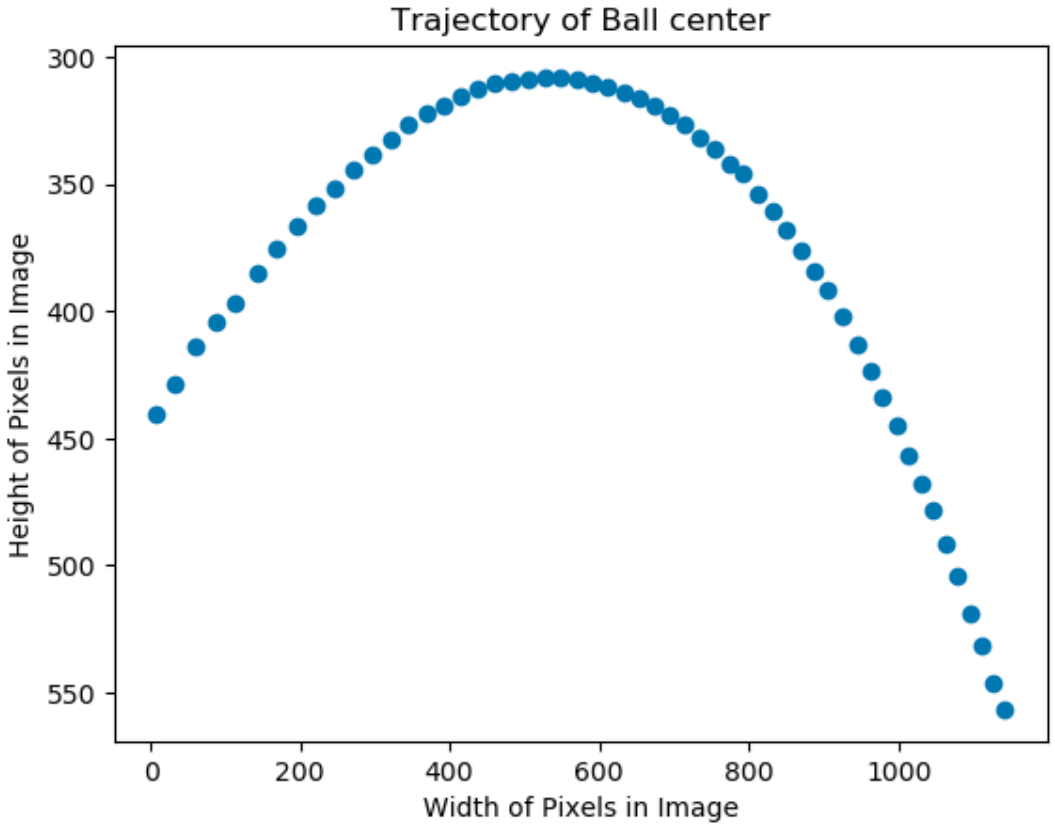

Figure 2: Trajectory of the Ball

1.2 Parabolic Curve Fitting

- **Approach**

1. The centers of ball coordinates in each frame are stored in separate lists. The general equation of parabola is : $y = ax^2 + bx + c$. Three equations are formulated and a matrix is formed from these equations (ref [4]) $\begin{bmatrix} \sum x^4 & \sum x^3 & \sum x^2 \\ \sum x^3 & \sum x^2 & \sum x \\ \sum x^2 & \sum x & n \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum x^2 y \\ \sum xy \\ \sum y \end{bmatrix}$
2. Using the numpy.linalg.solv function, we get the values of a,b,c after formulating above matrices.
3. The equation of parabola after substituting the constants derived is

   $y = 0.01x^2 - 0.606x + 458.406$

- **Problems & Solutions**

    1. Initially, when the plot is created, the parabola was inverted. When closely observed, the y-axis of the plot has starting point from left bottom. Realizing that the image pixel coordinates for y-axis always start from left-top, invert y-axis command from matplotlib library is used to get the correct parabolic curve representing the ball center.

- **Results** The parabolic curve formed using the above equation and the scatter plot of ball center coordinates are shown in Fig [3]

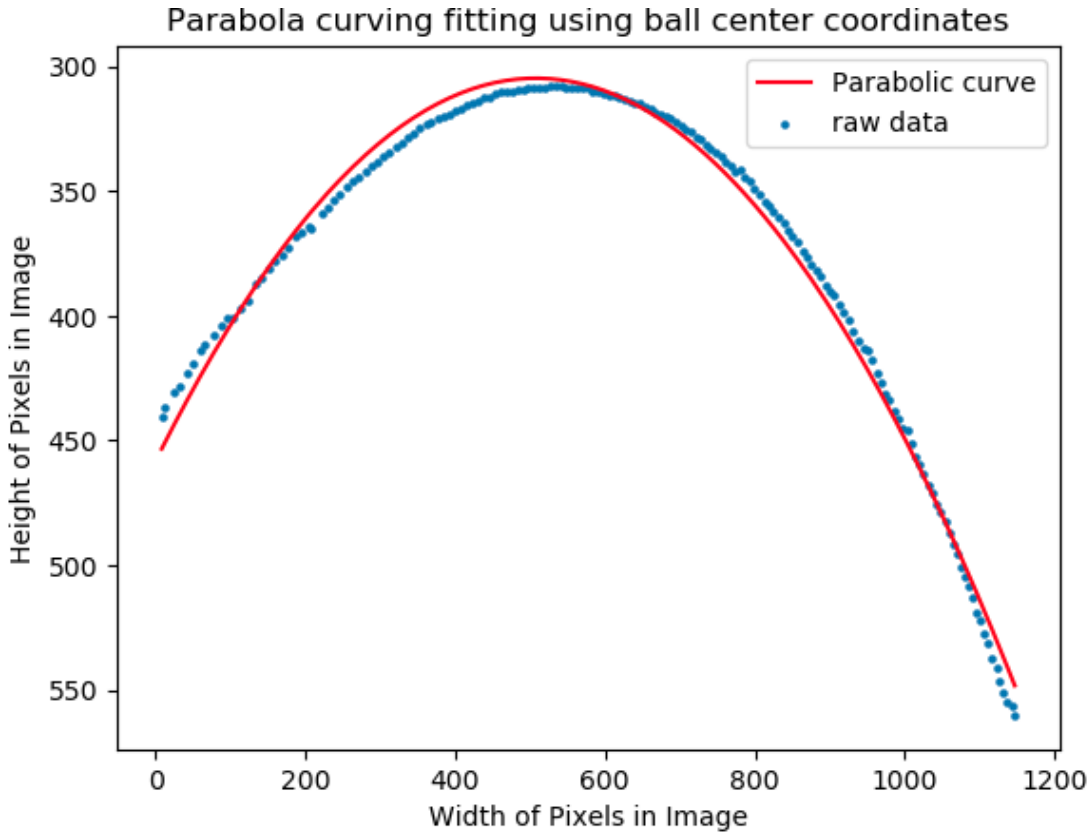

Figure 3: Parabolic Curve Fitting of the Ball

1.3 Landing X-Coordinate of Ball

- **Approach:** To find the landing x coordinate of the ball , the parabolic equation derived above is used. The y coordinate is 300 plus first element from the y center list which is the starting point of the ball. Using the y-coordinate and coefficients of the parabolic equation, a quadratic equation is formed. The roots of the quadratic equation gives the x-coordinate of the ball landing spot.

- **Results** The positive root among the roots derived is the desired x-coordinate. The X-coordinate of the ball landing spot is 1362

2 Problem 2

2.1 Covariance Matrix and Surface Normal Derivation

- **Approach**

  1. Covariance matrix is a square matrix that displays the variance exhibited by elements of datasets and the covariance between a pair of datasets. Variance is a measure of dispersion and can be defined as the spread of data from the mean of the given dataset. Covariance is calculated between two variables and is used to measure how the two variables vary together.
  2. The formula used for getting covariance between two variables is : $cov(a,b) = \frac{\sum_1^n (a-a_i)(b-b_i)}{n}$ where a_i & b_i are the means of respective data column
  3. The covariance matrix below is used to get the covariance matrix for the dataset "pc1.csv"

     $$\begin{bmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(x,y) & cov(y,y) & cov(y,z) \\ cov(x,z) & cov(y,z) & cov(z,z) \end{bmatrix}$$
  4. The Surface Normal of a plane is derived from covariance matrix. Surface Normal is defined as the normal vectors from all points on the surface of image. The eigen decomposition of the covariance matrix is performed to evaluate the surface normal. The eigen vector corresponding to minimum eigen value of the covariance matrix is the surface normal.
  5. The magnitude of the surface normal is derived from norm of the minimum eigen vector and the direction of surface normal is in the direction of minimum eigen vector

- **Results**

  1. The Covariance matrix calculated is given below

     $$\begin{bmatrix} 33.6375584 & -0.82238647 & -11.3563684 \\ -0.82238647 & 35.07487427 & -23.15827057 \\ -11.3563684 & -23.15827057 & 20.5588948 \end{bmatrix}$$
  2. The Magnitude of the Surface Normal is 0.99

2.2 Standard Least Squares

- **Approach**

  1. The method of least squares regression allows you to find a two-variable or a three-variable linear equation $y = mx + b$ or $z = ax + by + c$ that provides the best fit for the data points or point cloud in this scenario. In Standard least squares, fit is defined as minimizing the squared vertical errors, that is finding the values of m and b that minimize the function in case of two variable or finding a,b,c in case of three-variable

2. The Minimized error function for a three variable point cloud plane can be written as $F(a,b,c) = \sum(z - ax - by - x)^2$. By doing the parital derivatives of the Function F with respect to a,b,c gives rise to below matrix(in reference to [1]):

$$\begin{bmatrix} \sum x^2 & \sum xy & \sum x \\ \sum xy & \sum y^2 & \sum y \\ \sum x & \sum y & n \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum xy \\ \sum yz \\ \sum z \end{bmatrix}$$

3. The point cloud data of two data sets given are substituted in above matrix equation. The numpy.linalg.solv function is used to get the coefficients of a,b,c which represent the minimzed square function of the plane.

4. With the coefficients of plane derived above, now using the equation of the plane $z = ax + by + c$ a new z vector is formulated which represents the fitting of surface on the plane.

- **Problems & Solutions** Understanding the problem of Standard and Total Least Squares for a plane was difficult in the start since the lecture discussion was only about line fit. Also, the implementation of plane as a wireframe or 3D surface could not be achieved.

- **Results** The plane fitted using the standard least square regression along with the raw point clouds for both the data sets is given below Figures

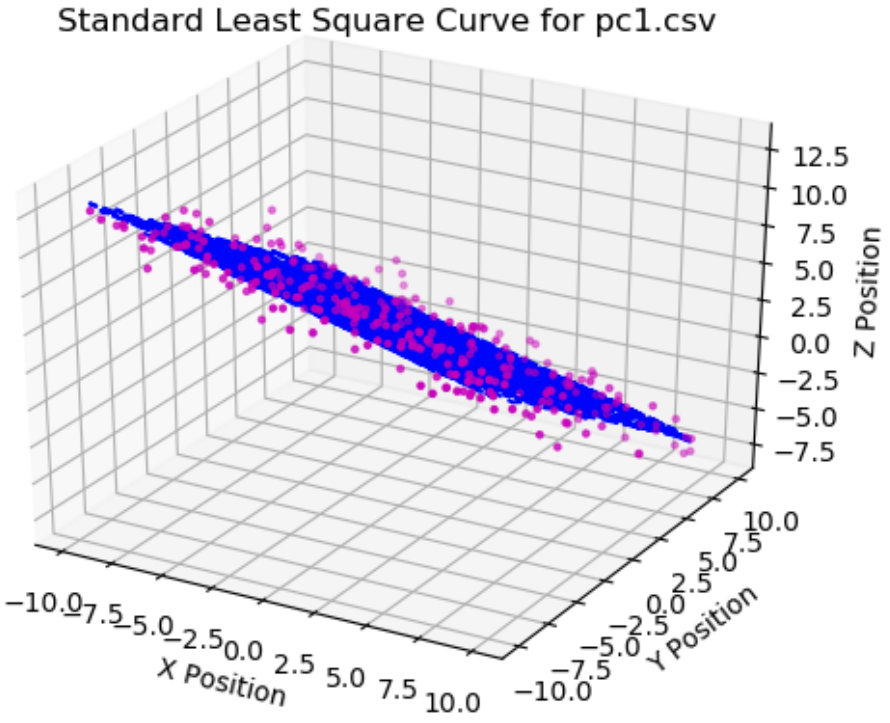

Figure 4: Standard Least Square Surface for First Point Cloud

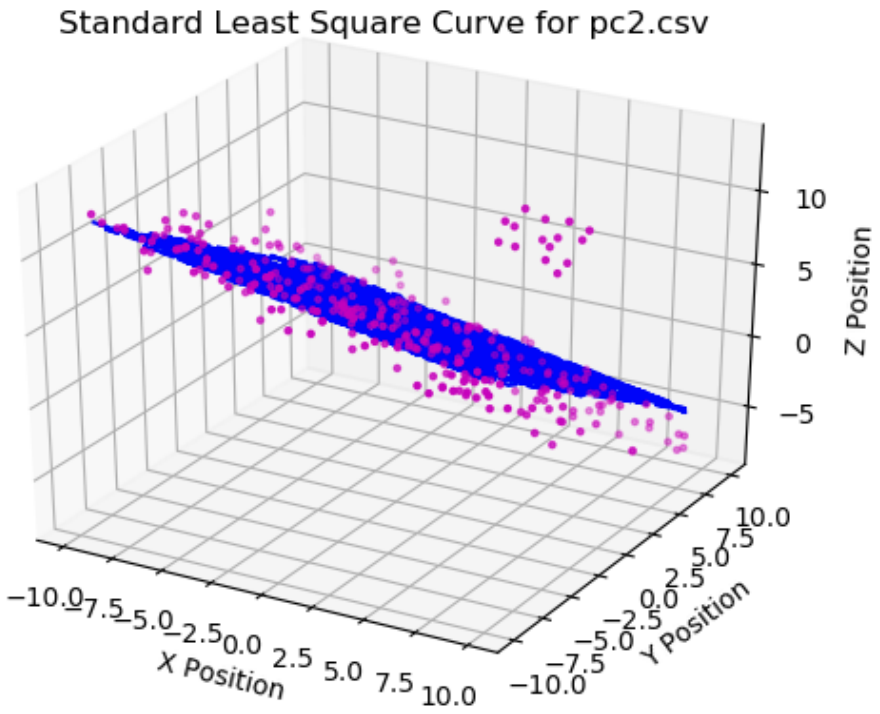

Figure 5: Standard Least Square Surface for Second Point Cloud

2.3 Total Least Squares

- **Approach**

1. Total least squares is one of the regression analysis methods to minimize the sum of squared errors between a response variable and a predicted variable. In total Least Squares we use the perpendicular distance points and perpendicular to plane and minimize the square of perpendicular distances.
2. The equation of the plane can be written as, $ax + by + cz = d$ and the minimized error function is written as $E = \sum(ax + bx + cz - d)^2$. Substituting the value of d which represents the plane equation over mean and rearranging terms in reference[2], we get the U matrix as
$$\begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} & z_1 - \bar{z} \\ \dots & \dots & \dots \\ x_n - \bar{x} & y_n - \bar{y} & z_n - \bar{z} \end{bmatrix}$$
where $\bar{x}, \bar{y}, \bar{z}$ represent the means calculated of the data set as in d
3. The eigen decomposition of $U^T U$ is used for deriving the coefficients of the minimized error equation.
4. The coefficients of a,b,c is given from the eigen vector corresponding to minimum eigen value. From these coefficients the dependent z array is formed through plane equation $z\_final = \frac{(d-(ax+by))}{c}$

- **Results** The plane fitted using the standard Total square regression along with the raw point clouds for both the data sets is given below Figures

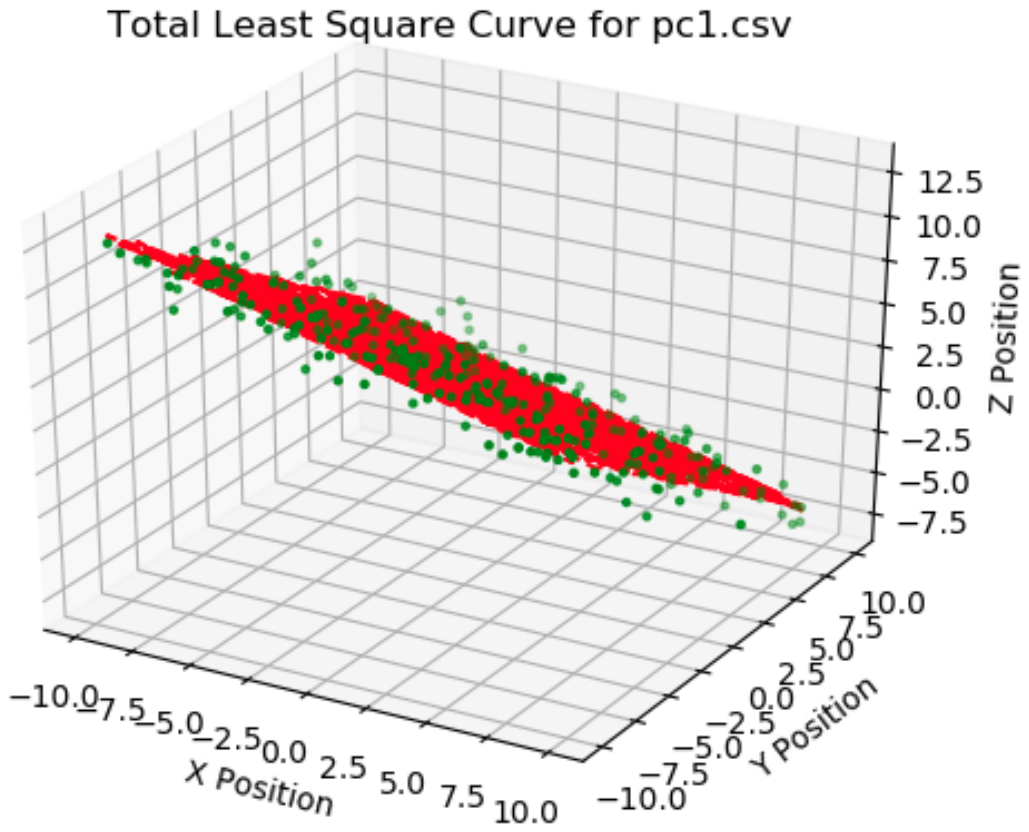

Figure 6: Total Least Square Surface for First Point Cloud

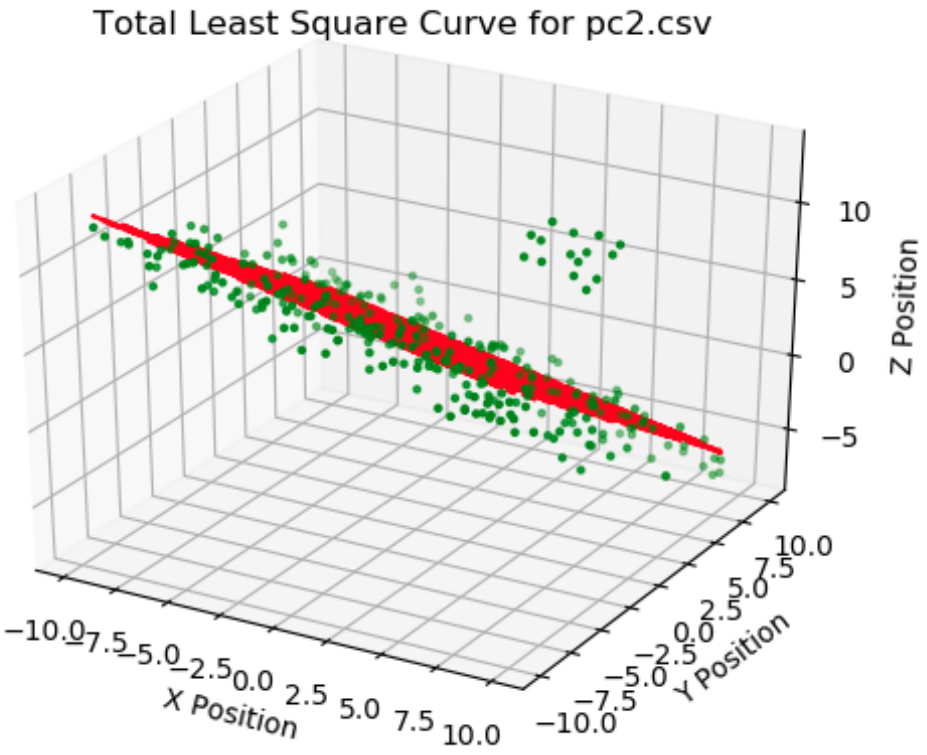

Figure 7: Total Least Square Surface for Second Point Cloud

2.4 RANSAC(Random Sample Consensus

- **Approach**

1. First a random sample of three points is selected from the (x,y,z) of the given point clouds.
2. A threshold of 0.08 and number of iterations is assumed as 1000
3. For each iteration with a new random sample of three points, coefficients are calculated of the plane($ax + by + cz + d = 0$) through Total Least Square method.
4. Deviation of all points in the cloud is calculated using the distance estimate in reference[3] which is given as
$$d = \frac{ax_i + by_i + cz_i + d}{\sqrt{a^2 + b^2 + c^2}}$$
5. If distance is within the threshold given, then it is considered as an inlier and stored in a list. The final inlier is selected when we have maximum amount of points in the plane that satisfy the threshold condition. From this final inlier through the coefficients a,b,c,d we calculate the dependent z vector to fit a surface.

- **Problems & Solutions**

  1. Understanding different kinds of implementation for RANSAC and choosing best understandable solution was the first problem.
  2. Secondly, RANSAC plane curve fitted was not a good fit for both the point clouds when threshold was 0.05. When increased the threshold by 10, the curves were completely off the data set. After manual tuning, a threshold of 0.08 gave a good fit plane for a good amount of random samples.

- **Results** The plane fitted using the RANSAC regression along with the raw point clouds for both the data sets is given below Figures. The below figures may vary since RANSAC considere different random samples in each RANSAC process

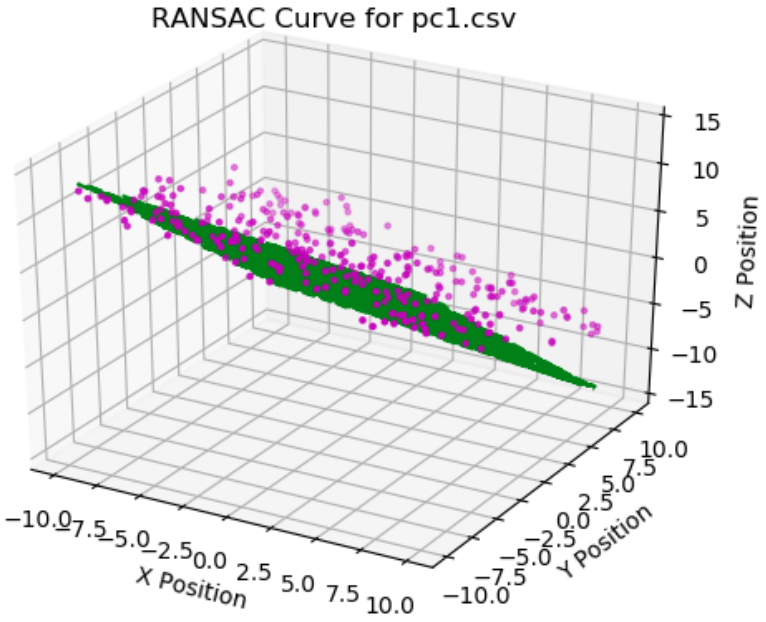

Figure 8: RANSAC for First Point Cloud

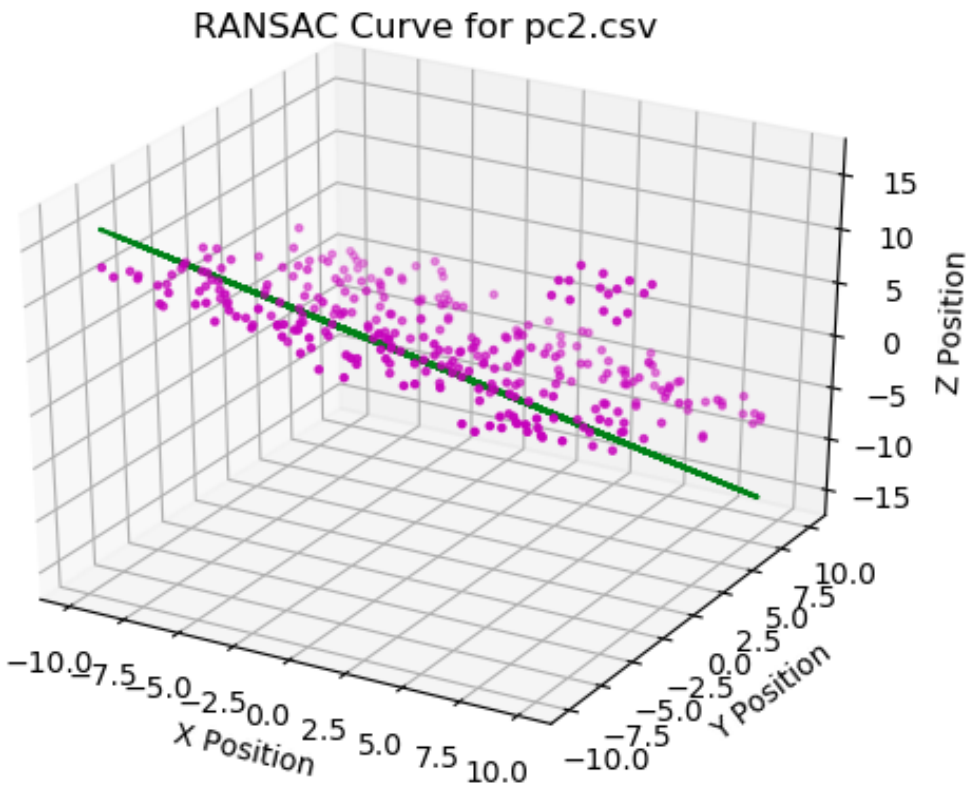

Figure 9: RANSAC for Second Point Cloud

2.5 Analysis

1. The Standard Least Square Fitting worked well with both noise data(outliers) and noiseless data based on the distribution of data in point clouds.

2. The Total Least Square Fitting hadn't much change for noiseless data but it produced a better plane for data containing noise in dealing with outliers.

3. RANSAC was not as good as TLS and SLS for noiseless data as the surface plane fit was not able to fit through data points as in TLS and SLS. But, with data containing noise the RANSAC provided a better plane fit by eliminating the outliers. The main problem with RANSAC is performing more number of tests and manual tuning due to its randomness.

3 References

1. Multiple Linear Regression : `https://www.had2know.org/academics/least-squares-plane-regressio html`

2. Lecture Notes-Perception for Autonomous Robots, Dr. Samer Charifa.

3. 3D RANSAC Algorithm for Lidar PCD Segmentation - `https://medium.com/@ajithraj_ gangadharan/3d-ransac-algorithm-for-lidar-pcd-segmentation-315d2a51351`

4. Quadratic Regression : `https://www.varsitytutors.com/hotmath/hotmath_help/topics/ quadratic-regression`