ENPM 673 Homework 1

Arjun Srinivasan Ambalam

February 13, 2020

1 Problem 1

Assume that you have a camera with a resolution of 5MP where the camera sensor is square shaped with a width of 14mm. It is also given that the focal length of the camera is 15mm.

- Compute the Field of View of the camera in the horizontal and vertical direction. [10 POINTS]
- Assuming you are detecting a square shaped object with width 5cm, placed at a distance of 20 meters from the camera, compute the minimum number of pixels that the object will occupy in the image. [10 POINTS]

1.1 Field of View Calculation

Since camera sensor is square shaped and also a single focal length is given,the Field of View (FOV) in Horizontal and Vertical directions are same.

$$\boxed{FOV = \ 2 * \phi} \tag{1}$$

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

$$\begin{aligned}
\phi &= \ \tan^{-1}\frac{14}{2 * 15} \\
\phi &= 25.0168^\circ \\
FOV &= 2 * 25.0168 \\
&= 50.036^\circ
\end{aligned}$$

where

- d is camera sensor width (d=14 mm)
- f is focal length (f=15 mm)

Therefore the Field of View of the camera (FOV) in horizontal and vertical direction is $\boxed{50.036^\circ}$

1.2 Minimum number of pixels occupying image

Given:

- Width of square shaped object (w) = 50 mm (5 cm)
- Distance of square shaped object from the camera (D) = 20000 mm (20 m)

Let the size of object in the image be 'x' mm. Then,

$$\boxed{\frac{x}{2f} = \frac{w}{2D}} \tag{3}$$

After substitution and solving we get,

$$\boxed{x = 0.0375\ mm} \tag{4}$$

$$5\ MP\ resolution\ for\ sensorarea = 14 \times 14\ mm^2 \tag{5}$$

$$For\ sensor\ area\ 0.0375 \times 0.0375\ mm^2 = 35.87 pixels \approx 36 pixels \tag{6}$$

Therefor minimum number of pixels occupying image is $\boxed{36}$

2 Problem 2

Two files of 2D data points are provided in the form of CSV files (Dataset1 and Dataset2). The data represents measurements of a projectile with different noise levels. Assuming that the projectile follows the equation of a parabola,

- Find the best method to fit a curve to the given data for each case. You have to plot the data and your best fit curve for each case. Submit your code along with the instructions to run it. [40 POINTS]
- Briefly explain all the steps of your solution and discuss why your choice of outlier rejection technique is best for that case. [20 POINTS]

2.1 Analysis of Curve Fitting Methods and Results

The problem of curve fitting is, given N discrete points in space, to find model parameters that can best characterize or generalize the trends prevailing between them. For this assignment, we are given with a set of 2D data points representing a projectile with different noise levels. In order to obtain the best model that can fit the given data points, We have discussed about Ordinary Least Squares (OLS), Total Least Squares (TLS), Least Squares with Regularization (LSR) and RANSAC (Random Sample Consensus) methodologies. Based on the results obtained from the above methods, the best fit for the problem is decided. Let us have a brief look at each of the above methods and their corresponding results.

Ordinary Least Squares (OLS):
Given the 2D projectile data $\{(X_1, Y_1), (X_2, Y_2), ..., (X_N, Y_N)\}$, we try to fit a quadratic equation $y = ax^2 + bx + c$ by reducing the sum of errors associated with the estimated model parameters and ground truth. This concept is mathematically represented by,

$$\boxed{E(a, b, c) = \sum_{n=1}^{N} (y - (ax^2 + bx + c))} \tag{7}$$

where,

- E(a, b, c) - sum of squared errors between the ground truth y and the estimate y obtained from our model parameters.

In order to find best fit for the above data we try to minimize the above error with respect to the model parameters. This can be easily represented using matrices.

$$\boxed{Y = XB} \tag{8}$$

where,

$$\boxed{Y = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_N \end{bmatrix} \quad X = \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ . & . & . \\ . & . & . \\ x_N^2 & x_N & 1 \end{bmatrix} B = \begin{bmatrix} a \\ b \\ c \end{bmatrix}} \tag{9}$$

$$\boxed{E(a,b,c) = (Y - XB)^2} \tag{10}$$

In order to minimize the above error,

$$\boxed{\frac{\partial E(a,b,c)}{\partial B} = 0} \tag{11}$$

On solving the above equation we obtain the following optimized model parameters,

$$\boxed{B = (X^T X)^{-1}(X^T Y)} \tag{12}$$

Algorithm 1: Least Squares Fitting

Result: $Y_{estimate}$
Load 2D data points;
Evaluate Y and X matrices;
Evaluate B matrix using $B = (X^T X)^{-1}(X^T Y)$;
Evaluate and Return $Y_{estimate}$;
Scatterplot original 2D data points and Plot the curve that fit the 2D data points

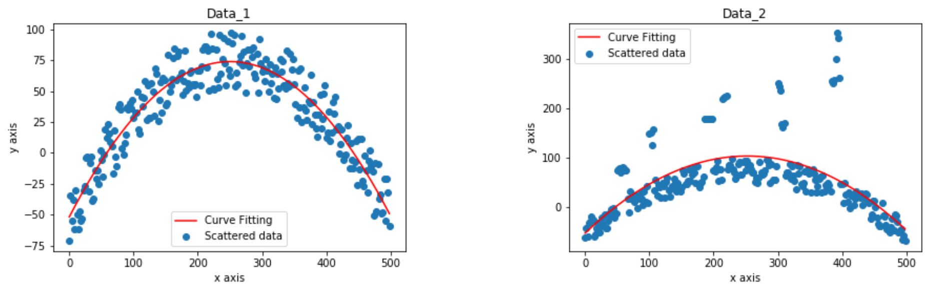

Figure 1: Least Squares Curve fitting for Dataset_1 and Dataset_2

Total Least Squares (TLS):
Given the 2D projectile data {(X_1, Y_1), (X_2, Y_2), ..., (X_N, Y_N)}, we try to fit a quadratic equation $y = ax^2 + bx + c$ by reducing the errors associated with both the dependent (y) and independent variables (x) corresponding to both the estimated model parameters and ground truth. This can be shown using matrices as follows,

$$\boxed{[Y + \tilde{Y}] = [X + \tilde{X}]B} \tag{13}$$

where,

- $\tilde{Y}$ - matrix consisting of errors associated with dependent variable
- $\tilde{X}$ - matrix consisting of errors associated with independent variable

Now the problem boils down to minimizing $\tilde{Y}$ and $\tilde{X}$ in order to fit a model to the data.

$$\boxed{min\| \ [\tilde{X} \ \tilde{Y}] \ \|^{\ 2} \ \text{ such that } \ [\mathrm{Y} + \tilde{Y}] = [\mathrm{X} + \tilde{X}]\mathrm{B}} \tag{14}$$

$$\boxed{\begin{bmatrix} X + \tilde{X} & X + \tilde{Y} \end{bmatrix} \begin{bmatrix} B \\ -1 \end{bmatrix} = \mathbf{0}_{\ \mathbf{m\ x\ 1}}} \tag{15}$$

A unique solution B can be found for above equation using Singular Value Decomposition (SVD).

$$\boxed{B = -V_{\mathrm{XY}} V^{\text{-}1}{}_{\mathrm{YY}}} \tag{16}$$

where,

- V $_{\mathrm{XY}}$ - first N elements of the (N + 1) $^{\mathrm{th}}$ column of the matrix of right singular vectors (V) of [X Y]
- V $_{\mathrm{YY}}$ - (N + 1) $^{\mathrm{th}}$ element of the (N + 1) $^{\mathrm{th}}$ column of V

Algorithm 2: Least Squares Fitting

Result: $\mathrm{Y}_{\mathrm{estimate}}$
Load 2D data points;
Evaluate Y and X matrices;
Concatenated Matrix = [X Y];
[U S V] = SVD(Concatenated Matrix);
Get V $_{\mathrm{XY}}$ (Take the block of V consisting of the first N rows and the N + 1 to last column);
Get V $_{\mathrm{YY}}$ (Take the bottom-right block of V);
Evaluate B matrix using $B = -V_{\mathrm{XY}} V^{\text{-}1}{}_{\mathrm{YY}}$;
Evaluate and Return $\mathrm{Y}_{\mathrm{estimate}}$;
Scatterplot original 2D data points and Plot the curve that fit the 2D data points

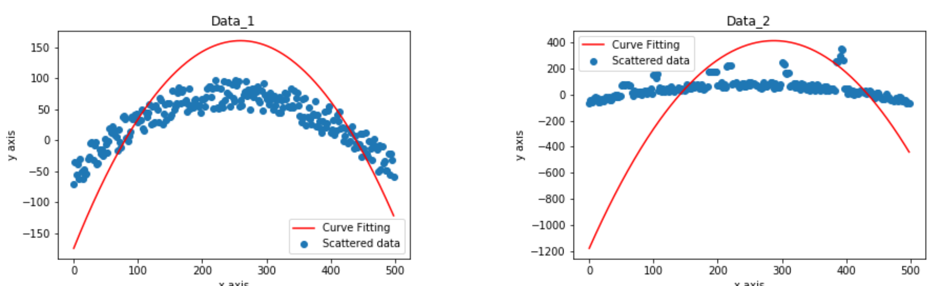

Figure 2: Total Least Squares for Dataset_1 and Dataset_2

Least Squares with Regularization (LSR):
Given the 2D projectile data {(X_1, Y_1), (X_2, Y_2), ..., (X_N, Y_N)}, we try to fit a quadratic equation $y = ax^2 + bx + c$ by reducing the sum of squared errors associated with the estimated model parameters and ground truth with an addition constraint called regularization parameter (λ). If there are more variables than observations in a linear system, the least squares problem is impossible to fit, because there are infinitely many solutions. With regularized least squares we can still get a uniquely determined solution. Basically the aim of regularization parameter is to penalize the model so that it improves the generalizability of the model. Mathematically, the model parameters can be obtained in a similar fashion to that of Least Squares with just an addition of λ.

$$\boxed{B = (X^T X + \lambda I)^{\text{-}1} (X^T Y)} \tag{17}$$

Algorithm 3: Least Squares Fitting with Regularization

Result: $\mathrm{Y}_{\mathrm{estimate}}$

Load 2D data points;

Initialize regularization parameter (λ);

Evaluate Y and X matrices;

Evaluate B matrix using $B = (X^T X + \lambda I)^{-1}(X^T Y)$;

Evaluate and Return $\mathrm{Y}_{\mathrm{estimate}}$;

Scatterplot original 2D data points and Plot the curve that fit the 2D data points

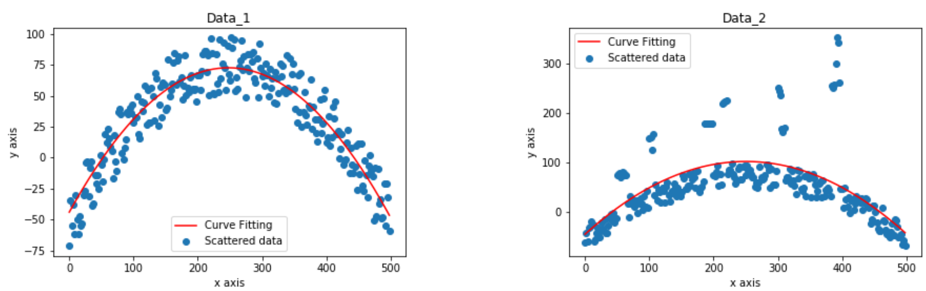

Figure 3: Least Squares Curve fitting using Regularization for Dataset_1 and Dataset_2

Random Sample And Consensus (RANSAC):
RANSAC is an iterative algorithm that estimates the best possible solution of a certain model using the given set of data points that is contaminated by large number of outliers. RANSAC basically comprises of two steps:

- Minimal Sample Sets were randomly selected from the input dataset (for a quadratic curve minimum number of points required and the model parameters are computed using only those elements; in order to compute model parameters we can use least squares or total least squares or any other optimization methodologies.
- We then evaluate the model estimated by checking it with rest of the dataset and classify the dataset into inliers and outliers based on a defined threshold.

RANSAC is a parallel hypothesis and test problem i.e. it simultaneously generates and tests multiple, unique hypotheses. The major set of hyperparameters in RANSAC are the threshold, the minimum number of inliers that the model should consist and the number of iterations. Setting threshold requires knowledge about the probability distribution of the inlier distances in the dataset. Also the number of iterations must be chosen carefully in order to reduce the computational complexity. If we know the probability for a sample set to be an inlier p and the probability for it to be outlier w = (1 - p) then we need to perform atleast N subset selections to have a profitable inlier ratio.

$$\boxed{(1 - w^s)^N = (1 - p)} \tag{18}$$

where,

- w - probability of an outlier
- p - probability of an inlier
- s - Minimal Sample Set
- N - Number of iterations or Number of subset selections

$$\boxed{N = \frac{\log(1-p)}{\log(1-w^s)}} \tag{19}$$

Algorithm 5: Random Sample And Consensus (RANSAC)

```
Result: Best Model Parameters
Obtain minimum number of samples, minimum number of inlier points, threshold, iterations;
Initialize best model = None, best inliers = 0, best error = infinity, i = 0;
while i ≤ iterations do
    Choose 3 random points from the dataset;
    Also obtain the remaining points from the dataset;
    Fit a curve for the 3 random points using least squares method;
    Estimate the y values for the remaining points;
    Find the error per point;
    Compute the number of inliers by comparing the error per point with threshold;
    if number of inliers ≥ minimum number of inlier points then
        Total number of inliers = number of inliers + minimum number of samples;
        Fit a curve again for the set containing Total number of inliers;
        Estimate the y values for the remaining points;
        Find the error per point and calculate the mean of the error;
        if current error ≤ best error then
            best model = current model;
            best error = current error;
            best inliers = total inliers;
        end
    end
    i = i + 1;
end
```

Now that we have discussed briefly about the methods that were used for fitting a curve for the given 2D dataset, let us analyse the results corresponding to them. It is clear from the plots that variants of Least Square fitting work well for dataset 1 than dataset 2. This is quite obvious because dataset 1 has no or less outliers compared to that of dataset 2. Also, dataset 1 resembles as if a standard gaussian noise was added to the measured y variables and this pattern is being exploited by the Least Square variants. On more careful observation we can see that Total Least Squares variant does not perform well for both the datasets. It is because both the datasets weren't mean centered. On the other hand, we can see that Least Squares with Regularization perform well for both datasets compared to that of other Least Sqaures variants by controlling the regularization parameter (λ). But in general the trouble with outliers in the Least Squares variants is because the Least Squares method has knowledge only about mean and squared differences from the mean of the sample. Outliers distort these means and on squaring the differences from the distorted mean will only enhance the distortion, thereby failing to identify the best model in case of outliers. In order to overcome this problem, we performed RANSAC. RANSAC with its parallel hypothesis generation and testing limits the distortion effect of outliers on the dataset thereby performing outlier rejection. This is evident in Fig. 4, where RANSAC was tested on both datasets by setting different values to the hyperparamters threshold, minimum number of inliers, and number of iterations. But RANSAC too has it's disadvantages that, it is not repeatable and therefore generates multiple curves when used multiple times.

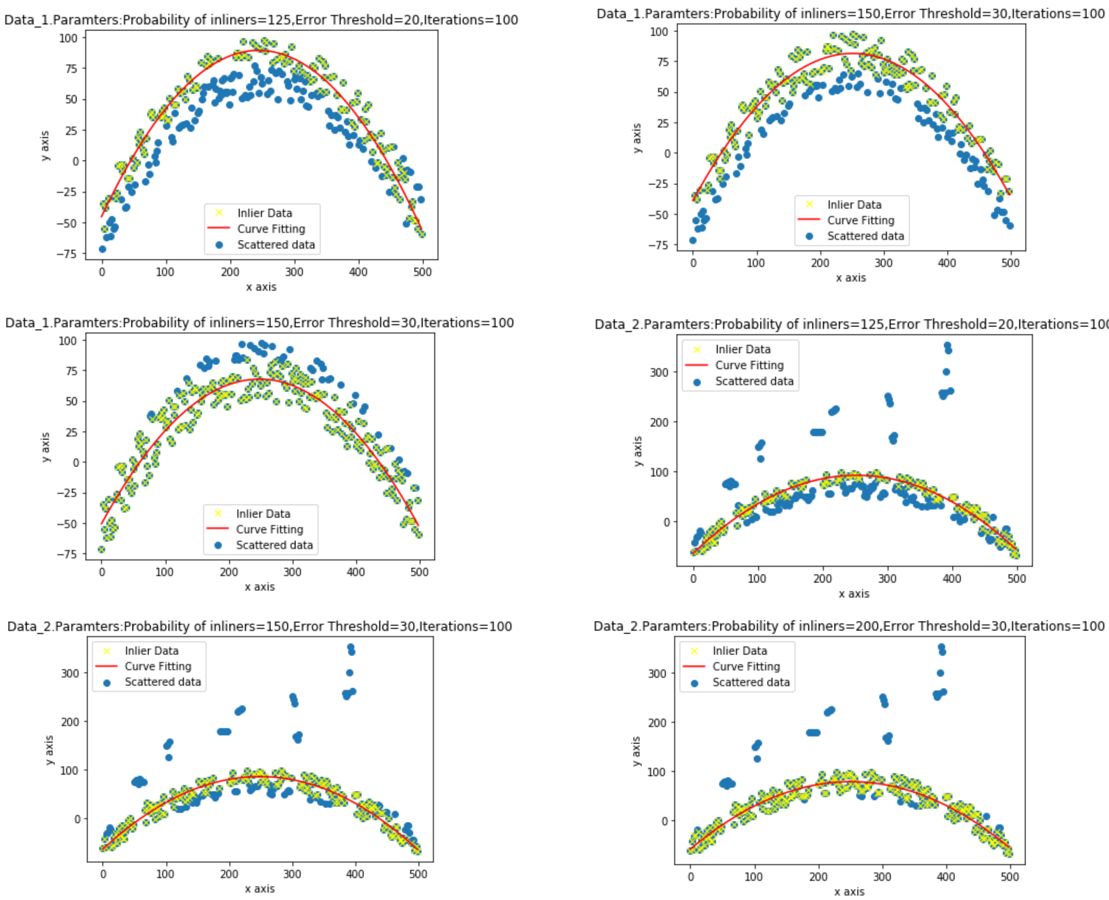

Figure 4: RANSAC curve fitting for different parameters

3 Homography in Computer Vision

$$A = \begin{bmatrix} -x1 & -y1 & -1 & 0 & 0 & 0 & x1*xp1 & y1*xp1 & xp1 \\ 0 & 0 & 0 & -x1 & -y1 & -1 & x1*yp1 & y1*yp1 & yp1 \\ -x2 & -y2 & -1 & 0 & 0 & 0 & x2*xp2 & y2*xp2 & xp2 \\ 0 & 0 & 0 & -x2 & -y2 & -1 & x2*yp2 & y2*yp2 & yp2 \\ -x3 & -y3 & -1 & 0 & 0 & 0 & x3*xp3 & y3*xp3 & xp3 \\ 0 & 0 & 0 & -x3 & -y3 & -1 & x3*yp3 & y3*yp3 & yp3 \\ -x4 & -y4 & -1 & 0 & 0 & 0 & x4*xp4 & y4*xp4 & xp4 \\ 0 & 0 & 0 & -x4 & -y4 & -1 & x4*yp4 & y4*yp4 & yp4 \end{bmatrix}$$

where,

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

Substituting the values, the A matrix becomes

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

Algorithm 6: Homography matrix calculation

Result: H (Homography Matrix)
Load A matrix;
[U S V] = SVD(A) (compute Singular Value Decomposition of A matrix);
Extract the last column of V matrix and reshape it into a square matrix, in our case a 3x3 matrix;

Algorithm 7: Singular Value Decomposition (SVD) Calculation

Result: U, S, V
Load A matrix;
Find eigen values and eigen vectors of A^{T}A;
Sort the eigen values and eigen vectors in descending order;
Form V matrix by placing the sorted eigen vectors of A^{T}A as columns;
Take square root of the sorted eigen values of A^{T}A and place them along the diagonal of a matrix and form S matrix;
Find eigen values and eigen vectors of AA^{T};
Sort the eigen values and eigen vectors in descending order;
Form V matrix by placing the sorted eigen vectors of AA^{T} as columns;

$$\boxed{AH = 0} \tag{20}$$

$$\boxed{A = U\Sigma V^T} \tag{21}$$

where,

- A - Matrix of size 8× 9
- U - Matrix of size 8× 8 comprising of orthogonal eigen vectors of AA^{T}
- S - Diagonal matrix diag($\sigma_{\,1}...\sigma_{\,9}$), where $\sigma_{\,\mathrm{i}} = \sqrt{\lambda_{\mathrm{i}}}$
- $\lambda_{\,\mathrm{i}}$ - Eigenvalues of A^{T}A
- V - Matrix of size 9× 9 comprising of orthogonal eigen vectors of A^{T}A

The values of U, S, and V matrix are given below,

$$V = \begin{bmatrix} 0.0028 & 0.0031 & -0.2464 & -0.1586 & -0.1752 & 0.1767 & 0.9137 & -0.1203 & 0.0531 \\ 0.0024 & -0.0013 & -0.3770 & 0.1766 & 0.6895 & 0.5903 & -0.0529 & -0.0022 & -0.0049 \\ 0.0000 & 0.0000 & -0.0024 & -0.0037 & 0.0052 & 0.0075 & 0.0660 & 0.7860 & 0.6146 \\ 0.0011 & 0.0012 & 0.6612 & 0.3412 & 0.5017 & -0.2325 & 0.3721 & -0.0426 & 0.0177 \\ 0.0016 & -0.0029 & 0.5743 & -0.0710 & -0.3145 & 0.7499 & -0.0620 & 0.0046 & -0.0039 \\ 0.0000 & 0.0000 & 0.0058 & -0.0022 & 0.0029 & -0.0057 & -0.1225 & -0.6049 & 0.7868 \\ -0.6961 & -0.7180 & -0.0001 & -0.0038 & 0.0025 & -0.0002 & 0.0044 & -0.0006 & 0.0002 \\ -0.7180 & 0.6961 & 0.0016 & -0.0038 & 0.0025 & 0.0037 & -0.0006 & -0.0000 & 0.0000 \\ -0.0062 & 0.0000 & -0.1735 & 0.9067 & -0.3783 & 0.0622 & 0.0252 & -0.0025 & 0.0076 \end{bmatrix} \tag{22}$$

$$S = \begin{bmatrix} 60214.89 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 31824.52 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 260.89 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 186.22 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 145.60 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 60.87 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3.87 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.81 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{23}$$

$$U = \begin{bmatrix} -0.0118 & -0.0003 & -0.0516 & 0.4661 & -0.2603 & 0.0678 & 0.0108 & -0.8411 & 0 \\ -0.0118 & -0.0003 & -0.0872 & 0.4594 & -0.2491 & 0.0886 & 0.7655 & 0.3542 & 0 \\ -0.3587 & -0.6549 & 0.0135 & 0.4651 & 0.1701 & -0.2936 & -0.2784 & 0.1823 & 0 \\ -0.1435 & -0.2620 & -0.4454 & -0.1361 & -0.5008 & 0.5875 & -0.2731 & 0.1529 & 0 \\ -0.7750 & -0.0227 & 0.4085 & -0.2849 & 0.0320 & 0.2352 & 0.2627 & -0.1597 & 0 \\ -0.2818 & -0.0082 & -0.6922 & -0.3159 & 0.0114 & -0.5019 & 0.2466 & -0.1696 & 0 \\ -0.1846 & 0.3168 & 0.2485 & 0.0347 & -0.6983 & -0.4673 & -0.2524 & 0.1816 & 0 \\ -0.3693 & 0.6336 & -0.2889 & 0.3933 & 0.3189 & 0.1750 & -0.2614 & 0.1526 & 0 \end{bmatrix} \tag{24}$$

The Homography matrix (H) is given below,

$$H = \begin{bmatrix} -0.0531 & 0.000491 & -0.614 \\ -0.0177 & 0.00393 & -0.786 \\ -0.000236 & 0.0000491 & -0.00762 \end{bmatrix} \tag{25}$$