ENPM673 – Perception for Autonomous Robots
Homework 1

Kavyashree Devadiga — 117398045

February 15, 2022

1 Problem 1: [20 Points]

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 25mm.

1.1 Compute the Field of View of the camera in the horizontal and vertical direction.

The field of view of a camera is the maximum area covered by the image captured and depends on the camera sensor size and it's focal length.
Given:

- Camera sensor width : 14mm
- Camera sensor height: 14mm
- Focal length (f) : 25mm

We can calculate the horizontal and vertical field of view as follows:

1. Horizontal Field of view (HFOV)

$$HFOV = 2\tan^{-1}\frac{0.5f}{width}$$
$$HFOV = 2\tan^{-1}\frac{0.5\times 14}{25}$$
$$HFOV = 0.546^{\circ} rad \approx 31.28^{\circ}(degrees)$$

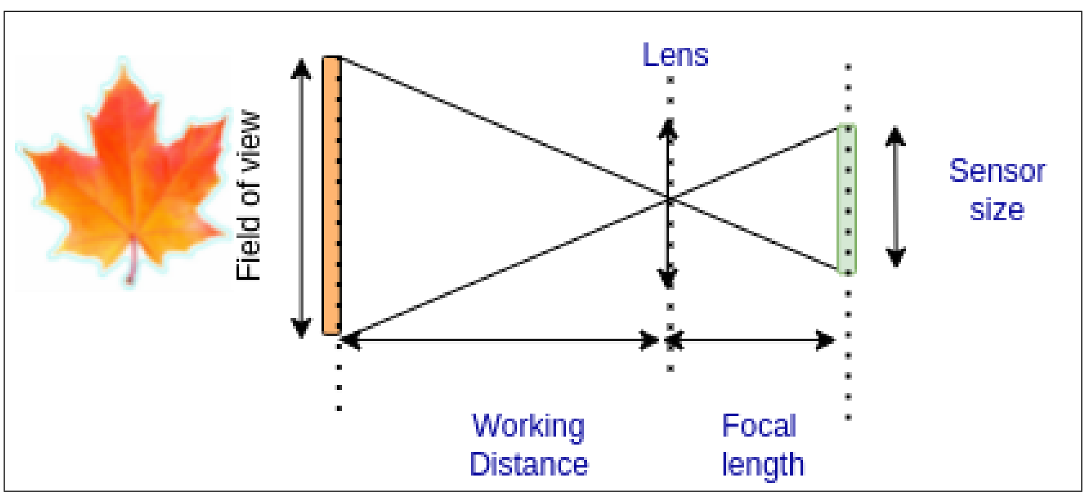

Figure 1: System design

2. Vertical Field of view (VFOV)

$$VFOV = 2\tan^{-1}\frac{0.5f}{height}$$

$$VFOV = 2\tan^{-1}\frac{0.5 \times 14}{25}$$

$$VFOV = 0.546° rad \approx 31.28° (degrees)$$

Here, the width and height of camera sensor is same, thus HFOV and VFOV have same value.

1.2 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.

Given:

- Width of object : 5cm = 50mm
- Distance of Object : 20m = 200 x 10^2mm
- Camera resolution : 5MP

$$sensor\ area = height \times width = 14 \times 14 = 196mm^2 \tag{1}$$

$$\frac{height\ of\ object}{distance\ of\ object} = \frac{height\ of\ image}{focal\ length}$$

$$\frac{50}{200 \times 10^2} = \frac{height\ of\ image}{25}$$

$$height\ of\ image = 6.25 \times 10^{-2} mm$$

$$image\ area = (6.5 \times 10^{-2}) \times (6.5 \times 10^-2) mm^2$$

$$number\ of\ pixels = image\ area \times (\frac{resolution}{sensor\ area}) \tag{2}$$

$$number\ of\ pixels = (6.5 \times 10^{-2}) \times (6.5 \times 10^{-2}) \times \frac{(5 \times 10^6)}{196} = 99.64 \approx 100$$

2 Problem 2: [30 Points] Standard Least Squares to fit curves to the given videos

A ball is thrown against a white background and a camera sensor is used to track its trajectory. We have a near perfect sensor tracking the ball in video1 and the second sensor is faulty and tracks the ball as shown in video2. Clearly, there is no noise added to the first video whereas there is significant noise in the second video. Assuming that the trajectory of the ball follows the equation of a parabola: Use Standard Least Squares to fit curves to the given videos in 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.

2.1 Solution

The solution is divided in two parts: Part 1 is reading video frame by frame and saving ball coordinates in a csv file, and Part 2 is plotting data from csv file, computing standard least square and visualizing the results in the plot

2.2 Reading data

The process can be describes as follows:

1. Open CSV file in write mode
2. Load the video
3. Read video frame by frame
4. Create a mask with lower and upper bound values for detecting red object
5. For each frame,find contours using mask and calculate the mid coordinates of the object
6. Save coordinates to the CSV file

The code and output can be found the the jupyter notebook here

2.3 Computing Standard least square

The equation of parabola is $y = a + bx + cx^2$. Here, we need to find the variables a, b, c such that the error is minimum. The error function can be written as:

$$\Pi = \sum_{i=1}^{n} \left[y_i - f\left(x_i\right)\right]^2 = \sum_{i=1}^{n} \left[y_i - \left(a + bx_i + cx_i^2\right)\right]^2$$

To minimize the error, the unknown variables must have zero derivatives. Hence:

$$\sum_{i=1}^{n} y_i = a + b\sum_{i=1}^{n} x_i + c\sum_{i=1}^{n} x_i^2.$$

$$\sum_{i=1}^{n} x_i y_i = a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 + c\sum_{i=1}^{n} x_i^3$$

$$\sum_{i=1}^{n} x_i^2 y_i = a\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i^3 + c\sum_{i=1}^{n} x_i^4$$

Next step is to solve the above three linear equations and get values for a, b, c. The code and result can be found here

3 Problem 3: [30 Points] RANSAC

In the above problem, we used the least squares method to fit a curve. However, if the data is scattered, this might not be the best choice for curve fitting. In this problem, you are given data for health insurance costs based on the person's age. There are other fields as well, but you have to fit a line only for age and insurance cost data. The data is available here.

3.1 Covariance matrix, its eigenvalues and eigenvectors

Plot the eigenvectors on the same graph as the data.

3.2 Fit a line to the data using linear least square method, total least square method and RANSAC

Plot the result for each method and explain drawbacks/advantages for each.

3.3 Choice of outlier rejection technique for each case

4 Problem 4: [20 Points] Homography and Singular Value Decomposition

Given 4 corresponding points on the two different planes, the homography between them is computed using the following system of equations Ax = 0, where A and x is given by

$$A = \begin{bmatrix} -x1 & -y1 & -1 & 0 & 0 & 0 & x1*xp1 & y1*xp1 & xp1 \\ 0 & 0 & 0 & -x1 & -y1 & -1 & x1*yp1 & y1*yp1 & yp1 \\ -x2 & -y2 & -1 & 0 & 0 & 0 & x2*xp2 & y2*xp2 & xp2 \\ 0 & 0 & 0 & -x2 & -y2 & -1 & x2*yp2 & y2*yp2 & yp2 \\ -x3 & -y3 & -1 & 0 & 0 & 0 & x3*xp3 & y3*xp3 & xp3 \\ 0 & 0 & 0 & -x3 & -y3 & -1 & x3*yp3 & y3*yp3 & yp3 \\ -x4 & -y4 & -1 & 0 & 0 & 0 & x4*xp4 & y4*xp4 & xp4 \\ 0 & 0 & 0 & -x4 & -y4 & -1 & x4*yp4 & y4*yp4 & yp4 \end{bmatrix}, x = \begin{bmatrix} H_{11} \\ H_{12} \\ H_{13} \\ H_{21} \\ H_{22} \\ H_{23} \\ H_{31} \\ H_{32} \\ H_{33} \end{bmatrix}$$

- Given points:

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

- Homography matrix:

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix}$$

We can represent any matrix A as:

$$\underbrace{\mathbf{A}}_{M\times N} = \underbrace{\mathbf{U}}_{M\times M} \times \underbrace{\mathbf{\Sigma}}_{M\times N} \times \underbrace{\mathbf{V}^{\mathrm{T}}}_{N\times N} \quad (3)$$

Where, U is an orthogonal matrix. Σ is a termed the singular values. V is an orthogonal matrix. The last column of V is the required homography matrix.

$$\begin{aligned} A^\top A &= \left(U\Sigma V^\top\right)^\top \left(U\Sigma V^\top\right) \\ &= V\Sigma^\top U^\top U\Sigma V^\top \\ &= V\Sigma^\top \Sigma V^\top \ (\ U \text{ is orthogonal }) \end{aligned}$$

Similarly,

$$\begin{aligned} AA^\top &= \left(U\Sigma V^\top\right)\left(U\Sigma V^\top\right)^\top \\ &= U\Sigma V^\top V\Sigma^\top U^\top \\ &= U\Sigma\Sigma^\top U^\top \ (\ V \text{ is orthogonal }) \end{aligned}$$

4.1 Compute the Singular Value Decomposition (SVD) for the matrix A

The eigenvectors of $A^T A$ make up the columns of V , the eigenvectors of AA^T make up the columns of U

1. Matrix A:

$$
A = \begin{pmatrix}
-5 & -5 & -1 & 0 & 0 & 0 & 500 & 500 & 100 \\
0 & 0 & 0 & -5 & -5 & -1 & 500 & 500 & 100 \\
-150 & -5 & -1 & 0 & 0 & 0 & 30000 & 1000 & 200 \\
0 & 0 & 0 & -150 & -5 & -1 & 12000 & 400 & 80 \\
-150 & -150 & -1 & 0 & 0 & 0 & 33000 & 33000 & 220 \\
0 & 0 & 0 & -150 & -150 & -1 & 12000 & 12000 & 80 \\
-5 & -150 & -1 & 0 & 0 & 0 & 500 & 15000 & 100 \\
0 & 0 & 0 & -5 & -150 & -1 & 1000 & 30000 & 200
\end{pmatrix} \tag{4}
$$

2. Compute $A^T A$

$$
A^T A = \begin{pmatrix}
45050 & 24025 & 310 & 0 & 0 & 0 & -9455000 & -5177500 & -64000 \\
24025 & 45050 & 310 & 0 & 0 & 0 & -5177500 & -7207500 & -49500 \\
310 & 310 & 4 & 0 & 0 & 0 & -64000 & -49500 & -620 \\
0 & 0 & 0 & 45050 & 24025 & 310 & -3607500 & -2012500 & -25500 \\
0 & 0 & 0 & 24025 & 45050 & 310 & -2012500 & -6304500 & -42900 \\
0 & 0 & 0 & 310 & 310 & 4 & -25500 & -42900 & -460 \\
-9455000 & -5177500 & -64000 & -3607500 & -2012500 & -25500 & 2278750000 & 1305800000 & 15530000 \\
-5177500 & -7207500 & -49500 & -2012500 & -6304500 & -42900 & 1305800000 & 2359660000 & 16052000 \\
-64000 & -49500 & -620 & -25500 & -42900 & -460 & 15530000 & 16052000 & 171200
\end{pmatrix}
$$

3. Similarly compute AA^T

4. For computing V, get eigen values and eigen vectors of $A^T A$ and arrange them in descending order with highest eigen value first. The transpose of computed V Matrix is as follows

$$
V^T = \begin{bmatrix}
2.84043894e-03 & 2.42121739e-03 & 2.20891154e-05 & 1.09109680e-03 & 1.63479471e-03 & 1.33908907e-05 & -6.96053715e-01 & -7.17950893e-01 & -6.16016024e-03 \\
3.14430147e-03 & -1.28321626e-03 & 1.13495064e-05 & 1.17416448e-03 & -2.90636016e-03 & -1.14077892e-05 & -7.17961695e-01 & 6.96067270e-01 & 2.29933343e-05 \\
-2.46384735e-01 & -3.77000733e-01 & -2.37217168e-03 & 6.61240940e-01 & 5.74279813e-01 & 5.80190908e-03 & -7.57487349e-05 & 1.62813209e-03 & -1.73453679e-01 \\
-1.58554932e-01 & 1.76600215e-01 & -3.65660431e-03 & 3.41172744e-01 & -7.10405325e-02 & -2.15181510e-03 & -3.79564118e-03 & -3.77529395e-03 & 9.06741660e-01 \\
-1.75245114e-01 & 6.89508147e-01 & 5.19584361e-03 & 5.01749740e-01 & -3.14549320e-01 & 2.88147957e-03 & 2.50334194e-03 & 2.49754245e-03 & -3.78319687e-01 \\
1.76705635e-01 & 5.90273326e-01 & 7.52000216e-03 & -2.32499365e-01 & 7.49883366e-01 & -5.73504703e-03 & -1.50223526e-04 & 3.65599795e-03 & 6.21986452e-02 \\
9.13738625e-01 & -5.29344506e-02 & 6.59901594e-02 & 3.72052358e-01 & -6.19835401e-02 & -1.22489909e-01 & 4.37766998e-03 & -6.00114914e-04 & 2.52338778e-02 \\
-1.20261073e-01 & -2.23230961e-03 & 7.85970681e-01 & -4.25903576e-02 & 4.58785876e-03 & -6.04930528e-01 & -5.55196293e-04 & 3.48250733e-05 & -2.47794677e-03 \\
5.31056350e-02 & -4.91718843e-03 & 6.14648552e-01 & 1.77018784e-02 & -3.93375075e-03 & 7.86750146e-01 & 2.36025045e-04 & -4.91718843e-05 & 7.62164204e-03
\end{bmatrix}
$$

5. The last row of V^T is the required **Homography Matrix (H)**

$$
\mathrm{H} = \begin{bmatrix}
5.31056350e-02 & -4.91718844e-03 & 6.14648552e-01 \\
1.77018784e-02 & -3.93375075e-03 & 7.86750146e-01 \\
2.36025045e-04 & -4.91718843e-05 & 7.62164205e-03
\end{bmatrix} \tag{5}
$$

.. **Answer**

6. In the same way, for computing U, get eigen values and eigen vectors of AA^T and arrange them in descending order with highest eigen value first. The computed U Matrix is as follows

$$
U = \begin{bmatrix}
1.17519867e-02 & 3.44207228e-04 & -5.15532162e-02 & -4.66128587e-01 & -2.60345896e-01 & -6.78428560e-02 & 1.08122936e-02 & -8.41087769e-01 \\
1.17517760e-02 & 3.43641967e-04 & -8.72103737e-02 & -4.59351955e-01 & -2.49098952e-01 & -8.85591890e-02 & 7.65455993e-01 & 3.54169471e-01 \\
3.58735699e-01 & 6.54942912e-01 & 1.34538659e-02 & -4.65084492e-01 & 1.70101644e-01 & 2.93617516e-01 & -2.78385485e-01 & 1.82289869e-01 \\
1.43494223e-01 & 2.61976394e-01 & -4.45383120e-01 & 1.36060221e-01 & -5.00795526e-01 & -5.87488150e-01 & -2.73099287e-01 & 1.52897236e-01 \\
7.74962678e-01 & 2.27117371e-02 & 4.08516159e-01 & 2.84937362e-01 & 3.19642679e-02 & -2.35211438e-01 & 2.62688692e-01 & -1.59658351e-01 \\
2.81806634e-01 & 8.24745878e-03 & -6.92167142e-01 & 3.15915567e-01 & 1.14149714e-02 & 5.01908806e-01 & 2.46628160e-01 & -1.69560615e-01 \\
1.84643411e-01 & -3.16806256e-01 & 2.48466337e-01 & -3.46544961e-02 & -6.98268275e-01 & 4.67261587e-01 & -2.52393736e-01 & 1.81630339e-01 \\
3.69278450e-01 & -6.33614920e-01 & -2.88917222e-01 & -3.93333286e-01 & 3.18917542e-01 & -1.75016528e-01 & -2.61429000e-01 & 1.52633610e-01
\end{bmatrix}
$$

7. Compute Sigma (Σ)
Matrix Σ is computed by taking square roots of eigenvalues from AA^T or $A^T A$. The eigen values are arranged as diagonal elements in descending order. All other elements are set as zero.

$$
\Sigma = \begin{bmatrix}
6.02148954e+04 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 3.18245207e+04 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 2.60893068e+02 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 1.86219278e+02 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 1.45606434e+02 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 6.08809411e+01 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 3.89873639e+00 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 8.10241300e-01 & 0.0
\end{bmatrix}
$$