Homework 1
ENPM673

Sakshi Kakde
M. Eng. Robotics
University of Maryland
College Park, MD, 20742
Email: sakshi@umd.edu

I. Problem 1

A. Part 1

The relation between field of view, focal length and sensor height/width is given by the following relation:

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

Where, ϕ is the field of view, f is the focal length and d is the sensor dimension. As per the question,

$$f = 15mm$$

$$d = 14mm$$

Since the sensor shape is square, the horizontal and vertical FoV shall be the same. Substituting the values,

$$\phi = 2 \times tan^{-1}(\frac{15}{2 \times 14})$$

$$\phi = 57.357^o$$

The horizontal and vertical fields of view are 57.357^o.

B. Part 2

Given the object height and the object distance from the camera, the image height can be calculated using the following relation:

$$\frac{h_o}{d_o} = \frac{h_i}{f}$$

Where,
(h_o is actual height of the object, (d_o is camera to object distant (h_i is the image height, and (f is the focal length of camera. Substituting the values(all values converted to mm), we get

$$\frac{5 \times 10^1}{20 \times 10^3} = \frac{h_i}{15}$$

$$h_i = 37.5 \times 10^{-3}mm$$

Since the object and the sensor are square, the height(h_i) and width(w_i) on the image will be equal to $37.5 \times 10^{-3}mm$.
The area covered by the image will be ($h_i \times w_i$). Let the sensor height and width is denoted by h_s and w_s. Let R denote resolution of the camera and P be the number of pixels covered by the image of the square object. Then, by ratio and proportion,

$$\frac{R}{Areaofsensor} = \frac{P}{Areaofimage}$$

$$\frac{R}{h_s \times w_s} = \frac{P}{h_i \times w_i}$$

$$\frac{9 \times 10^6}{14 \times 14} = \frac{P}{37 \times 10^{-3} \times 37.5 \times 10^{-3}}$$

$$P = 61.12pixels$$

Note: 1 MP is assumed as 10^6 *pixels. All dimensions are converted to* mm.

The image of the square object will occupy approximately 62 pixels.

II. Problem 2

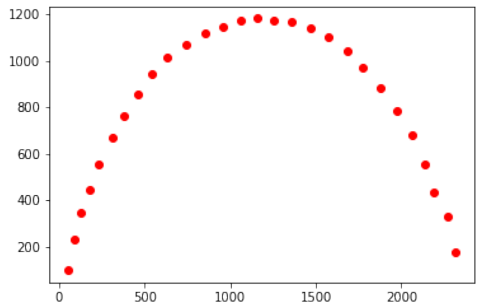
In this section, three curve fitting techniques: Least Square Method, Total Least Square method and RANSAC have been implemented.

A. Extracting data points

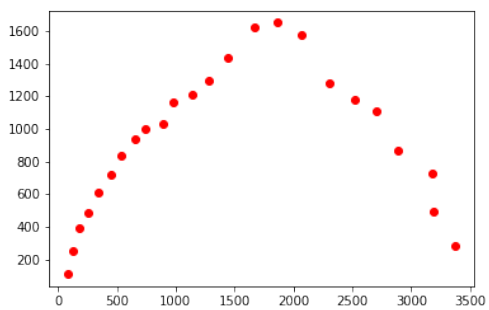
1) Read the video file from the folder.
2) Convert it to gray scale.
3) Set appropriate thresholds to get a binary output so that the algorithm is color independent.
4) Extract the black pixels, i.e. the pixels occupied by the ball.
5) Find the location of top most and bottom most pixel.
6) Take average of the top and bottom points to get the centre of the ball.
7) These centre points are used to fit a curve, which is assumed as a parabola.

B. Converting the points to a suitable reference frame

Since the image co-ordinate system is different than the one used by NumPy, a transformation is used to converted the image co-ordinates to a suitable frame. Refer fig. 1 for the extracted data points.

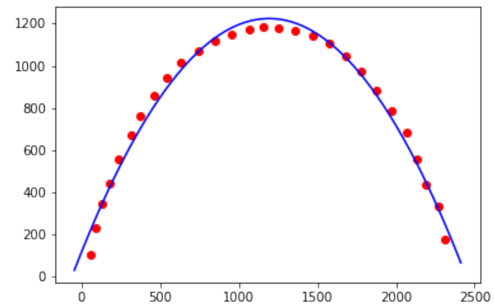


(a) from video 1(without noise)

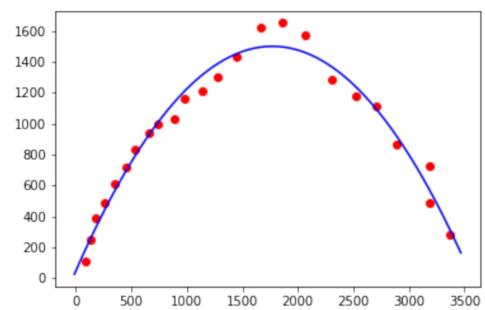


(b) from video 2(with noise)

Fig. 1. Data points extracted from videos.



(a) Video 1(without noise)



(b) Video 2(with noise)

Fig. 2. Least Square Curve Fitting.

C. *Curve Fitting*

1) Least Square Method(LS): The equation of parabola is given by,

$$y = ax^2 + bx + c$$

The error function can be written as,

$$E = \Sigma_n (y - ax^2 - bx - c)^2$$

Let $X = [x^2, x, 1]$, $Y = y$ and $D = [a, b, c]^T$. Then,

$$E = \|(Y - XD)\|^2$$

$$E = (Y - XD)^T (Y - XD)$$

$$E = Y^T Y - 2(XD)^T Y + (XD)^T XD$$

Differentiating wrt D,

$$\frac{\delta E}{\delta D} = -2X^T Y + 2X^T XD$$

Since $\frac{\delta E}{\delta D} = 0$,

$$-2X^T Y + 2X^T XD = 0$$

$$D = (X^T X)^{-1} X^T Y$$

Refer fig 2 for the obtained curves.

2) Total Least Square Method(TLS): The equation of a parabola cab be written as,

$$ax^2 + bx + cy = d$$

The error function cab be written as,

$$E = \Sigma_n (ax^2 + bx + cy - d)^2$$

Since $\frac{\delta E}{\delta D} = 0$,

$$2\Sigma_n (ax^2 + bx + cy - d) = 0$$

$$nd = \Sigma_n (ax^2 + bx + cy$$

$$d = \frac{\Sigma_n (ax^2 + bx + cy}{n}$$

$$d = a\overline{x^2} + b\bar{x} + c\bar{y}$$

Where, $\overline{x^2}, \bar{x}, \bar{y}$ represent the mean over n data points. Substituting this value of d, we get the error function as,

$$E = \Sigma_n (a(x^2 - \overline{x^2}) + b(x - \bar{x}) + c(y - \bar{y}))^2$$

Let $U = [x^2 - \overline{x^2}, x - \bar{x}, y - \bar{y}]$ and $N = [a, b, c]$, then,

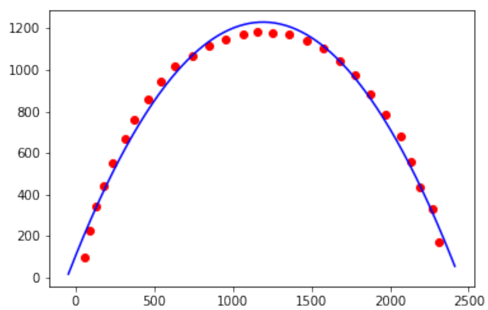
$$E = (UN)^T (UN)$$

$$\frac{\delta E}{\delta D} = 2U^T UN$$

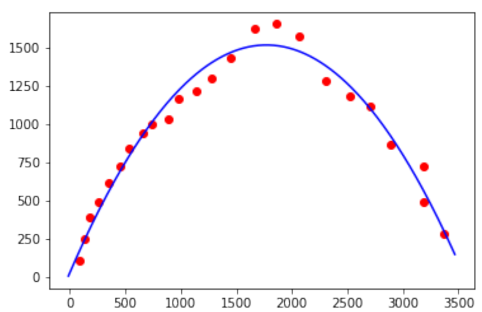
Therefore,

$$U^T UN = 0$$

This is a homogeneous equation. Let $U^T U = A$. Then, the equation becomes $AN = 0$. To find the solution of this homogeneous equation, we can use SVD. Refer fig 3 for the obtained curves.



(a) Video 1(without noise)



(b) Video 2(with noise)

Fig. 3. Total Least Square Curve Fitting.

3) Random Sample Consensus(RANSAC):

1) Select three random points from the data points.
2) Try to fit a curve using the TLS cure fitting method discussed earlier using those three points.
3) Calculate the error for all the data points wrt to the curve obtained. The error function is,

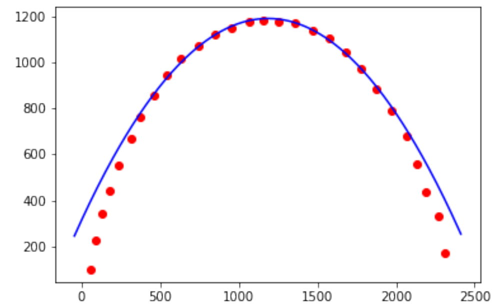
$$E = (ax^2 + bx + cy - d)^2$$

4) Select a threshold error value and reject all the point above the threshold value.
5) Count the number of points below the threshold value.
6) Continue the process for N iterations and select the curve where maximum points are below the threshold error value.

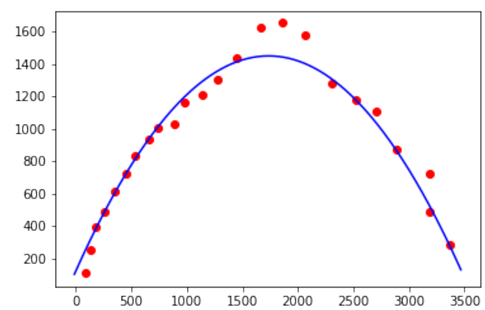
The number of iterations(N) can be calculated using the following relation,

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

Where p is the desired accuracy, e is the outliers ratio and s is the minimum number of points needed to fit the curve. Refer fig 3 for the obtained curves.



(a) Video 1(without noise)



(b) Video 2(with noise)

Fig. 4. RANSAC Curve Fitting.

III. Problem 3

A. Computing SVD of a matrix

Any matrix can be factored into three pieces: $U, \Sigma, andV$.

- U is an orthogonal matrix. Therefore $U^T.U = I$. (Also called as left singular vectors)
- Σ is a diagonal matrix, where the values are called singular values.
- V is also an orthogonal matrix. Therefore, $V^T.V = I$. (Also called as right singular vectors)

This is physically equivalent to rotation, stretching and rotation.[1] As per the SVD,

$$A = U\Sigma V^T$$
$$A^T A = V\Sigma^T U^T U\Sigma V^T$$
$$A^T A = V\Sigma^T \Sigma V^T \tag{2}$$

This equation looks similar to the eigen vector decomposition. Therefore, the columns of V are the orthogonal eigen vectors of $A^T A$ and Σ^2 are the eigen values of $A^T A$.
Lets look at the decomposition again.

$$A = U\Sigma V^T$$
$$AA^T = U\Sigma V^T V\Sigma^T U^T$$

$$AA^T = U\Sigma\Sigma^T U^T \tag{3}$$

By the previous logic, columns of U are the orthogonal eigen vectors of AA^T and Σ^2 are the eigen values of AA^T.
Also, since V is an orthogonal matrix, $V^{-1} = V^T$ Therefore,

$$AV = U\Sigma \tag{4}$$

All equations are referred from [1], [2], [3].

B. *Mathematical computation of SVD for matrix A*

The matrix A is

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

1) Calculate AA^T.

$$\begin{pmatrix} 510051 & 510000 & 15520776 & 6208000 & 33023501 & 12008000 & 7760776 & 15520000 \\ 510000 & 510051 & 15520000 & 6208776 & 33022000 & 12009501 & 7760000 & 15520776 \\ 15520776 & 15520000 & 901062526 & 360416000 & 1023067251 & 372016000 & 30021501 & 60040000 \\ 6208000 & 6208776 & 360416000 & 144188926 & 409217600 & 148829651 & 12008000 & 24017501 \\ 33023501 & 33022000 & 1023067251 & 409217600 & 2178093401 & 792017600 & 511545251 & 1023044000 \\ 12008000 & 12009501 & 372016000 & 148829651 & 792017600 & 288051401 & 186008000 & 372039251 \\ 7760776 & 7760000 & 30021501 & 12008000 & 511545251 & 186008000 & 225282526 & 450520000 \\ 15520000 & 15520776 & 60040000 & 24017501 & 1023044000 & 372039251 & 450520000 & 901062526 \end{pmatrix}$$

2) As per eq. 3, U matrix represent the orthogonal eigen vectors of AA^T. Therefor, find eigen vectors and eigen values of AA^T.
3) Arrange the eigen values in descending order and rearrange the eigen vector matrix as per the corresponding eigen value for each eigen vector. The U matrix is,

$$\begin{pmatrix} 1.17519867e-02 & 3.44207228e-04 & -5.15532162e-02 & -4.66128587e-01 & -2.60345896e-01 & -6.78428560e-02 & 1.08122929e-02 & -8.41087769e-01 \\ 1.17517760e-02 & 3.43641967e-04 & -8.72103737e-02 & -4.59351955e-01 & -2.49098952e-01 & -8.85591890e-02 & 7.65455993e-01 & 3.54169470e-01 \\ 3.58735699e-01 & 6.54942912e-01 & 1.34538659e-02 & -4.65084492e-01 & 1.70101644e-01 & 2.93617516e-01 & -2.78385484e-01 & 1.82289869e-01 \\ 1.43494223e-01 & 2.61976394e-01 & -4.45383120e-01 & 1.36060221e-01 & -5.00795526e-01 & -5.87488150e-01 & -2.73099287e-01 & 1.52897236e-01 \\ 7.74962678e-01 & 2.27117371e-02 & 4.08516159e-01 & 2.84937362e-01 & 3.19642679e-02 & -2.35211438e-01 & 2.62688692e-01 & -1.59658351e-01 \\ 2.81806634e-01 & 8.24745878e-03 & -6.92167142e-01 & 3.15915567e-01 & 1.14149714e-02 & 5.01908806e-01 & 2.46628160e-01 & -1.69560615e-01 \\ 1.84643411e-01 & -3.16806256e-01 & 2.48466337e-01 & -3.46544961e-02 & -6.98268275e-01 & 4.67261587e-01 & -2.52393736e-01 & 1.81630339e-01 \\ 3.69278450e-01 & -6.33614920e-01 & -2.88917222e-01 & -3.93333286e-01 & 3.18917542e-01 & -1.75016528e-01 & -2.61429000e-01 & 1.52633610e-01 \end{pmatrix}$$

4) Since Σ^2 is the eigen value of AA^T, Σ(singular value) shall be the square roots of the eigen values. Find square roots of the eigen values on AA^T and arrange them as a diagonal matrix. The Σ matrix is,

$$\begin{pmatrix} 6.021490e+04 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 \\ 0.000000e+00 & 3.182452e+04 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 \\ 0.000000e+00 & 0.000000e+00 & 2.608900e+02 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 \\ 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 1.862200e+02 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 \\ 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 1.456100e+02 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 \\ 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 6.088000e+01 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 \\ 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 3.900000e+00 & 0.000000e+00 & 0.000000e+00 \\ 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 8.100000e-01 & 0.000000e+00 \end{pmatrix}$$

5) Calculate A^TA.

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

6) As per eq. 2, V matrix represent the orthogonal eigen vectors of A^TA. Therefor, find eigen vectors and eigen values of A^TA.
7) Again. arrange the eigen values in descending order and rearrange the eigen vector matrix as per the corresponding eigen value for each eigen vector. The V matrix is,

$$\begin{pmatrix} 2.84043894e-03 & 3.14430147e-03 & -2.46384735e-01 & -1.58554932e-01 & -1.75245114e-01 & 1.76705635e-01 & 9.13738625e-01 & -1.20261073e-01 & 5.31056350e-02 \\ 2.42121739e-03 & -1.28321626e-03 & -3.77000733e-01 & 1.76600215e-01 & 6.89508147e-01 & 5.90273326e-01 & -5.29344506e-02 & -2.23230964e-03 & -4.91718844e-03 \\ 2.20891154e-05 & 1.13495064e-05 & -2.37217168e-03 & -3.65660431e-03 & 5.19584361e-03 & 7.52000216e-03 & 6.59901592e-02 & 7.85970680e-01 & 6.14648552e-01 \\ 1.09109680e-03 & 1.17416448e-03 & 6.61240940e-01 & 3.41172744e-01 & 5.01749740e-01 & -2.32499365e-01 & 3.72052359e-01 & -4.25903575e-02 & 1.77018784e-02 \\ 1.63479471e-03 & -2.90636016e-03 & 5.74279813e-01 & -7.10405325e-02 & -3.14549320e-01 & 7.49883366e-01 & -6.19835401e-02 & 4.58785873e-03 & -3.93375075e-03 \\ 1.33908907e-05 & -1.14077892e-05 & 5.80190908e-03 & -2.15181510e-03 & 2.88147957e-03 & -5.73504703e-03 & -1.22489909e-01 & -6.04930528e-01 & 7.86750146e-01 \\ -6.96053715e-01 & -7.17961695e-01 & -7.57487350e-05 & -3.79564118e-03 & 2.50334194e-03 & -1.50223526e-04 & 4.37766998e-03 & -5.55196291e-04 & 2.36025045e-04 \\ -7.17950893e-01 & 6.96067270e-01 & 1.62813209e-03 & -3.77529395e-03 & 2.49754245e-03 & 3.65599795e-03 & -6.00114914e-04 & 3.48250731e-05 & -4.91718843e-05 \\ -6.16016024e-03 & 2.29933343e-05 & -1.73453679e-01 & 9.06741660e-01 & -3.78319687e-01 & 6.21986452e-02 & 2.52338778e-02 & -2.47794676e-03 & 7.62164205e-03 \end{pmatrix}$$

1) Computing Homography Matrix:

The homogeneous equation is given by,

$$Ax = 0$$

Where is x is a flattened homograph matrix. Therefore, computing SVD of matrix A can help us calculate the matrix x. Matrix A can be written as,

$$A = U\Sigma V^T$$

The values for matrix x will be the last value of matrix V. We can also say that the eigen vector of A^TA corresponding to the smallest eigen value is the solution for x. Using this relationship, the values for H matrix was computed.

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

Normalizing the matrix, i.e. making the last element as 1(discusses in sec. IV).

$$\text{H} = \begin{bmatrix} 6.96774195e+00 & -6.45161293e-01 & 8.06451613e+0 \\ 2.32258065e+00 & -5.16129035e-01 & 1.03225806e+02 \\ 3.09677420e-02 & -6.45161292e-03 & 1.00000000e+00 \end{bmatrix}$$

IV. Analysis

A. *Curve Fitting*

1) LS curve fitting method performed well for the noiseless data, the curve fitting output depends on the data distribution.
2) TLS curve fitting method performed well for the noiseless data, but the outliers were penalized more for the noisy data. Hence, the curve for noisy data is biased towards the outliers.

3) RANSAC output was not as good as LS and TLS for noiseless data. However, it performed well for the noisy data by eliminating the outliers during curve fitting. The only problem with RANSAC is that same output is not guaranteed if we limit our iterations to N, where N is calculated using eq. 1. That can be solve by always iterating the loop for a large number of times. But that will make the code inefficient when the number of points are more.

B. SVD

For obtaining SVD, we need to calculate the eigen vectors of $A^T A$ and AA^T. However, when we use the numpy.linalg.eig() function form the NumPy library, we might get the discrepancies in the signs. This is because, an eigen vector multiplied by a constant is still an eigen vector. So the utput from the NumPy function can vary in terns of signs. So, if we try obtain the original matrix by multiplying the U, Σ and V matrices, it may happen that the results will not match. But since we are using the last column of matrix V to get solution for the homogeneous equation, this problem can be ignored.

C. Homography matrix

It is important to note that we make the last element of matrix H as 1. The reason being, since we are using four points in calculating the homography matrix, we can obtain eight equation. But since H is a 3×3 matrix, we have nine unknowns. Hence, we assume that the last element is 1 and compute the remaining eight unknowns using the eight equations.

References

[1] https://youtu.be/mBcLRGuAFUk
[2] https://youtu.be/WmDnaoY2Ivs
[3] http://web.mit.edu/course/other/be.400/OldFiles/www/SVD /Singular_Value_Decomposition.htm
[4] https://docs.opencv.org/master/d9/dab/tutorial_homography.html