RYERSON UNIVERSITY

Faculty of Engineering, Architecture and Science

Department of Electrical and Computer Engineering

Course Number	COE843
Course Title	Introduction to Computer Vision
Semester/Year	F2023

Instructor	Guanghui Richard Wang

ASSIGNMENT No.	3

Assignment Title	Problem Set #3

Submission Date	November 6th, 2023
Due Date	November 6th, 2023

Student Name	Hamza Iqbal
Student ID	500973673
Signature*	H.I

**By signing above you attest that you have contributed to this written lab report and confirm that all work you have swung the lab contributed to this lab report is your own work.*

Part 1
<u>Question 1:</u>

(1)

We must first translate both equations into their homogenous forms to obtain the correct x0,x1,x3 & y0,y1,y2 coordinates before computing the intersection. In particular, using the cross product and its arrangement for i, j, and k columns

Line 1: $y = -0.5x + 2 \rightarrow 0.5x + y - 2 = 0$
Line 2: $3x + 6y = 5 \rightarrow 3x + 6y - 5 = 0$

i	j	k
x1	x2	x3
y1	y2	y3

i	j	k
0.5	1	-2
3	6	5

(2)

```
%% Problem 1.

% Compute Intersection point using cross product

L1 = [0.5 1 -2];
L2 = [3 6 -5];

X = cross (L1, L2);

disp (X);
```

Command Window

New to MATLAB? See resources for Getting Started.

```
    7.0000   -3.5000         0

fx >>
```

Code 1: Code for computing cross-product of two lines

In this case, X = l x l’ is the point where two lines, line 1 and line 2, cross.

```
%%

I = [0 0 1];
verify = dot (X, I);
disp (verify);
```

Command Window

New to MATLAB? See resources for Getting Started.

```
     0

fx >>
```

Code 2: Code for verifying cross-product of two lines

The cross product X yields (7, -3.5, 0) as (x1, x2, x3)^T. At infinity, the line equals = (0,0,1)^T. We can verify that the ideal point is on the line of infinity by producing the cross product of the lines. 0 indicates that this result is on the line.

(3)

The conic's equation in:

<u>Inhomogeneous:</u>
ax^2 + bxy + cy^2 + dx + ey + f = 0
7x^2 +-3.5xy + 0 + 0 + 0 = 0
<u>Homogeneous:</u>
ax + by + c = 0
7x -3.5y = 0

<u>Matrix:</u>

x^TCx = 0.

Where
C =

i	j	k
a	b/2	d/2
b/2	c	e/2
d/2	e/2	f

i	j	k
7	-3/5/2	0
-3.5/2	0	0
0	0	0

Problem 2:

(1)

Ideal point can be mapped to a finite point as in this case, the point is associated with numerous parallel lines, but only one of them crosses the specific finite point.

This can be described using the formula:

$$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathtt{A}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ 0 \end{pmatrix} \qquad \mathbf{l}' = \mathtt{H}^{-\mathsf{T}}\mathbf{l}$$

As an ideal point maps to an ideal point, and the line at infinity stays at infinity, but points move along the line.

This causes a resulting shift in the points, and hence this causes the ideal point to become a finite point, and a line at infinity to become a finite line.

This can also be described as:

$$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^\mathsf{T} & v \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathtt{A}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ v_1 x_1 + v_2 x_2 \end{pmatrix}$$

(2)

Given point transformation:

$$\mathbf{x}'_i = \mathtt{H}\mathbf{x}_i$$

The conic can be transformed to:

$$\mathtt{C}' = \mathtt{H}^{-\mathsf{T}}\mathtt{C}\mathtt{H}^{-1}.$$

The conic can be written as:

$$\begin{aligned} \mathbf{x}^{\mathsf{T}}\mathtt{C}\mathbf{x} &= \mathbf{x}'^{\mathsf{T}}[\mathtt{H}^{-1}]^{\mathsf{T}}\mathtt{C}\mathtt{H}^{-1}\mathbf{x}' \\ &= \mathbf{x}'^{\mathsf{T}}\mathtt{H}^{-\mathsf{T}}\mathtt{C}\mathtt{H}^{-1}\mathbf{x}' \end{aligned}$$

The conic can also be defined as:

$$\mathbf{x}^T C\mathbf{x} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} a & \frac{b}{2} & \frac{d}{2} \\ \frac{b}{2} & c & \frac{e}{2} \\ \frac{d}{2} & \frac{e}{2} & f \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

<u>Problem 3:</u>

(1)

<u>Projective Transformation</u>

<u>General Form:</u>

- It can be represented as a 3x3 matrix as:

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

<u>Degrees of Freedom (d.o.f):</u>

- 8 d.o.f

<u>Invariant Properties:</u>

- Collinearity, concurrency, intersection, tangency, inflections
- Cross-ratio of 4 points (ratio of ratio of lengths).

Affine Transformation

General Form:

- It can be represented as a 3x3 matrix as:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Degrees of Freedom (d.o.f):

- 6 d.o.f (2 scales, 2 rotations, 2 translations), the bottom row has a constraint of [0 0 1]
- Non-isotropic scaling

Invariant Properties:

- parallelism, the ratio of areas, the ratio of lengths on parallel lines
- Cross-ratio of 4 points (ratio of ratio of lengths).

Similarity Transformation

General Form:

- It can be represented as a 3x3 matrix as:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Degrees of Freedom (d.o.f):

- 4 d.o.f (1 scale, 1 rotation, 2 translations)
- Metric structure (structure up to similarity)
- Non-isotropic scaling

Invariant Properties:

- Ratios of lengths, angles, ratios of areas, parallel lines

(2) / (3)

We can verify that an affine transformation maps an ideal point to an ideal point using the following affine transformation formula provided:

$$\mathbf{x}' = \mathrm{H}_{\mathrm{A}}\mathbf{x} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \mathbf{x}$$

This thereby equals:

$$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathtt{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ 0 \end{pmatrix}$$

Since affine transformations do not affect the points within its space if a line contains a point it will still contain that ideal point at infinity after undergoing an affine transformation, which describes an ideal point mapping to another ideal point.

Similarly, we can also verify that an affine transformation maps a line at infinity to a line at infinity using the same affine transformation formula, as an affine transformation will also keep the line at infinity and not cause any changes. A line at infinity is a fixed-line under projective transformation H if H is an affinity.

<u>Problem 4.</u>

(1)

Hamza Iqbal
500973673

Problem 4

A circle in a 2D plane intersects the line at infinity at the two circular points as the circular points observes all circles intersecting the infinite line.

where the infinity line is $x_1^2 + x_2^2 = 0$ & $x_3 = 0$

Assuming the circle is:

$$x_1^2 + x_2^2 + dx_1x_3 + ex_2x_3 + fx_3^2 = 0$$

If $I = (1, i, 0)^T$ and $J = (1, -i, 0)^T$, then when the circular points are identified, I changes.

The orthogonality can then be defined as:

$$I = (1,0,0)^T + i(0,1,0)^T$$

I, J are circular, fixed points if and only if H is a similarity.

Hence, $I' = H_s I$

$$= \begin{bmatrix} s\cos\theta & -\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} 1 \\ i \\ 0 \end{pmatrix}$$

$$= se^{-i\theta} \begin{pmatrix} 1 \\ i \\ 0 \end{pmatrix} = I$$

This can be done the same with J. Hence, the 2D circle intersects the line at infinity at the two circular points.

(2) / (3)

(2) If H is a similarity, then the dual conic under the projective transformation H, will be fixed.

Under point transformation $x' = H_s x$, making the dual conic $C^*_\infty = H_s C^*_\infty H_s^T = C^*_\infty$

This causes the line at infinity to be a null vector of dual conic, defined by:

$$I^T l_\infty = J^T l_\infty = 0 \quad \text{and} \quad C^*_\infty l_\infty = (IJ^T + JI^T) l_\infty$$

$$= I(J^T l_\infty) + J(I^T l_\infty)$$

$$= 0$$

(3) We can verify two orthogonal lines are conjugates by first assuming lines l & m are orthogonal if $l^T C^*_\infty m = 0$, and where:

$$l = (l_1, l_2, l_3)^T \quad \text{and} \quad m = (m_1, m_2, m_3)^T$$

The conic can be identified using: $\cos\theta = \dfrac{l^T C^*_\infty m}{\sqrt{(l^T C^*_\infty l)(m^T C^*_\infty m)}}$

Afterwards, the angles for the Euclidean space are measured:

$$\cos\theta = \frac{l_1 m_1 + l_2 m_2}{\sqrt{(l_1^2 + l_2^2)(m_1^2 + m_2^2)}}$$

Hence the two orthogonal lines are conjugates of the dual conic

Problem 5:

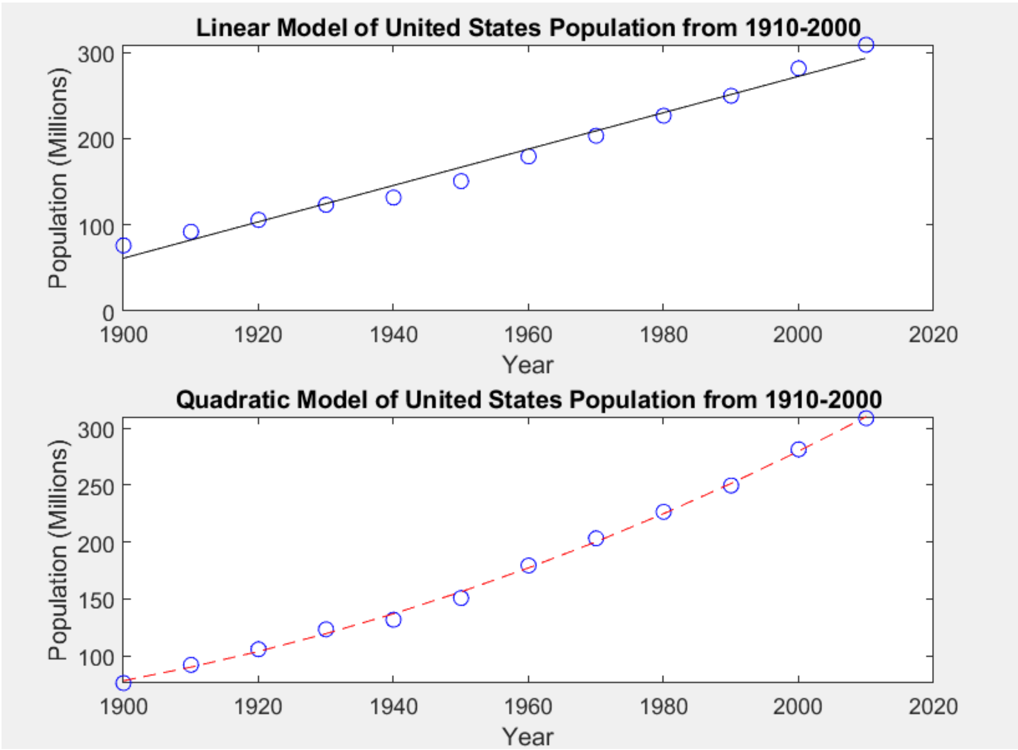

Figure 1: Linear & Quadratic Model of United States Population from 1910-2000

```
%% Problem 5
% United States Population
us_pop = [75.995, 91.972, 105.711, 123.203, 131.669, 150.697, ...
    179.323, 203.212, 226.505, 249.633, 281.422, 308.748]';

% Year (1900-2010)
year = (1900:10:2010)';

% Matrix for linear and quadratic models
arr = [ones(length(year), 1), (year - 1960) / 60];

% Linear & % Quadratic model
linear = arr \ us_pop;
quadratic = [arr, ((year - 1960)/60).^2] \ us_pop;

% Plot of linear Model
subplot(2, 1, 1);
plot(year, us_pop, 'bo');
title('Linear Model of United States Population from 1910-2000');
xlabel('Year');
ylabel('Population (Millions)');
hold on
plot(year,arr*linear,'k-');
hold off

% Plot of quadratic Model
subplot(2, 1, 2);
plot(year, us_pop, 'bo');
title('Quadratic Model of United States Population from 1910-2000');
xlabel('Year');
ylabel('Population (Millions)');
hold on
plot(year, [arr, ((year - 1960)/60).^2]*quadratic, 'r--');
hold off
```

Code 3: Code for Linear & Quadratic Model of United States Population from 1910-2000

Part 2

the MLESAC algorithm is often used for estimating various tasks, such as point data to obtain a solid grasp analysis of, to estimate complex surfaces that are more productive and move efficiently, or various multiple view relations that are present among images related by rigid motions. The view relations have purposes including motion model selection, motion segregation, matching, and structure recovery. Alongside non-adjacent features, point data can be used to represent discrete data points. Neither length nor area can be measured with this dataset as point data has zero dimensions Examples would be schools, points of interest, and bridge and culvert locations.

The MLESAC algorithm needed to be developed as the methodologies that were currently needed to be revised. Due to the set minimal point initially, it was mostly based on point parametrization for estimating multiple views. Using the RANSAC estimator, this is the first part of the MLESAC. Algorithms are preferred because they generate putative solutions that provide maximum accuracy in estimations. In the second part of the method, these relations are automatically parameterized. This method uses a quadratic transformation algorithm for estimating fundamental matrices, using constrained optimization.

<u>Sphere fitting from 3D point cloud:</u>

Because I was unable to locate the source for the object3d.mat file, and no other matlab versions contained it, I was unable to finish the rest of part 2.

Part 3

Our group decided to look into edge detection due to its importance in computer vision, image processing, and machine learning. In certain computer vision processes, edge detection prioritizes recovering lost information from images, geometric and other image features, and methods for extracting compressed image features. Currently, we have begun researching various details along with methods for identifying edges and curves with discontinuities in digital images.

I have enrolled in a group on d2l. I am a part of group 96.

References

Toronto Metropolitan University. 2023. Course Content: Week 4. In Computer Vision, CPS 843. Toronto Metropolitan University's Learning Management System. wk4 - CP8307/CPS843 - Intro to Computer Vision - F2023 (torontomu.ca)

Toronto Metropolitan University. 2023. Course Content: Week 5. In Computer Vision, CPS 843. Toronto Metropolitan University's Learning Management System. wk5 - CP8307/CPS843 - Intro to Computer Vision - F2023 (torontomu.ca)

Toronto Metropolitan University. 2023. Course Content: Week 6. In Computer Vision, CPS 843. Toronto Metropolitan University's Learning Management System. wk6 - CP8307/CPS843 - Intro to Computer Vision - F2023 (torontomu.ca)

Toronto Metropolitan University. 2023. Course Content: Week 7. In Computer Vision, CPS 843. Toronto Metropolitan University's Learning Management System. wk7 - CP8307/CPS843 - Intro to Computer Vision - F2023 (torontomu.ca)

Torr , P., & Zisserman, A. (1996, July 15). MLESAC: A new robust estimator with application ... - university of Oxford. https://www.robots.ox.ac.uk/~vgg/publications/2000/Torr00/torr00.pdf