Open Book Exam for EE4212 (Due: 5th Mar, 2021)

Your name and Matric Number: Chua Pei Yuan A0181003Y
Your pledge: I have not given or received aid from anyone else in solving the problems.
Your Signature:

1. **Projection Model; Homogeneous coordinates**

(a) * Suppose you have two parallel lines in 3-space, one passing through the point (100,100,1000), the other through (200,200,1100). The lines are parallel to the vector (1,2,1). The lines are observed by a unit focal length camera at the origin (i.e. the camera reference frame and the world reference frame are identical). All coordinates are in camera coordinates. What is their point of intersection in the image? (Hint: the point of infinity along $\mathbf{l}_i$ in 3-space is given by $\lim_{\lambda_i \to \infty}$ $\mathbf{l}_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}_C + \lambda_i \begin{bmatrix} a \\ b \\ c \end{bmatrix}$, where $\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}$ and $\begin{bmatrix} a \\ b \\ c \end{bmatrix}$ are a point on the line and its direction respectively)

1a) We denote the two parallel lines using this general eqn

$$l_i = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} + \lambda_i \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

$$l_1 = \begin{pmatrix} 100 \\ 100 \\ 1000 \end{pmatrix} + \lambda_1 \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \quad l_2 = \begin{pmatrix} 200 \\ 200 \\ 1100 \end{pmatrix} + \lambda_2 \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

let $\lambda_1 = 1$ and $\lambda_1 = 2$

$$\therefore P_1 = \begin{pmatrix} 101 \\ 102 \\ 1001 \end{pmatrix} \text{ and } P_{1'} = \begin{pmatrix} 102 \\ 104 \\ 1002 \end{pmatrix}$$

By using linear mapping:

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

For the two points above,

$$P_1' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 101 \\ 102 \\ 1001 \\ 1 \end{pmatrix} = \begin{pmatrix} 101 \\ 102 \\ 1001 \end{pmatrix}$$

$$P_{1'}' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 102 \\ 104 \\ 1002 \\ 1 \end{pmatrix} = \begin{pmatrix} 102 \\ 104 \\ 1002 \end{pmatrix}$$

The line of l_1 on the image formed by the two points are as such

$$l_1' = \begin{pmatrix} 101 \\ 102 \\ 1001 \end{pmatrix} \times \begin{pmatrix} 102 \\ 104 \\ 1002 \end{pmatrix}$$

$$= \begin{pmatrix} -1900 \\ 900 \\ 100 \end{pmatrix}$$

The same can be done for l_2

$$P_2' = \begin{pmatrix} 201 \\ 202 \\ 1101 \end{pmatrix} \quad P_{2'}' = \begin{pmatrix} 202 \\ 204 \\ 1102 \end{pmatrix}$$

$$l_2' = \begin{pmatrix} 201 \\ 202 \\ 1101 \end{pmatrix} \times \begin{pmatrix} 202 \\ 204 \\ 1102 \end{pmatrix}$$

$$= \begin{pmatrix} -2000 \\ 900 \\ 200 \end{pmatrix}$$

(a) Now to find the intersection point of the two parallel lines in 3-space on the image

$$P = l_1' \times l_2'$$

$$= \begin{pmatrix} -1900 \\ 900 \\ 100 \end{pmatrix} \times \begin{pmatrix} -2000 \\ 900 \\ 200 \end{pmatrix}$$

In homogenous $= \begin{pmatrix} 1 \\ 2 \end{pmatrix}$

$$= \begin{pmatrix} 90\,000 \\ 180\,000 \\ 90\,000 \end{pmatrix}$$

$\therefore$ From the above, we can deduce point of intersect by converting back.

$\therefore$ Point of intersection $= (1, 2)$ #

| (b) Given the 3D coordinates of several corresponding points P_i and P_i' in two views, you are required to find the 3D rotation R and translation T that relate the two views ($P_i' = RP_i + T$). Formulate a linear least squares algorithm (of the form Ax=b) that ignores the orthogonality constraint associated with R (that is, it is ok if the solution for R returned by your formulation is not orthogonal). State the entries of the matrix A, and the vectors x and b.

1b) Let R be a 3X3 matrix and T be a 3X1 matrix

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad T = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \quad P_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \quad P_i' = \begin{pmatrix} x_i' \\ y_i' \\ z_i' \end{pmatrix}$$

$P_i' = RP_i + T$ → can be written as a linear mapping in homogenous Coordinate

$$\begin{pmatrix} x'_i \\ y'_i \\ z'_i \\ 1 \end{pmatrix} = \begin{pmatrix} R & T \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x'_i \\ y'_i \\ z'_i \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix}$$

Hence, we can obtain 3 different eqn from above

$x_i' = r_{11} x_i + r_{12} y_i + r_{13} z_i + t_1 + \dots$ —①

$y_i' = r_{21} x_i + r_{22} y_i + r_{23} z_i + t_2 + \dots$ —②

$z_i' = r_{31} x_i + r_{32} y_i + r_{33} z_i + t_3 + \dots$ —③

... → number of 0 to be added

From eqn ①

$$\underset{n\times 4}{\overset{A}{\begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ x_i & y_i & z_i & 1 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}}} \cdot \underset{4\times 1}{\overset{x}{\begin{pmatrix} r_{11} \\ r_{12} \\ r_{13} \\ t_1 \end{pmatrix}}} = \underset{n\times 1}{\overset{b}{\begin{pmatrix} \vdots \\ x_i' \\ \vdots \end{pmatrix}}}$$

From eqn ②

$$\underset{n\times 4}{\begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ x_i & y_i & z_i & 1 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}} \cdot \underset{4\times 1}{\begin{pmatrix} r_{21} \\ r_{22} \\ r_{23} \\ t_2 \end{pmatrix}} = \underset{n\times 1}{\begin{pmatrix} \vdots \\ y_i' \\ \vdots \end{pmatrix}}$$

From eqn ③

$$\underset{n\times 4}{\begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ x_i & y_i & z_i & 1 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}} \cdot \underset{4\times 1}{\begin{pmatrix} r_{31} \\ r_{32} \\ r_{33} \\ t_3 \end{pmatrix}} = \underset{n\times 1}{\begin{pmatrix} \vdots \\ z_i' \\ \vdots \end{pmatrix}}$$

(c) You are now given two sets of corresponding point clouds P_i and P_i', in the files pts.txt and pts_prime.txt respectively. Each row is a 3D coordinate, possibly contaminated with some noise. Write a Matlab routine to estimate the optimal values of R and T in the least squares sense via SVD, using the formulation in (b). Compute the determinant of R using the Matlab function det, and comment on the resulting value.

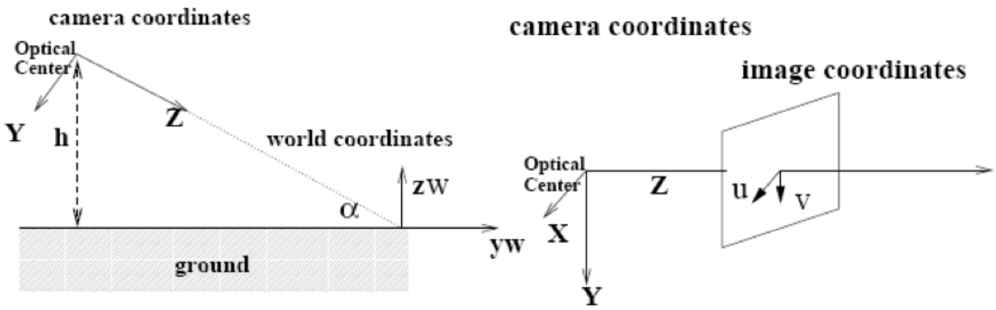
```
Translation T is:
        -0.036545702984502
        -4.00090799340819
         4.00208427886235

Rotation R is:
         0.978227975686295       -0.0084384088198038       0.00939696351887666
        0.0202242852915523       -0.00455704109432295       0.997855473639844
      -0.00158052360791941       -0.994024546260197      -0.000748214891269083

Determinant R is:
         0.970124856796782
```

problems arise as the rotational matrix is not an othorgonal matrix due to the noise.
Hence, the determinant of the matrix will not be 1.

(d) If the world homogeneous coordinates are (X_w, Y_w, Z_w, W_w), the image plane homogeneous coordinates are (u, v, w), and they are related by:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} \sim M \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ W_w \end{pmatrix},$$

i. find the 3x4 matrix M as a function of α, h, according to the diagram below (in the left diagram, the axis X and Xw are pointing directly out of the paper). Assume that the only intrinsic camera parameter is the focal length f (given in pixel unit). Use $\boldsymbol{P_c} = \boldsymbol{RP_w} + \boldsymbol{t}$ to relate the camera coordinates $\boldsymbol{P_c}$ and the world coordinates $\boldsymbol{P_w}$. $\boldsymbol{R}$ is the orientation of the world with respect to the camera; $\boldsymbol{t}$ is the world origin expressed in the camera frame.



ii. * Find the 3x3 matrix for the linear transformation that maps points on the world plane $Z_w = d$ to the image plane (u, v, w).

di) $P_c = RP_w + t$

Firstly, we need to find the rotation matrix for all 3 dimension

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -\sin\alpha & -\cos\alpha & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \end{pmatrix}$$

Next, we will have to find the translational matrix

$$T = \begin{pmatrix} 0 \\ -\frac{h}{\tan\alpha} \\ h \\ 1 \end{pmatrix}$$

$P_c = RP_w + t$

$P_c = R(P_w - T)$

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -\sin\alpha & -\cos\alpha & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -\sin\alpha & -\cos\alpha & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \end{pmatrix} \begin{pmatrix} 0 \\ -\frac{h}{\tan\alpha} \\ h \\ 1 \end{pmatrix}$$

$$\tan\alpha = \frac{\sin\alpha}{\cos\alpha}$$

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} X_w \\ -Y_w \sin\alpha - Z_w\cos\alpha \\ Y_w\cos\alpha - Z_w \sin\alpha \end{pmatrix} - \begin{pmatrix} 0 \\ \frac{h\sin\alpha}{\tan\alpha} - h\cos\alpha \\ -\frac{h\cos\alpha}{\tan\alpha} - h\sin\alpha \end{pmatrix}$$

$$-\frac{h\cos\alpha}{\frac{\sin\alpha}{\cos\alpha}} - h\sin\alpha = -h\left(\frac{\cos^2\alpha}{\sin\alpha} + \sin\alpha\right) = -h\left(\frac{\cos^2\alpha + \sin^2\alpha}{\sin\alpha}\right) = -\frac{h}{\sin\alpha}$$

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} X_w \\ -Y_w \sin\alpha - Z_w\cos\alpha \\ Y_w\cos\alpha - Z_w \sin\alpha \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ -\frac{h}{\sin\alpha} \end{pmatrix}$$

$$\frac{h\sin\alpha}{\frac{\sin\alpha}{\cos\alpha}} - h\cos\alpha = 0$$

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} X_w \\ -Y_w \sin\alpha - Z_w\cos\alpha \\ Y_w\cos\alpha - Z_w \sin\alpha + \frac{h}{\sin\alpha} \end{pmatrix}$$

1di)

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -\sin\alpha & -\cos\alpha & 0 \\ 0 & \cos\alpha & -\sin\alpha & \frac{h}{\sin\alpha} \end{pmatrix} \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix}$$

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X_w \\ -Y_w \sin\alpha - Z_w \cos\alpha \\ Y_w \cos\alpha - Z_w \sin\alpha + \frac{h}{\sin\alpha} \end{pmatrix}$$

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \begin{pmatrix} f X_w \\ -f Y_w \sin\alpha - f Z_w \cos\alpha \\ Y_w \cos\alpha - Z_w \sin\alpha + \frac{h}{\sin\alpha} \end{pmatrix}$$

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \underbrace{\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & -f\sin\alpha & -f\cos\alpha & 0 \\ 0 & \cos\alpha & -\sin\alpha & \frac{h}{\sin\alpha} \end{pmatrix}}_{M} \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix}$$

$$\therefore M = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & -f\sin\alpha & -f\cos\alpha & 0 \\ 0 & \cos\alpha & -\sin\alpha & \frac{h}{\sin\alpha} \end{pmatrix}$$

1dii) Since $\begin{pmatrix} u \\ v \\ w \end{pmatrix} = M \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$

$Z_w = d$

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & -f\sin\alpha & -f\cos\alpha & 0 \\ 0 & \cos\alpha & -\sin\alpha & \frac{h}{\sin\alpha} \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ d \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} f X_w \\ -Y_w f \sin\alpha - d f \cos\alpha \\ Y_w \cos\alpha - d \sin\alpha + \frac{h}{\sin\alpha} \end{pmatrix}$$

$$= \begin{pmatrix} f & 0 & 0 \\ 0 & -f\sin\alpha & -df\cos\alpha \\ 0 & \cos\alpha & -d\sin\alpha + \frac{h}{\sin\alpha} \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ 1 \end{pmatrix}$$

$$M = \begin{pmatrix} f & 0 & 0 \\ 0 & -f\sin\alpha & -df\cos\alpha \\ 0 & \cos\alpha & -d\sin\alpha + \frac{h}{\sin\alpha} \end{pmatrix}$$

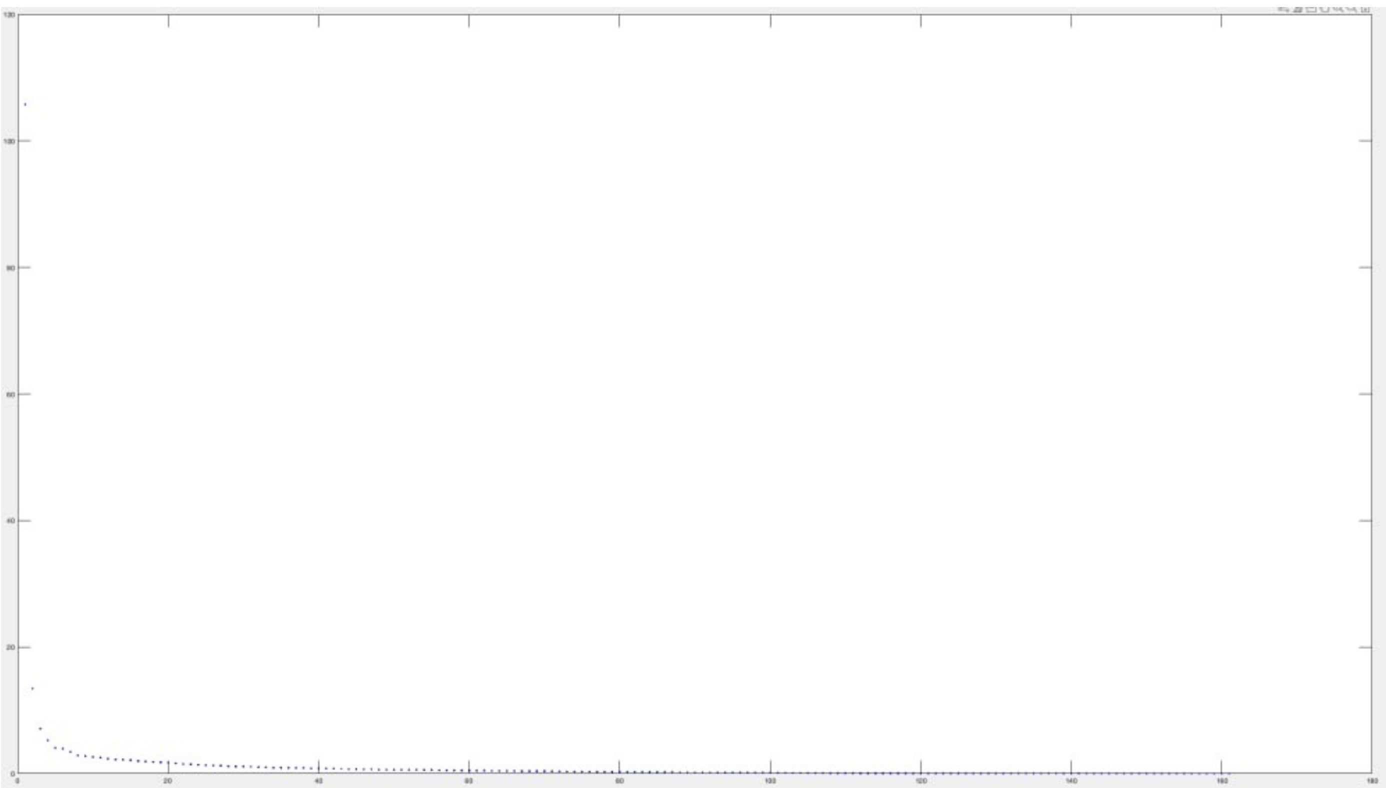
Q2 Using Singular Value Decomposition for Image Compression and PCA.

Many thousands of outdoor cameras are currently connected to the Internet. They can be used to measure plant growth (e.g. in response to recent warming trends), survey animal populations (e.g. in national parks), monitor surf conditions, and security, etc. You can see some examples from the following websites:
Santa Catalina Mountains, Arizona - Dec 5 2006 (http://www.cs.arizona.edu/camera);
Yosemite: Half-dome from Glacier Point - Dec 19 2006 (http://www.halfdome.net/);
Tokyo Riverside Skyline - Jan 08 2007 (http://tokyosky.to/).
Nathan Jacobs AMOS archive: http://cs.uky.edu/~jacobs/datasets/amos/

In this question, you are provided with a 150-frame time-lapse video of a city scene taken between 6.30-7.30pm. Each frame is of the dimension 161x261 pixels. Watch the video, and you can see that clouds are moving, and planes take off and land from a nearby airport.

a. Using Matlab, load the first image in the video sequence provided with this question and convert it to an appropriate data type. Just type the following commands in the command window:
   I= im2double(imread('image001.png'));

b. Do a singular value decomposition using the command 'svd':
   [U S V]=svd(I);
   This will give you the singular values and the singular vectors. The singular values in S have been sorted in descending order. Plot the singular value spectrum using the command: plot(diag(S),'b.').
   Submit this plot. What do you notice in this plot?



The singular value has been plotted as shown above

① The first singular value is a huge number compared to the rest of the values

② As the number of rows increases, the singular value approaches 0

③ The difference between the singular decreases and become very small as the number of rows increases

④ The singular value decreases very quickly

c. Let K=20, Extract the first K singular values and their corresponding vectors in U and V:

```
K=20;
Sk=S(1:K,1:K);
Uk=U(:,1:K);
Vk=V(:,1:K);
```

Uk, Vk, Sk contain the compressed version of the image. To see this, form the compressed image:

Imk=Uk*Sk*Vk'; display it by :imshow(Imk).

Print out a copy of this compressed image and submit it.

d. Repeat question c for K=40,60,80. **Submit the compressed images for the different values of K. Compare the 4 compressed images. Briefly describe what you notice.**

e. Thus, in image transmission, instead of transmitting the original image you can transmit Uk, Vk, Sk, which should be much less data than the original. **Is it worth transmitting when K=100** (i.e. do you save any bits in transmission when K=100)? Explain your answer.

When k=20 The compressed image using the first 20 singular values is shown below

When k= 40

When $k=60$

when $k=80$

The following are observed:

① Even when $k=20$, it is able to capture majority of the features and give a good approximation

② When k increases, the quality of the image becomes better

e. Thus, in image transmission, instead of transmitting the original image you can transmit Uk, Vk, Sk, which should be much less data than the original. **Is it worth transmitting when K=100** (i.e. do you save any bits in transmission when K=100)? Explain your answer.

It is not worth transmitting when $k=100$.

The total storage of the image is

$S = k(m+n+1)$

when $m = 161$ $n=261$ and $K=100$

$S = 100(161+261+1)$

$= 42300$

When $k=100$, the value is greater than the oringinal image which can be calculated as such $161 \times 261 = 42021$. If we transmit $k=100$, we need more data as compared to the original image. Hence, it is not worth.

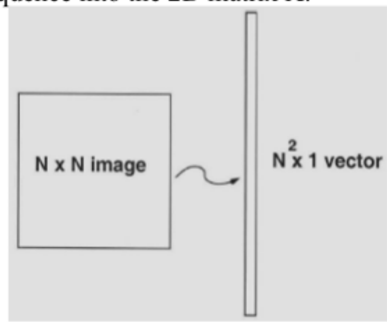
f. ** Find an expression that bounds the per-pixel error (the difference between the original image and the compressed image for a particular pixel (i, j)) in terms of K, the singular values and the elements in U and V with the largest absolute magnitude (u_{max}, v_{max} respectively).

Each individual pixel is as such $a_{ij} = \sum_{k=1}^{n} \sigma_{k} u_{ik} V_{kj}$

per pixel error $= \sqrt{(a_{ij} - a_{ij}')^2}$

$= \sum_{k+1}^{261} u_{ik} \cdot \sigma_{k+1} V^T_{kj} \leq \sum_{k+1}^{261} u_{max} \sigma_{k+1} V_{max}$

g. ** SVD is intimately related to PCA (Principal components analysis). Some of you might have learned about PCA in the EE3731C course, but it is not a pre-requisite for this question. Basically, finding the principal components of a matrix X amounts to finding an orthonormal basis that spans the column space of X (these are the column vectors $\mathbf{u}_i$ in the matrix U). Here we create the data matrix X by first vectorizing each of the 150 images into a column vector (by scanning in either row-major order or column-major order), and then stacking these vectors together into a matrix of size 42021x150. In effect, we have captured the entire video sequence into the 2D matrix X.



Vectorizing an image.

Before we proceed further, mean subtraction (a.k.a. "mean centering") to center the data at the origin is necessary for performing PCA. That is, the images are mean centered by subtracting the mean image vector from each image vector. This is to ensure that the first principal component $\mathbf{u}_1$ really describes the direction of maximum variance. Again, if you do not have background in PCA, it is ok; just take it as a preprocessing step (or you can do some independent learning).

Apply SVD to the resultant mean-centered matrix. Take only the first 10 principal components and reconstruct the image sequence. [NB: you can use the Matlab reshape command to convert a matrix to a vector and vice versa]. Observe <u>the dynamics</u> in the reconstructed video, comment on what you find, and explain why. Remember to add back this mean image vector when you are displaying your reconstructed results.

[Non-mandatory]: You can also explore other possibilities, like investigating what each principal component means, and their implications for processing and editing of outdoor webcam imagery.

We are taking only the first 10 singular values. We can deduce that even though we are just taking the first 10 values, there should be minimal changes to the images. Also, the first 10 singular values are the most prominient othargonal axes that is able to capture the 10 largest variability of the data. It is expected that strong variation can be captured and the less prominiant variability will not be captured.

Question 3: Projection Model; Homogeneous coordinates

a. An affine camera is a simplification of the full perspective camera but is more complicated than the scaled orthographic model. It has a projection relationship given by the following equations:

$$[x,y]^T = A[X,Y,Z]^T + b$$

where A is a 2×3 matrix, and b is a 2×1 vector. If the world point (X,Y,Z) and image point (x,y) are represented by homogeneous vectors, write down the matrix representing the linear mapping between their homogeneous coordinates.

* Show that the point at infinity in space $(X,Y,Z,0)^T$ is mapped to point of infinity in the image plane. What does this result imply about the projection of parallel lines in space onto the image plane?

Points are in homogenous coordinates

3a) Image point $= \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ World point $= \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$

$$\begin{pmatrix} x \\ y \end{pmatrix} = Ax + b$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

$$= \begin{pmatrix} a_{11}x + a_{12}Y + a_{13}Z \\ a_{21}X + a_{22}Y + a_{23}Z \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

$$= \begin{pmatrix} b_1 + a_{11}x + a_{12}Y + a_{13}Z \\ b_2 + a_{21}x + a_{22}Y + a_{33}Z \end{pmatrix}$$

$$= \begin{pmatrix} b_1 + a_{11} + a_{12} + a_{13} \\ b_2 + a_{21} + a_{22} + a_{33} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ X \\ Y \\ Z \end{pmatrix}$$

$$= \begin{pmatrix} a_{11} + a_{12} + a_{13} + b_1 \\ a_{21} + a_{22} + a_{33} + b_2 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

For $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} + a_{12} + a_{13} + b_1 \\ a_{21} + a_{22} + a_{33} + b_2 \\ 0 \quad 0 \quad 0 \quad 1 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

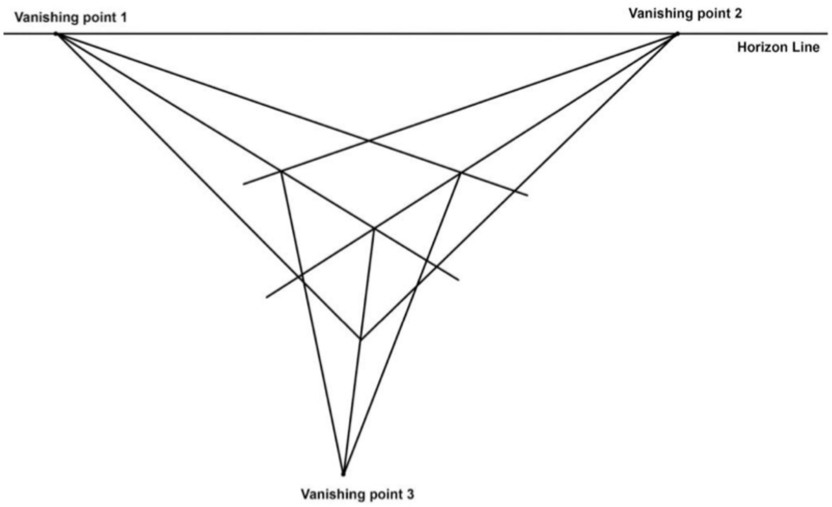
World point $= \begin{pmatrix} X \\ Y \\ Z \\ 0 \end{pmatrix}$

world point projection on image plane will be as such:

$$\begin{pmatrix} a_{11} + a_{12} + a_{13} + b_1 \\ a_{21} + a_{22} + a_{33} + b_2 \\ 0 \quad 0 \quad 0 \quad 1 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} a_{11}X + a_{12}Y + a_{13}Z + b_1 \\ a_{21}X + a_{22}Y + a_{23}Z + b_2 \\ 0 \end{pmatrix}$$

b. ** Given an image of a scene containing a cube as shown in the following.



Each vanishing point is the image of a point at infinity of the form (**d**, 0), where **d** is a Euclidean vector with 3 coordinates expressing the direction of a cube edge. Show that the coordinates of a vanishing point **v** can be expressed as **v** = **K R d**, where **K** is the intrinsic calibration matrix and **R** is the rotation matrix between the camera and world coordinate system. Hence express an edge direction **d** as a function of **K**, **R** and **v**. (Hint: Start from the 3x4 projection matrix equation in Ch 1)

** The 3 directions of the cube edges are mutually perpendicular; therefore the dot product between any two directions is zero. Show that this condition leads to an equation in terms of the vanishing points and the unknown calibration matrix **K**. Note that such an equation can be written for each pair of the 3 vanishing points.

*** (Note that the derived equation above can be used to solve for **K**, but you are not required to explicitly propose a scheme for solving this equation. However, doing so will earn you bonus point.)

3b) $P_c = R(P_w - T)$ → 3X4 matrix $\begin{pmatrix} R' & 0 \\ 0 & 1 \end{pmatrix}$

Assuming there is no translation,

$T = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$

$\Rightarrow P_c = R P_w \quad -①$

$P_{im} = M_{int} \cdot P_c \quad -②$

Sub ① into ②

$P_{im} = M_{int} \cdot R P_w$

when the last row of the world point → 0, the world point reaches a infinite point

Hence we deduce that $P_w = \begin{pmatrix} d \\ 0 \end{pmatrix}$ given by the question

We know that on the image plane, $P_{im} = v$

$P_{im} = K \cdot R P_w$

$V = K \cdot R P_w$

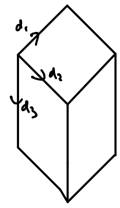
$V = K \cdot \begin{pmatrix} R & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} d \\ 0 \end{pmatrix}$

$V = K \cdot R d$ ✱ (shown)

Since K and R are non singular, $d = R^{-1} K^{-1} v$

Since the dot product any 2 vectors is 0, We can deduce the 3 direction Stated are othorgonal to each other.

Hence, we will have to obtain the othorgonality constraint



Their vanishing point are V_1, V_2 and V_3

$\therefore$ $d_1 \cdot d_2 = 0$

$d_1 \cdot d_3 = 0$

$d_2 \cdot d_3 = 0$

Since d_1, d_2, d_3 are 3×1 matrix

$\Rightarrow (1\times3)\cdot(3\times1)$

$\Rightarrow$ $d_1^T \cdot d_2 = 0$

$d_2^T \cdot d_3 = 0$

$d_1^T \cdot d_3 = 0$

From the previous part,

$$d = R^{-1} K^{-1} v$$

$$Rd = K^{-1} v \quad — (3)$$

The Rotational Matrix is a orthognal matrix.

$$R \cdot R^T = I \quad - (4)$$

$$R^T = R^{-1}$$

$$d_1^T \cdot d_2 = 0$$

From (4) Since $R \cdot R^T = I$

Dot product with Identity matrix is itself.

$$\therefore d_1^T (R^T R) d_2 = 0$$

$$(d_1 R)^T (R d_2) = 0$$

$$(R d_1)^T (R d_2) = 0$$

Sub eqn (3)

$$(K^{-1} V_1)^T (K^{-1} V_2) = 0$$

$$K^{-T} V_1^T K^{-1} V_2 = 0$$

$$V_1^T (K^T K)^{-1} V_2 = 0 \#$$

We rewrite $V_1^T (K^T K)^{-1} V_2 = 0$ into $Ax = 0$ form. From the eqn above, $x = (K^T K)^{-1}$. By performing Singular Value decomposition, the last column of the vector will be the answer.

c. The projection of a scene point in world coordinates to pixel coordinates in an image can be represented using a camera projection matrix P as follows:

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

i. * Given a set of lines in a scene that are all parallel to the world *X*-axis, what is the *vanishing point, (u, v),* of these lines in the image? Can you conclude whether an infinite line in 3D space always yield an infinite line in the 2D image plane?

ii. What is the significance of the image point (represented as a homogeneous 3-vector) given by the last column (p_{14}, p_{24}, p_{34}) of P? That is, which world point gives rise to (p_{14}, p_{24}, p_{34})?

iii. ** Consider the 1-dimensional right null-space of P, i.e., the 4-vector ***C*** such that P***C***=0. In this case, the image point of ***C*** is (0, 0, 0) which is not defined. What is the point ***C*** which possesses this property? Explain.

3ci) Since parallel to world X axis,

Direction vector = $\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

The corresponding vanishing point = $\begin{pmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} P_{11} \\ P_{21} \\ P_{31} \end{pmatrix}$

Hence, the vanishing point will be $\left(\frac{P_{11}}{P_{31}}, \frac{P_{21}}{P_{31}} \right)$

We cannot conclude that an infinite line in 3D space will always yield an infinite line in a 2D image plane. There is a chance that the vanishing point above can be finite.

3cii) The world point that will give the last column of the projection matrix is (0,0,0,1)

This point is the origin of the world point. It is the image point where the translation vector intersect with the image plane.

3ciii) Point C should be the camera center.

This can be proven in the following step.

let z be a point in 3D space. A line can be formed by connecting A and C.

$x(\lambda) = \lambda z + (1-\lambda) C \quad , -\infty < \lambda < \infty$

projection $x = Px(\lambda)$

$= \lambda P z + (1-\lambda) PC \qquad PC = 0$

$= \lambda P z \qquad — ①$

Eqn ① shows that all points on this 3D line are mapped to the same point. Hence the line will have to go through the camera center.

d. The equation of a conics in inhomogeneous coordinates is given by $a x^2 + b xy + c y^2 + d x + e y + f = 0$. Homogenize this equation by the replacement $x \to x_1 / x_3$, $y \to x_2 / x_3$, and write down the homogeneous form. Finally, express this homogeneous form in the matrix form $\mathbf{x}^T \mathbf{C} \mathbf{x} = 0$ where $\mathbf{C}$ is symmetric. Write down the elements of the symmetric matrix $\mathbf{C}$.

** If $\mathbf{C}$ has the special form $\mathbf{C} = \mathbf{l}\,\mathbf{m}^T + \mathbf{m}\,\mathbf{l}^T$, clearly $\mathbf{C}$ is symmetric and therefore represents a conic. Show that the vector $\mathbf{x} = \mathbf{l} \times \mathbf{m}$ is the null vector of $\mathbf{C}$. Since a null vector exists, $\mathbf{C}$ is a degenerate conic. Show in this degenerate case, $\mathbf{C}$ contains two lines $\mathbf{l}$ and $\mathbf{m}$, that is, show that the points on the lines $\mathbf{l}$ and $\mathbf{m}$ satisfy $\mathbf{x}^T \mathbf{C} \mathbf{x} = 0$.

3d) Given

$$ax^2 + bxy + cy^2 + dx + ey + f = 0$$

$$x = \frac{x_1}{x_3} \qquad y = \frac{x_2}{x_3}$$

$$a\left(\frac{x_1^2}{x_3^2}\right) + b\left(\frac{x_1 x_2}{x_3^2}\right) + c\left(\frac{x_2^2}{x_3^2}\right) + d\left(\frac{x_1}{x_3}\right) + e\left(\frac{x_2}{x_3}\right) + f = 0$$

$$ax_1^2 + bx_1x_2 + cx_2^2 + dx_1x_3 + ex_2x_3 + fx_3^2 = 0$$

$$ax_1^2 + cx_2^2 + fx_3^2 + bx_1x_2 + dx_1x_3 + ex_2x_3 = 0$$

$$ax_1^2 + cx_2^2 + fx_3^2 + \tfrac{b}{2}x_1x_2 + \tfrac{b}{2}x_1x_2 + \tfrac{d}{2}x_1x_3 + \tfrac{d}{2}x_1x_3 + \tfrac{e}{2}x_2x_3 + \tfrac{e}{2}x_2x_3 = 0$$

$$\left(ax_1 + \tfrac{b}{2}x_2 + \tfrac{d}{2}x_3 + \tfrac{b}{2}x_1 + cx_2 + \tfrac{e}{2}x_3 + \tfrac{d}{2}x_1 + \tfrac{e}{2}x_2 + \tfrac{f}{2}x_3\right)\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 0$$

$$(x_1 \; x_2 \; x_3)\begin{pmatrix} a & \frac{b}{2} & \frac{d}{2} \\ \frac{b}{2} & c & \frac{e}{2} \\ \frac{d}{2} & \frac{e}{2} & f \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 0$$

Based on the above, we can deduce the Symmetry matrix C.

$$C = \begin{pmatrix} a & \frac{b}{2} & \frac{d}{2} \\ \frac{b}{2} & c & \frac{e}{2} \\ \frac{d}{2} & \frac{e}{2} & f \end{pmatrix} \#$$

$$C = lm^T + ml^T$$

Need to show $X = l \times m$

$$l = \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix} \qquad m = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix}$$

$$C = lm^T + ml^T$$

$$= \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix}(m_1 \; m_2 \; m_3) + \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix}(l_1 \; l_2 \; l_3)$$

$$= \begin{pmatrix} l_1m_1 & l_1m_2 & l_1m_3 \\ l_2m_1 & l_2m_2 & l_2m_3 \\ l_3m_1 & l_3m_2 & l_3m_3 \end{pmatrix} + \begin{pmatrix} m_1l_1 & m_1l_2 & m_1l_3 \\ m_2l_1 & m_2l_2 & m_2l_3 \\ m_3l_1 & m_3l_2 & m_3l_3 \end{pmatrix}$$

3d)

$$Cx = lm^T x + ml^T x$$

$$\begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix} \times \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = \begin{pmatrix} l_2 m_3 - l_3 m_2 \\ l_3 m_1 - l_1 m_3 \\ l_1 m_2 - l_2 m_1 \end{pmatrix} = x$$

For $ml^T x$:

$$ml^T x$$

$$= \begin{pmatrix} m_1 l_1 & m_1 l_2 & m_1 l_3 \\ m_2 l_1 & m_2 l_2 & m_2 l_3 \\ m_3 l_1 & m_3 l_2 & m_3 l_3 \end{pmatrix} \cdot \begin{pmatrix} l_2 m_3 - l_3 m_2 \\ l_3 m_1 - l_1 m_3 \\ l_1 m_2 - l_2 m_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

The first row : $\cancel{m_1 l_1 l_2 m_3} - \cancel{m_1 l_1 l_3 m_2} + \cancel{m_1^2 l_2 l_3} - \cancel{m_1 l_2 l_1 m_3} + \cancel{m_1 l_3 l_1 m_2} - \cancel{m_1^2 l_2 l_3} = 0$

The Second row : $\cancel{m_2 l_1 l_2 m_3} - \cancel{m_2 l_1 l_3 m_2} + \cancel{m_2 l_2 l_3 m_1} - \cancel{m_2 l_2 l_1 m_3} + \cancel{m_2 l_3 l_1 m_2} - \cancel{m_2 l_3 l_2 m_1} = 0$

The third row : $\cancel{m_3 l_1 l_2 m_3} - \cancel{m_3 l_1 l_3 m_2} + \cancel{m_3 l_2 l_3 m_1} - \cancel{m_3 l_2 l_1 m_3} + \cancel{m_3 l_3 l_1 m_2} - \cancel{m_3 l_3 l_2 m_1} = 0$

For $lm^T x$:

$$lm^T x$$

$$= \begin{pmatrix} l_1 m_1 & l_1 m_2 & l_1 m_3 \\ l_2 m_1 & l_2 m_2 & l_2 m_3 \\ l_3 m_1 & l_3 m_2 & l_3 m_3 \end{pmatrix} \cdot \begin{pmatrix} l_2 m_3 - l_3 m_2 \\ l_3 m_1 - l_1 m_3 \\ l_1 m_2 - l_2 m_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

First row : $\cancel{l_1 m_1 l_2 m_3} - \cancel{l_1 m_1 l_3 m_2} + \cancel{l_1 m_2 l_3 m_1} - \cancel{l_1^2 m_2 m_3} + \cancel{l_1^2 m_3 m_2} - \cancel{l_1 m_3 l_2 m_1} = 0$

Second row : $\cancel{l_2 m_1 l_2 m_3} - \cancel{l_2 m_1 l_3 m_2} + \cancel{l_2 m_2 l_3 m_1} - \cancel{l_2 m_2 l_1 m_3} + \cancel{l_2 m_3 l_1 m_2} - \cancel{l_2 m_3 l_2 m_1} = 0$

third row : $\cancel{l_3 m_1 l_2 m_3} - \cancel{l_3 m_1 l_3 m_2} + \cancel{l_3 m_2 l_3 m_1} - \cancel{l_3 m_2 l_1 m_3} + \cancel{l_3 m_3 l_1 m_2} - \cancel{l_3 m_3 l_2 m_1} = 0$

$$Cx = lm^T x + ml^T x$$

$$= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Hence x is the null vector of C

$$x^T C x = x^T lm^T x + x^T ml^T x$$
$$= (ml^T x)^T x + (lm^T x)^T x$$
$$= 0(x) + 0(x)$$
$$= 0 \quad (\text{shown})$$

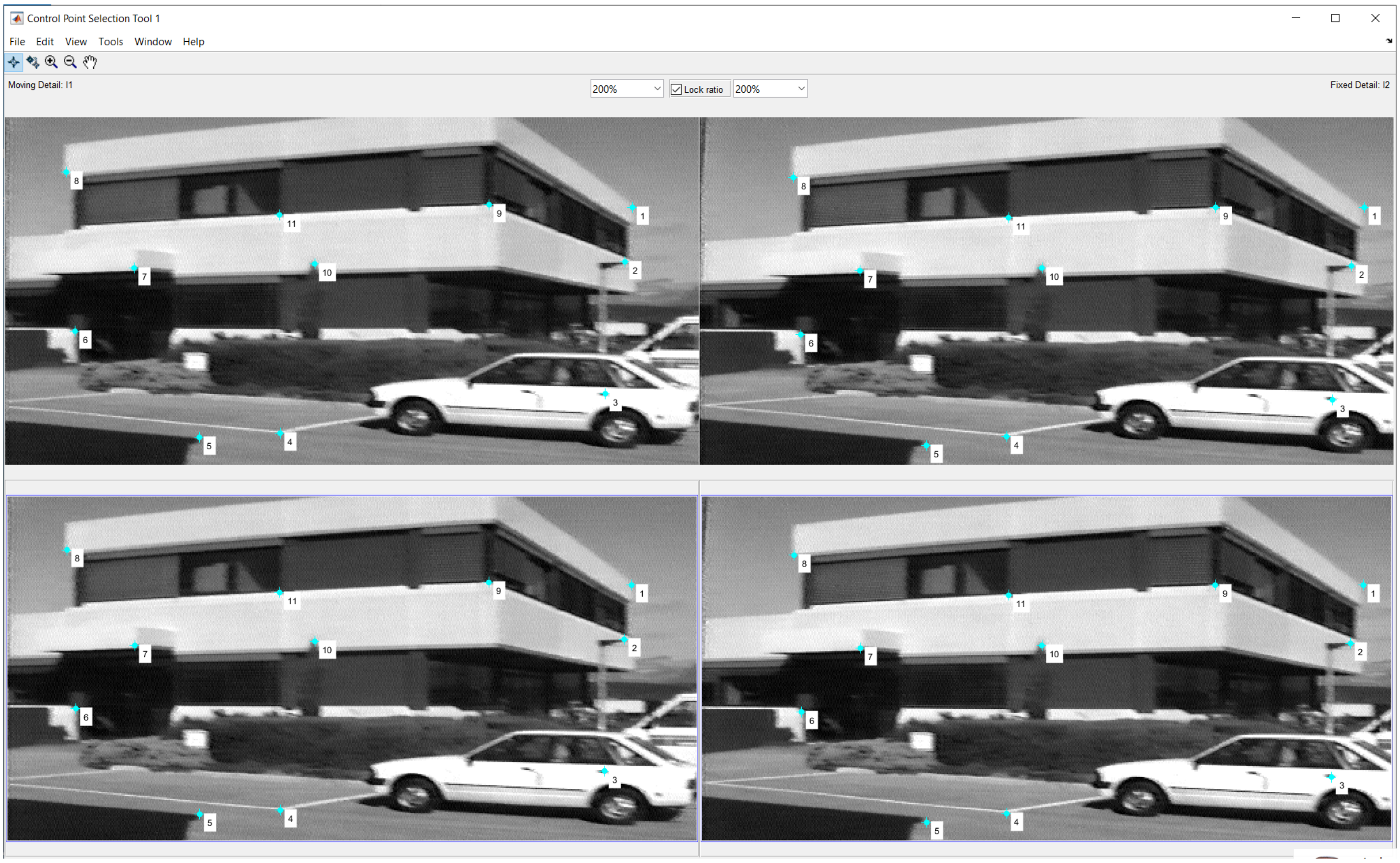
e. Given a set of two-dimensional points (x,y) as follows, write a Matlab routine to find the conics best (in the least squares sense) represented by these points, in terms of the parameters *a, b, c, d, e*, and *f*, as formulated in Question 3d.

3,-8	4,-9	5,-9	6,-9	7,-9	8,-9	9,-9	10,-9	11,-8	12,-8
13,-8	14,-8	15,-7	16,-6	16,-5	17,-4	17,-3	17,-2	18,-1	18,0
18,1	18,2	18,3	18,4	18,5	18,6	18,7	18,8	18,9	17,10
17,11	16,12	16,13	15,14	15,15	14,16	13,17	12,18	11,18	10,19
9,20	8,20	7,20	6,21	5,21	4,21	3,22	2,22	1,22	0,22
-1,21	-2,21	-3,20	-4,19	-5,18	-6,17	-7,16	-7,15	-8,14	-8,13
-8,12	-8,11	-8,10	-8,9	-8,8	-8,7	-8,6	-8,5	-8,4	-7,3
-7,2	-6,1	-5,0	-4,-1	-4,-2	-3,-3	-2,-4	-2,-5	-1,-6	0,-7
1,-7	2,-8								

```
answer =

   0.00954056094243955
   0.00333960309258384
   0.00705787126602787
  -0.118534068061405
  -0.105429838562329
  -0.987259963257305
```

By using Matlab, the last column of V^T is as above.

Q4)



The reason for choosing those points are as such. Firstly, the corners point of the different structure are chosen. This is because the points are easily identifiable or locate. Secondly the points of features chosen must be present in the both images.

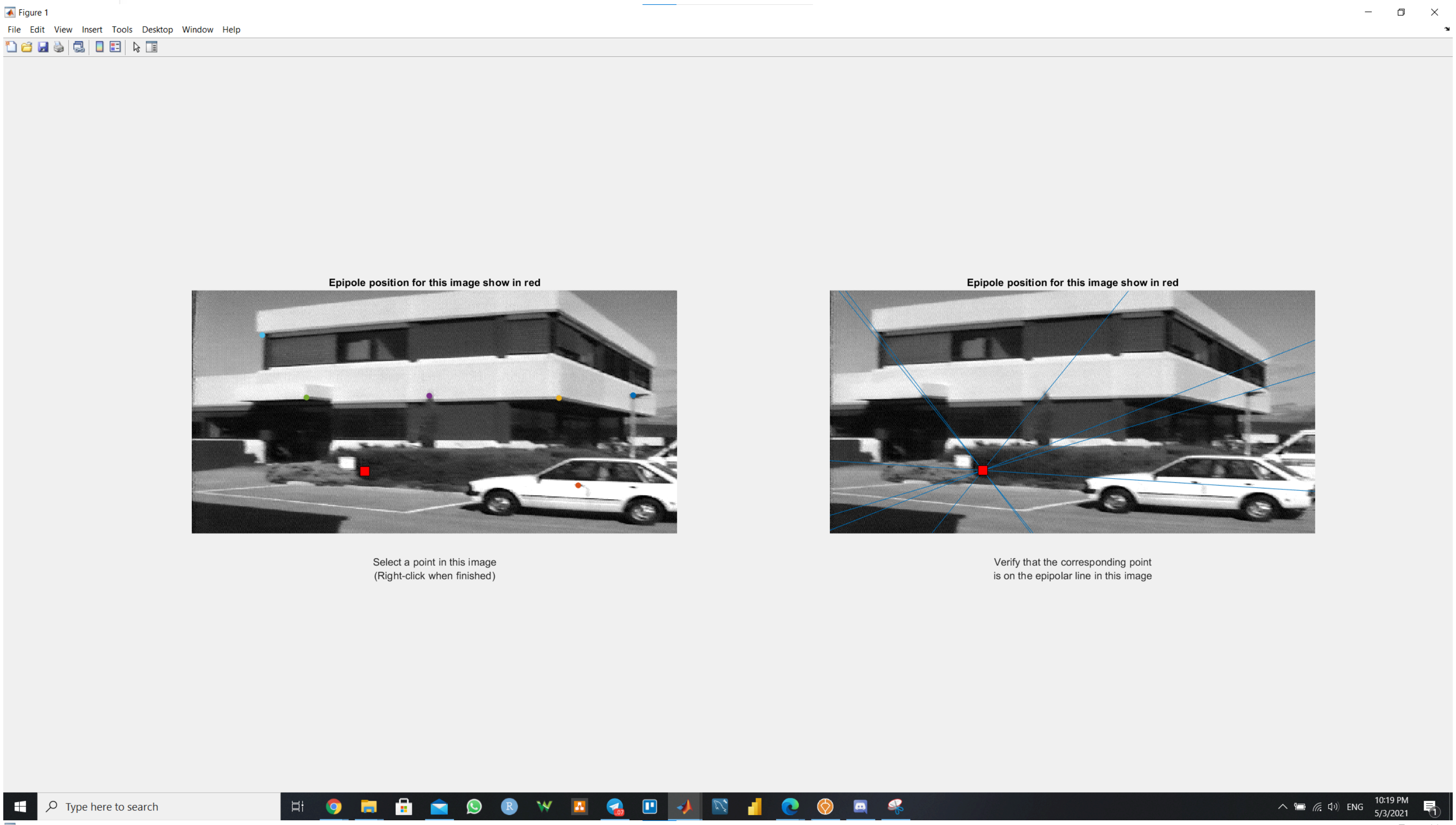
(2)

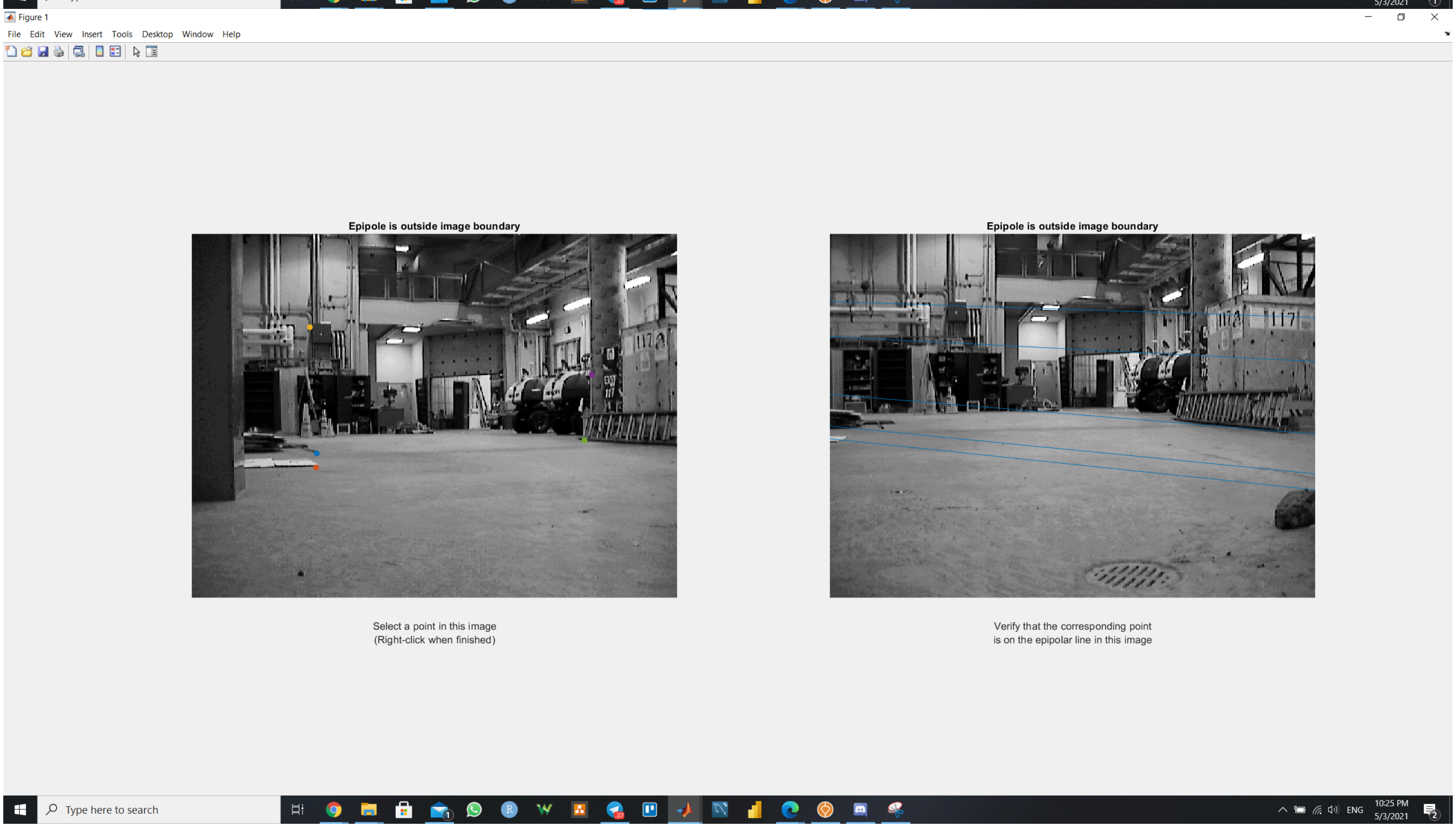
```
function F = estimateF(x1, x2)

matrix_data = size(x1); matrix_height = matrix_data(1);

X1 = [x1 ones(matrix_height, 1)]; X1_mean = mean(x1);
X2 = [x2 ones(matrix_height,1)]; X2_mean = mean(x2);

X1_x = (x1(:,1) - X1_mean(1) * ones(matrix_height,1)).^2;
X1_y = (x1(:,2) - X1_mean(2) * ones(matrix_height,1)).^2;
X1_xy = X1_x + X1_y; X1_total = sum(X1_xy);
S1 = X1_total / (2*matrix_height); S1 = S1^(1/2);

X2_x = (x2(:,1) - X2_mean(1) * ones(matrix_height,1)).^2;
X2_y = (x2(:,2) - X2_mean(2) * ones(matrix_height,1)).^2;
X2_xy = X2_x + X2_y; X2_total = sum(X2_xy);
S2 = X2_total / (2*matrix_height); S2 = S2^(1/2);

Transform1 = [1/S1 0 -X1_mean(1)/S1;0 1/S1 -X1_mean(2)/S1;0 0 1]; X1_new = (Transform1 * X1')';

Transform2 = [1/S2 0 -X2_mean(1)/S2;0 1/S2 -X2_mean(2)/S2;0 0 1]; X2_new = (Transform2 * X2')';

a = X1_new(:,1); b = X1_new(:,2); c = X2_new(:,1); d = X2_new(:,2);
A = [a.*c c.*b c a.*d b.*d d a b ones(matrix_height, 1)];

[U, S, V] = svd(A);
f = V(:,end);

F = [f(1) f(4) f(7); f(2) f(5) f(8);f(3) f(6) f(9)];

[UF, SF, VF] = svd(F);
UFK = UF(:,1:2);SFK = SF(1:2, 1:2);VFK = VF(:, 1:2);
F_singular_prime = UFK*SFK*VFK';

F = Transform2'*F_singular_prime*Transform1;
```

(3)

F =

$$\begin{bmatrix} 1.45563328543133e-07 & -3.81151518352047e-05 & 0.00725394577786924 \\ 3.73879388679822e-05 & 9.20384432309565e-07 & -0.00701935306154317 \\ -0.00712874731901321 & 0.00599497086675212 & 0.159723597272154 \end{bmatrix}$$





(4) The variance of result obtain is due to the fact that the point selected are different. When different points are chosen, it will cause the result of F to be different. Hence the variance is due to the different method of selecting the points.

Additionally, the in accuracy can also be due to the fact that the error calculated was not properly normalized. If it is not properly normalized, the residual have no geometrical meaning. For my case, normalize was done properly. Hence it will not be affected by this problem. Also, the number of points selected will also affect the accuracy. However, if too many points are chosen, it would not make much of a different. This is due to the fact that the additional information might be already captured by the other existing points. Also, if many points are chosen chosen within a small area, this will also result in inaccuracy.

Q5 Motion Perception [you should be able to attempt (a)-(c) by the end of week 5]

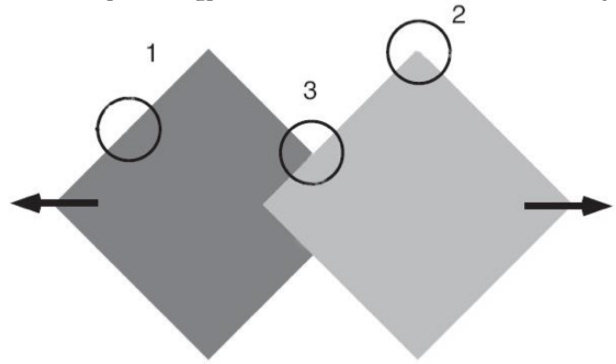


Figure. Two squares translating horizontally in the opposite directions.

(a) In the Figure above, the two squares are translating horizontally in the opposite directions as indicated by the arrows. Indicate the respective perceived motions if you are looking through the apertures 1, 2, and 3. Explain your answers.

According to Brightness Constancy Equation, We can observe the normal flow in an aperture. The aperture problem refer to the fact that the motion of a spatial structure, such as a bar or edge, cannot be determine unambiguosly if it is viewed through a small aperture such that the ends of the stimulus are not visible.

At aperture 1, its edge motion will be ambigious since it is not a unique corner. This happened due to the aperture problem mentioned above that occurs here.

At aperture 2, the corner motion will be unambiguous. Since it is a unique corner.

At aperture 3, the intersection edge motion will be unambigous just like aperture 2. Since at aperture 3, since the intersecting edge is not a actual corner, the motion perceived by the human eye will be false and not be what it purports to be.

5 (b) Barber-pole with Occlusion: The figure in this question illustrates various barber-pole configurations, with occluders placed along either vertical or horizontal sides of the barber-pole. The arrows indicate the perceived direction of barber-pole motion. Explain why the perceived motion tends to be biased in the direction orthogonal to the occlusion boundary.

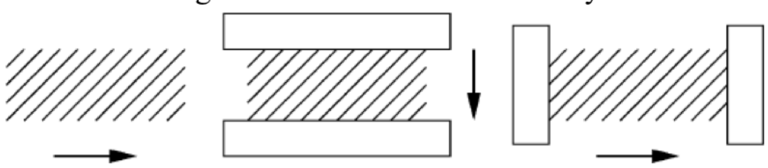
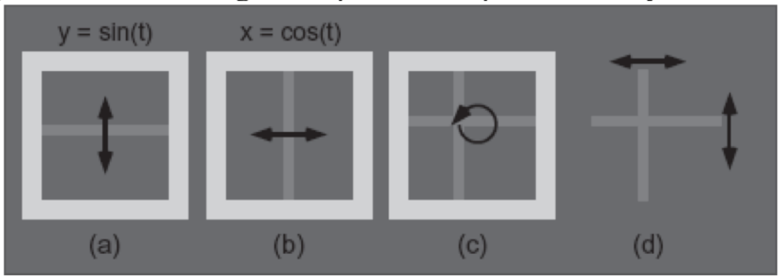


Figure. Barber-pole with occlusion. Arrows indicate the perceived motion. Left: Barber-pole with no occluders. Middle: Occluders placed on the top and bottom cause the perceived motion to be mostly vertical. Right: Occluders placed on the left and right sides of the barber-pole bias the perceived motion toward horizontal.

As mentioned, the arrows indicate the perceived direction of barber pole motion. The motion perceived when occlusion happens is due to the fact that we are following the motion of the occluded end points and treating the the motion direction of those unoccluded end points as the direction of motion. Hence, the perceived motion tends to be biased in the direction othorgonal boundary since we assume the lines are extending behind the occlusion boundary.

(c) *Referring to the diagram below, the stimulus consists of two orthogonal bars that move sinusoidally, 90 degree out of phase (Figure a and b). When presented together within an occluding aperture (Figure c), the bars perceptually cohere and appear to move in a circle as a solid cross. However, when presented alone (Figure d), they appear to move separately (the horizontal bar translates vertically and the vertical bar translates horizontally), even though the image motion is unchanged in Figures c and d. In either stimulus condition, both percepts are legitimate interpretations of the image motion. Yet a single interpretation is predominantly seen in each case. Why?



The human brain recovers rigid transformation from the two different view we perceived. In order to recover the transformation, we need to know the correspondance which is the information of which element this correspond in the next scene. The change that we observed from (c) to d can be explained using junction. The junction are formed where the occlusion overlap. Hence the motion observed will not be what it purport to be. Additionally, unique points at the end of the bar will be ignored by the system when occluder causes the junction at the mentioned locations. When there is no occluder, the endpoint motion cannot be ignored. The motions are then essential and needed. Hence, the main reason for the different interpretation of the motion is due to different image feature used.