EE4212: Computer Vision

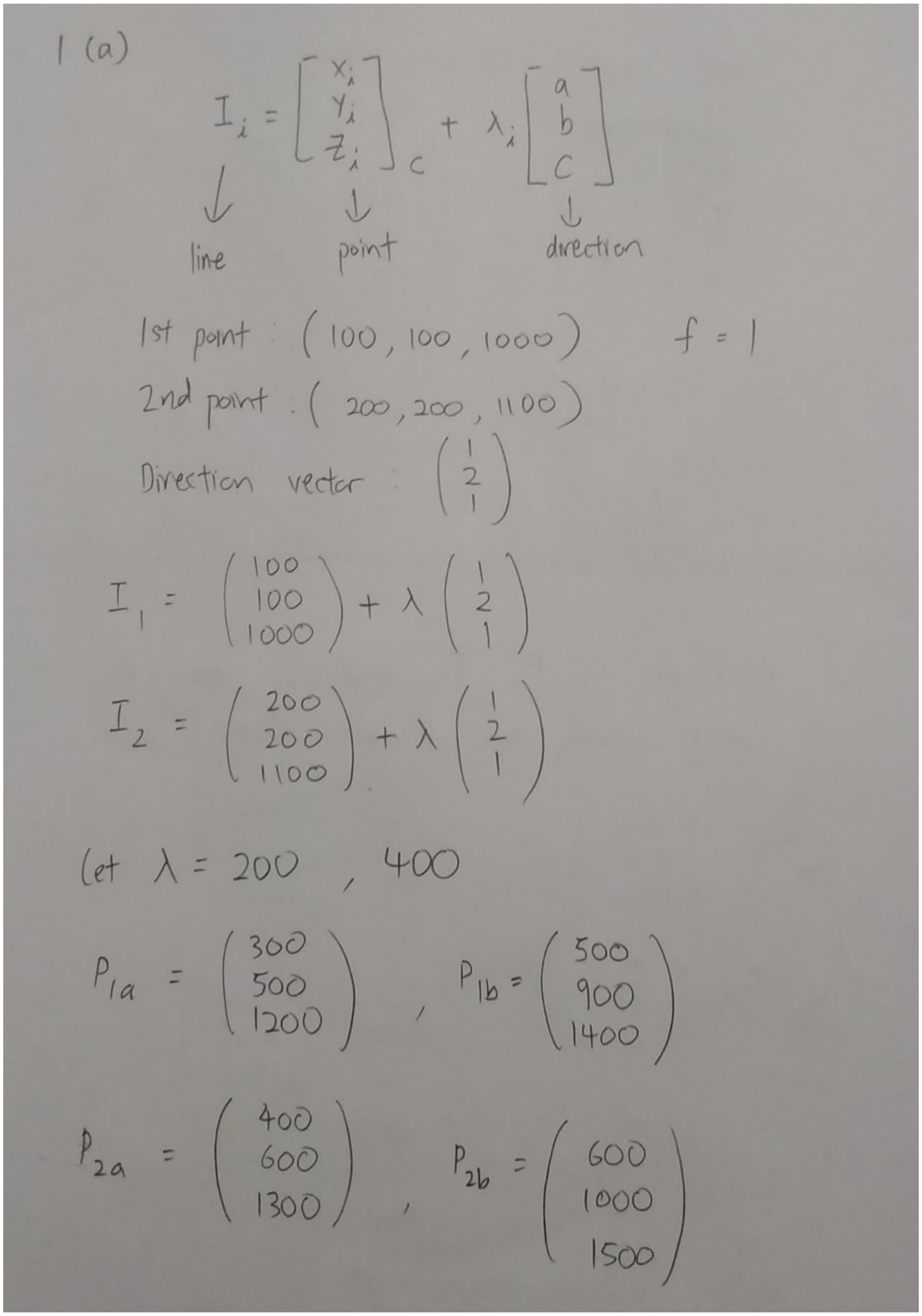
CA1

Lim Yu Guang

A0172618B

1. **Projection Model; Homogeneous coordinates**

(a)



1 (a)

$$\underbrace{I_i}_{\text{line}} = \underbrace{\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}_c}_{\text{point}} + \lambda_i \underbrace{\begin{bmatrix} a \\ b \\ c \end{bmatrix}}_{\text{direction}}$$

1st point : $(100, 100, 1000)$ $\quad f = 1$

2nd point : $(200, 200, 1100)$

Direction vector : $\begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$

$$I_1 = \begin{pmatrix} 100 \\ 100 \\ 1000 \end{pmatrix} + \lambda \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

$$I_2 = \begin{pmatrix} 200 \\ 200 \\ 1100 \end{pmatrix} + \lambda \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

let $\lambda = 200 \; , \; 400$

$$P_{1a} = \begin{pmatrix} 300 \\ 500 \\ 1200 \end{pmatrix}, \quad P_{1b} = \begin{pmatrix} 500 \\ 900 \\ 1400 \end{pmatrix}$$

$$P_{2a} = \begin{pmatrix} 400 \\ 600 \\ 1300 \end{pmatrix}, \quad P_{2b} = \begin{pmatrix} 600 \\ 1000 \\ 1500 \end{pmatrix}$$

Linear mapping between homogeneous coordinates,

$$\overset{3\times 1}{\begin{pmatrix} x \\ y \\ w \end{pmatrix}} = \overset{3\times 4}{\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}} \overset{4\times 1}{\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}}$$

$f = 1$,

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}$$

Points of P_{1a}, P_{2a}, P_{1b} and P_{2b} on the image (using the formula above),

$$P_{1a}' = \begin{pmatrix} 300 \\ 500 \\ 1200 \end{pmatrix}, \quad P_{1b}' = \begin{pmatrix} 500 \\ 900 \\ 1400 \end{pmatrix}$$

$$P_{2a}' = \begin{pmatrix} 400 \\ 600 \\ 1300 \end{pmatrix}, \quad P_{2b}' = \begin{pmatrix} 600 \\ 1000 \\ 1500 \end{pmatrix}$$

$$l_1' = \begin{pmatrix} 300 \\ 500 \\ 1200 \end{pmatrix} \times \begin{pmatrix} 500 \\ 900 \\ 1400 \end{pmatrix}$$

$$= \begin{pmatrix} -380\,000 \\ 180\,000 \\ 20\,000 \end{pmatrix}$$

$$l_2' = \begin{pmatrix} 400 \\ 600 \\ 1300 \end{pmatrix} \times \begin{pmatrix} 600 \\ 1000 \\ 1500 \end{pmatrix}$$

$$= \begin{pmatrix} -400\,000 \\ 180\,000 \\ 40\,000 \end{pmatrix}$$

Point of intersection on the image:

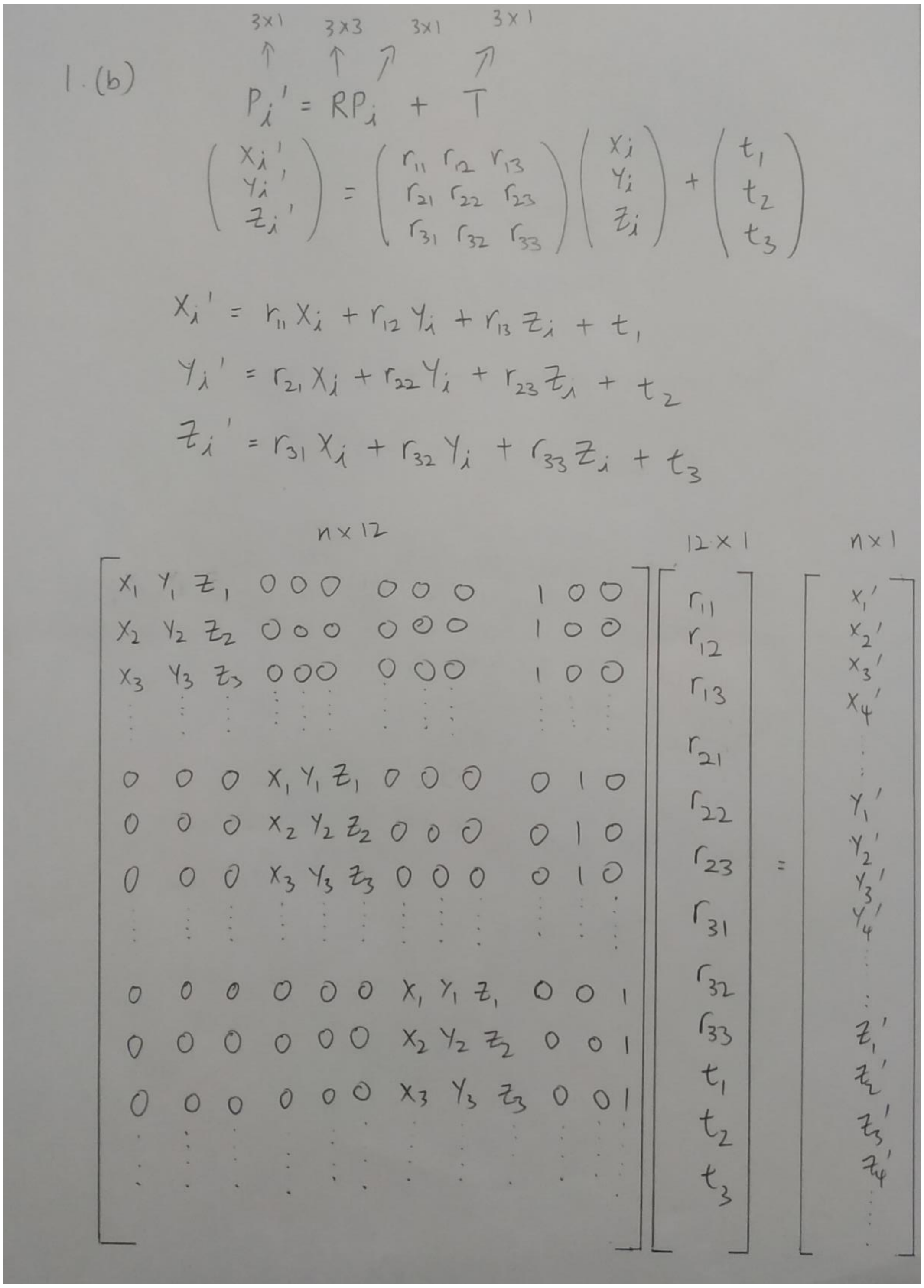
$$p' = l_1' \times l_2'$$

$$= \begin{pmatrix} -380\,000 \\ 180\,000 \\ 20\,000 \end{pmatrix} \times \begin{pmatrix} -400\,000 \\ 180\,000 \\ 40\,000 \end{pmatrix}$$

$$= \begin{pmatrix} 3.6 \times 10^9 \\ 7.2 \times 10^9 \\ 3.6 \times 10^9 \end{pmatrix}$$

$\therefore$ The point of intersection in the image $= \begin{pmatrix} 1 \\ 2 \end{pmatrix}$

(b)



1. (b)

$$\overset{3\times 1}{P_i'} = \overset{3\times 3}{R}\overset{3\times 1}{P_i} + \overset{3\times 1}{T}$$

$$\begin{pmatrix} X_i' \\ Y_i' \\ Z_i' \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}$$

$$X_i' = r_{11}X_i + r_{12}Y_i + r_{13}Z_i + t_1$$

$$Y_i' = r_{21}X_i + r_{22}Y_i + r_{23}Z_i + t_2$$

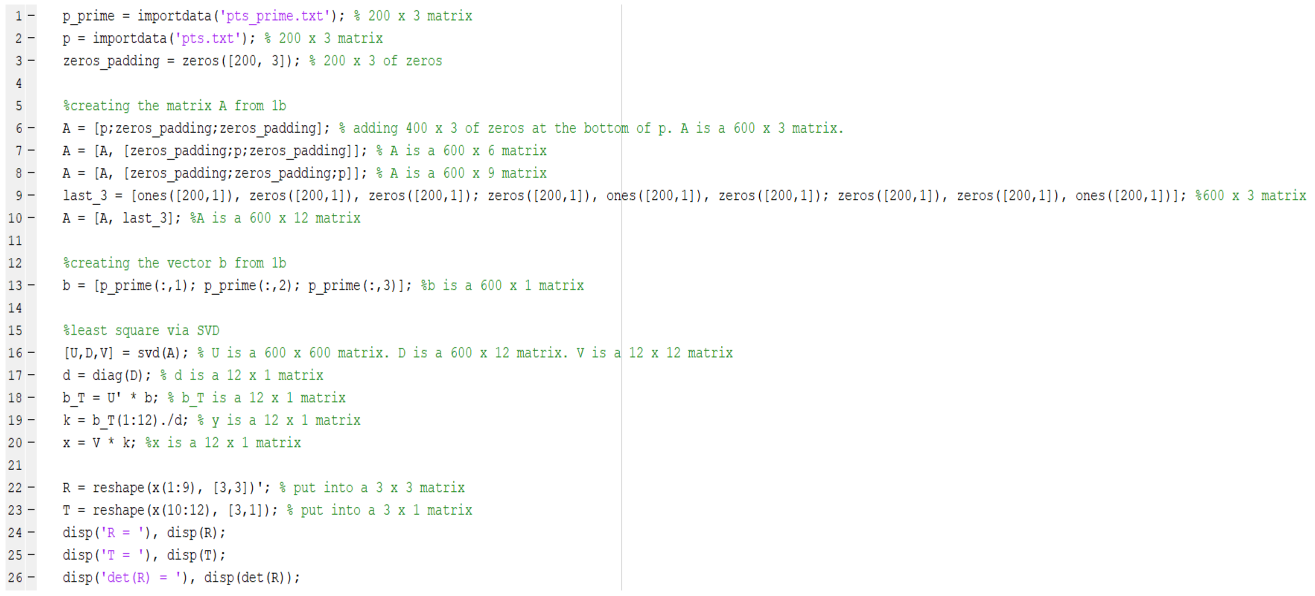
$$Z_i' = r_{31}X_i + r_{32}Y_i + r_{33}Z_i + t_3$$

$$\overset{n\times 12}{\begin{bmatrix} X_1 & Y_1 & Z_1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ X_2 & Y_2 & Z_2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ X_3 & Y_3 & Z_3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & X_3 & Y_3 & Z_3 & 0 & 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & X_3 & Y_3 & Z_3 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}} \overset{12\times 1}{\begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \\ r_{21} \\ r_{22} \\ r_{23} \\ r_{31} \\ r_{32} \\ r_{33} \\ t_1 \\ t_2 \\ t_3 \end{bmatrix}} = \overset{n\times 1}{\begin{bmatrix} X_1' \\ X_2' \\ X_3' \\ X_4' \\ \vdots \\ Y_1' \\ Y_2' \\ Y_3' \\ Y_4' \\ \vdots \\ Z_1' \\ Z_2' \\ Z_3' \\ Z_4' \\ \vdots \end{bmatrix}}$$

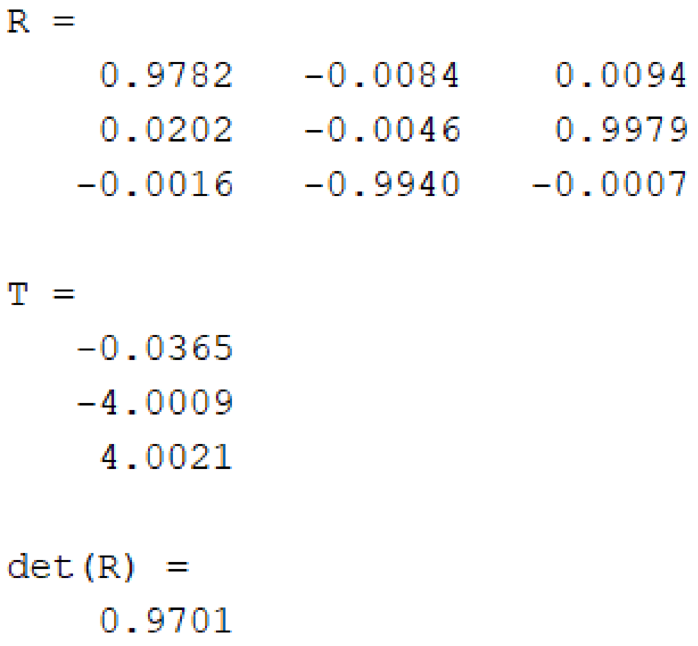
$$A = \underbrace{\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_i & y_i & z_i & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & x_i & y_i & z_i & 0 & 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & x_i & y_i & z_i & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}}_{}^{n \times 12}, \quad i = 1,2,3,\dots,n$$

$$\vec{x} = \underbrace{\begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \\ r_{21} \\ r_{22} \\ r_{23} \\ r_{31} \\ r_{32} \\ r_{33} \\ t_1 \\ t_2 \\ t_3 \end{bmatrix}}_{12 \times 1} \qquad \vec{b} = \underbrace{\begin{bmatrix} \vdots \\ x_i' \\ \vdots \\ y_i' \\ \vdots \\ z_i' \\ \vdots \end{bmatrix}}_{n \times 1}, \quad i = 1,2,3,\dots,n$$

(c)



```
p_prime = importdata('pts_prime.txt'); % 200 x 3 matrix
p = importdata('pts.txt'); % 200 x 3 matrix
zeros_padding = zeros([200, 3]); % 200 x 3 of zeros

%creating the matrix A from 1b
A = [p;zeros_padding;zeros_padding]; % adding 400 x 3 of zeros at the bottom of p. A is a 600 x 3 matrix.
A = [A, [zeros_padding;p;zeros_padding]]; % A is a 600 x 6 matrix
A = [A, [zeros_padding;zeros_padding;p]]; % A is a 600 x 9 matrix
last_3 = [ones([200,1]), zeros([200,1]), zeros([200,1]); zeros([200,1]), ones([200,1]), zeros([200,1]); zeros([200,1]), zeros([200,1]), ones([200,1])]; %600 x 3 matrix
A = [A, last_3]; %A is a 600 x 12 matrix

%creating the vector b from 1b
b = [p_prime(:,1); p_prime(:,2); p_prime(:,3)]; %b is a 600 x 1 matrix

%least square via SVD
[U,D,V] = svd(A); % U is a 600 x 600 matrix. D is a 600 x 12 matrix. V is a 12 x 12 matrix
d = diag(D); % d is a 12 x 1 matrix
b_T = U' * b; % b_T is a 12 x 1 matrix
k = b_T(1:12)./d; % y is a 12 x 1 matrix
x = V * k; %x is a 12 x 1 matrix

R = reshape(x(1:9), [3,3])'; % put into a 3 x 3 matrix
T = reshape(x(10:12), [3,1]); % put into a 3 x 1 matrix
disp('R = '), disp(R);
disp('T = '), disp(T);
disp('det(R) = '), disp(det(R));
```



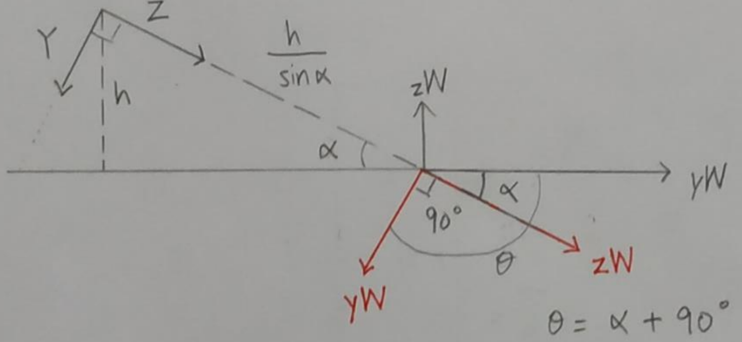
```
R = 
    0.9782   -0.0084    0.0094
    0.0202   -0.0046    0.9979
   -0.0016   -0.9940   -0.0007

T = 
   -0.0365
   -4.0009
    4.0021

det(R) = 
    0.9701
```

The estimated optimal values of R and T from MatLAB are shown in the above image. The determinant of R is 0.9701. The determinant of R is close to 1. The determinant of the rotational matrix, R should deviate from 1. This is because the rotational matrix, R is not an orthogonal matrix and there is error due to Gaussian noise. We ignored the orthogonality constraint associated with R when formulating the linear least squares algorithm.

(d)

1. (d)(i) $P_c = RP_w + t$

$$\underset{3\times 1}{P_c} = \underset{3\times 4}{M}\underset{4\times 1}{P_w}$$



The red axis represents the world axis after rotation

$\theta = \alpha + 90^\circ$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

Consider this matrix because rotation is about the x-axis

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

$$t = \begin{bmatrix} 0 \\ 0 \\ \frac{h}{\sin\alpha} \end{bmatrix}$$

$\sin x = \frac{O}{H}$ → opposite, → hypotenuse

$\sin\alpha = \frac{h}{H}$

$H = \frac{h}{\sin\alpha}$

$$M = \overset{3\times4}{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}} \overset{4\times4}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\sin\alpha & -\cos\alpha & 0 \\ 0 & \cos\alpha & -\sin\alpha & \frac{h}{\sin\alpha} \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

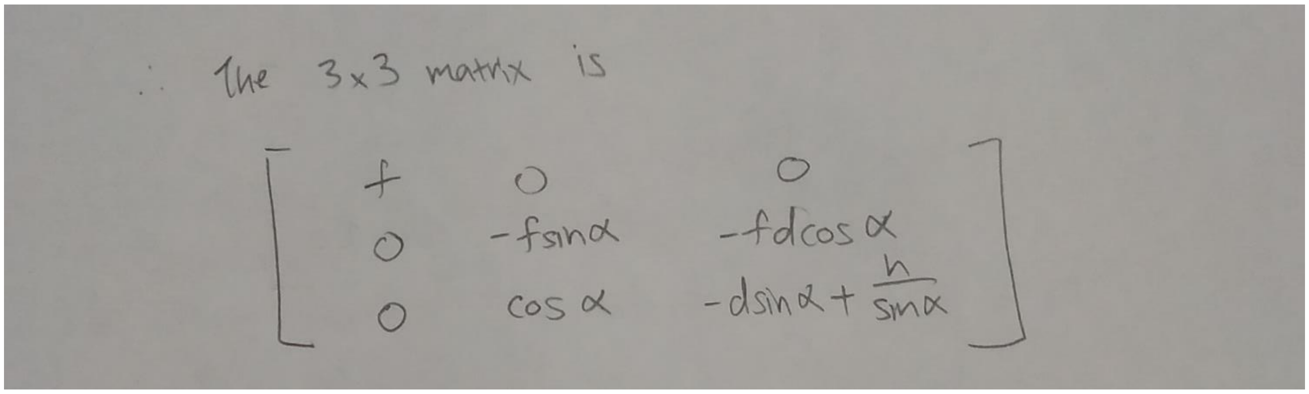
$$= \overset{3\times4}{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & -f\sin\alpha & -f\cos\alpha & 0 \\ 0 & \cos\alpha & -\sin\alpha & \frac{h}{\sin\alpha} \end{bmatrix}}$$

(ii) $$\overset{3\times1}{\begin{bmatrix} u \\ v \\ w \end{bmatrix}} = \overset{3\times4}{M} \overset{4\times1}{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}$$

$$= \overset{3\times4}{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & -f\sin\alpha & -f\cos\alpha & 0 \\ 0 & \cos\alpha & -\sin\alpha & \frac{h}{\sin\alpha} \end{bmatrix}} \overset{4\times1}{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}$$

$$= \overset{3\times1}{\begin{bmatrix} fX_w \\ -fY_w\sin\alpha - fd\cos\alpha \\ Y_w\cos\alpha - d\sin\alpha + \frac{h}{\sin\alpha} \end{bmatrix}} , \; Z_w = d$$

$$= \overset{3\times3}{\begin{bmatrix} f & 0 & 0 \\ 0 & -f\sin\alpha & -fd\cos\alpha \\ 0 & \cos\alpha & -d\sin\alpha + \frac{h}{\sin\alpha} \end{bmatrix}} \overset{3\times1}{\begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}}$$



∴ The 3x3 matrix is

$$\begin{bmatrix} f & 0 & 0 \\ 0 & -f\sin\alpha & -fd\cos\alpha \\ 0 & \cos\alpha & -d\sin\alpha + \frac{h}{\sin\alpha} \end{bmatrix}$$

2. Using Singular Value Decomposition for Image Compression and PCA

(b)

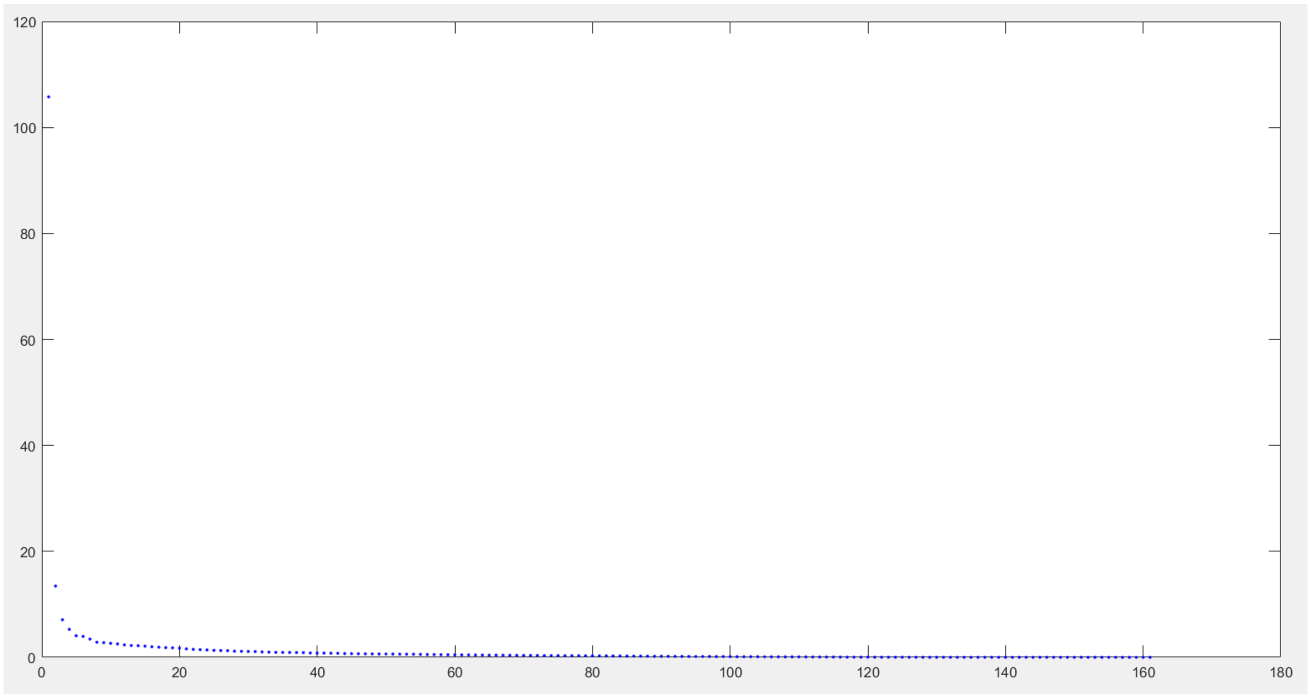


Figure 1: Singular Value Spectrum

From the above plot, the singular values decrease drastically. The first singular value is approximately 105 while the remaining singular values are less than 20, and eventually the singular values tend towards 0.

(c)

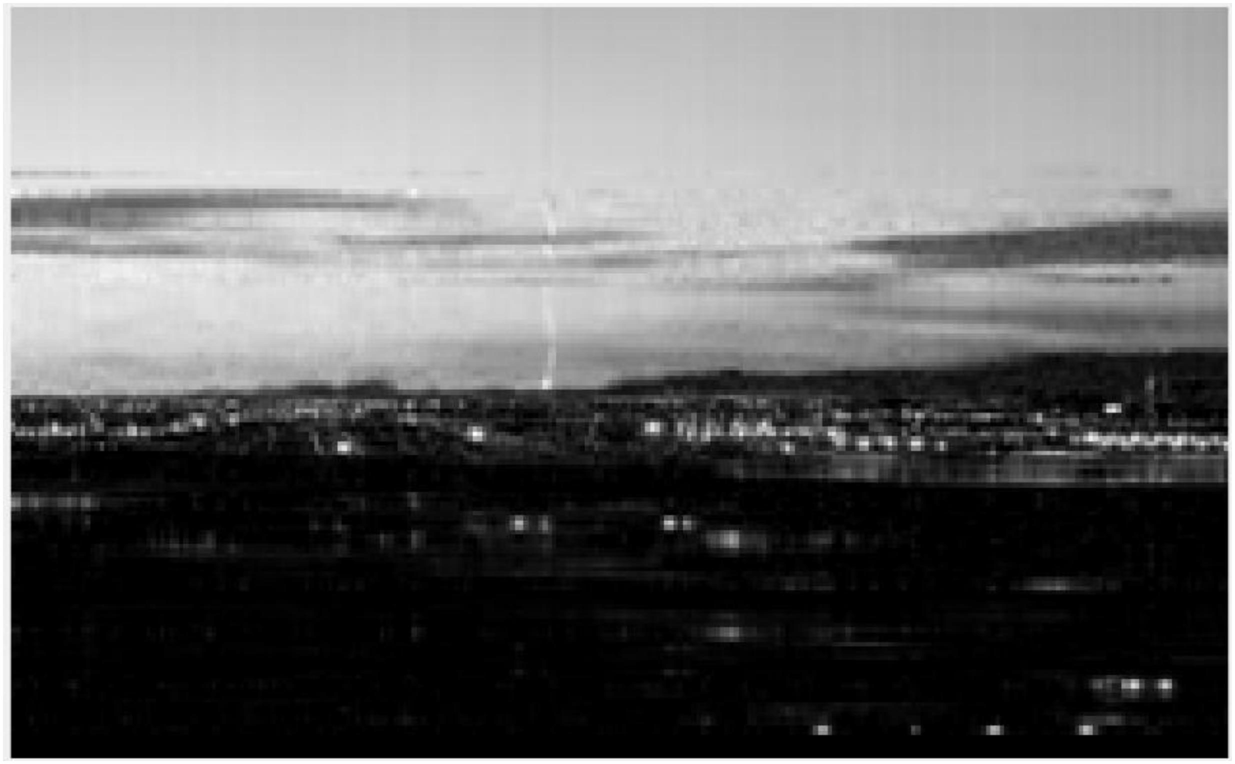


Figure 2: Compressed Image (K = 20)

(d)

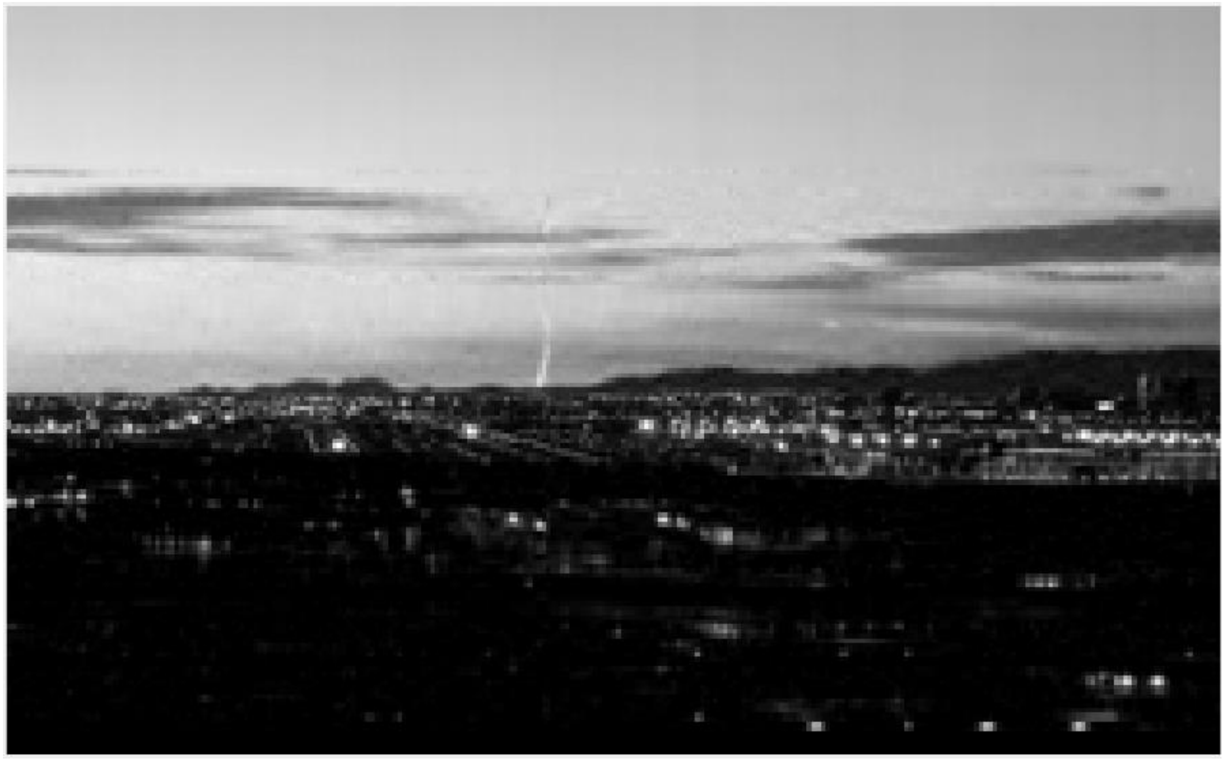


Figure 3: Compressed Image (K = 40)

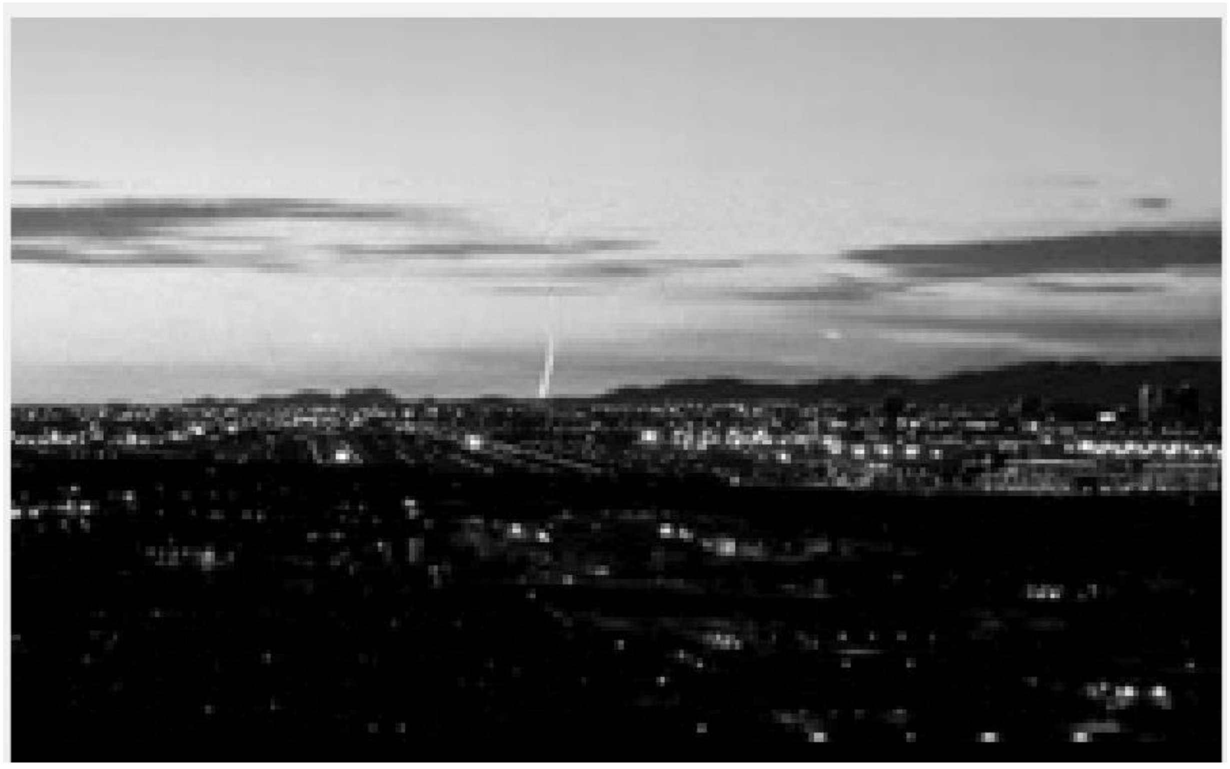


Figure 4: Compressed Image (K = 60)

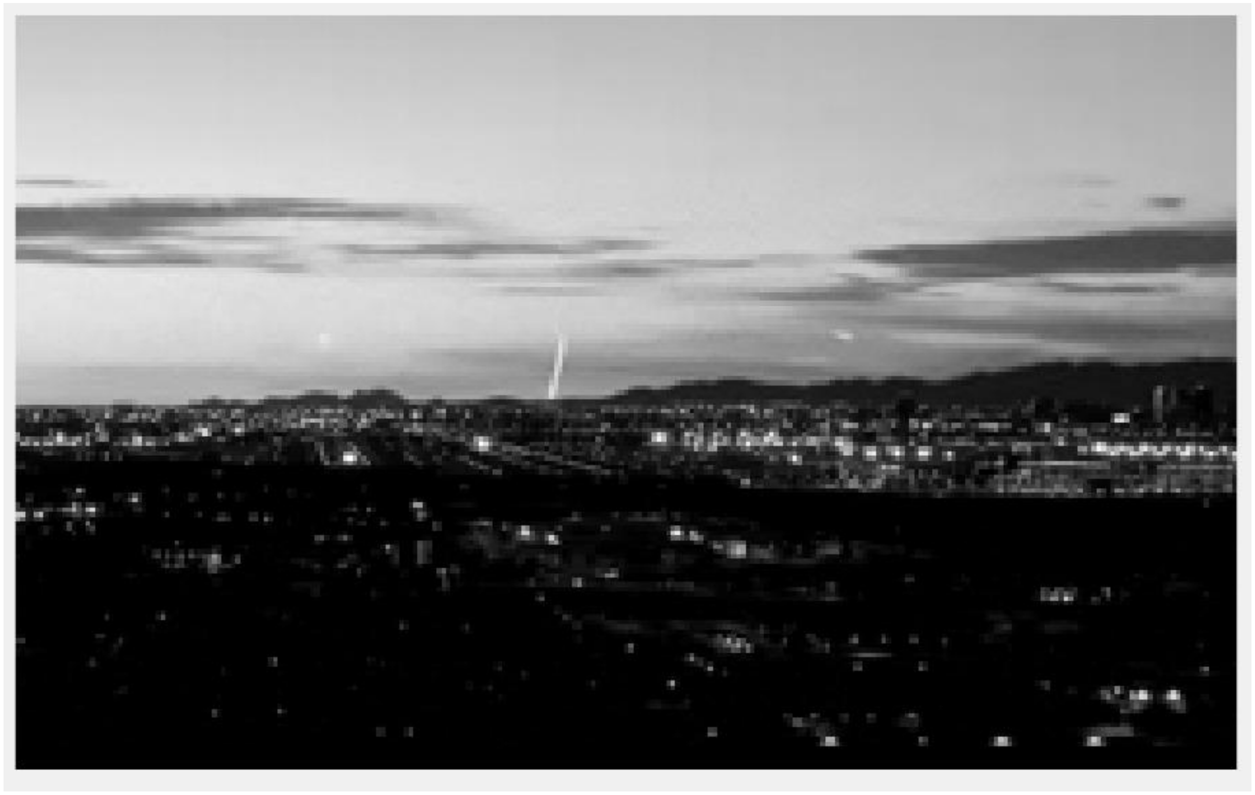


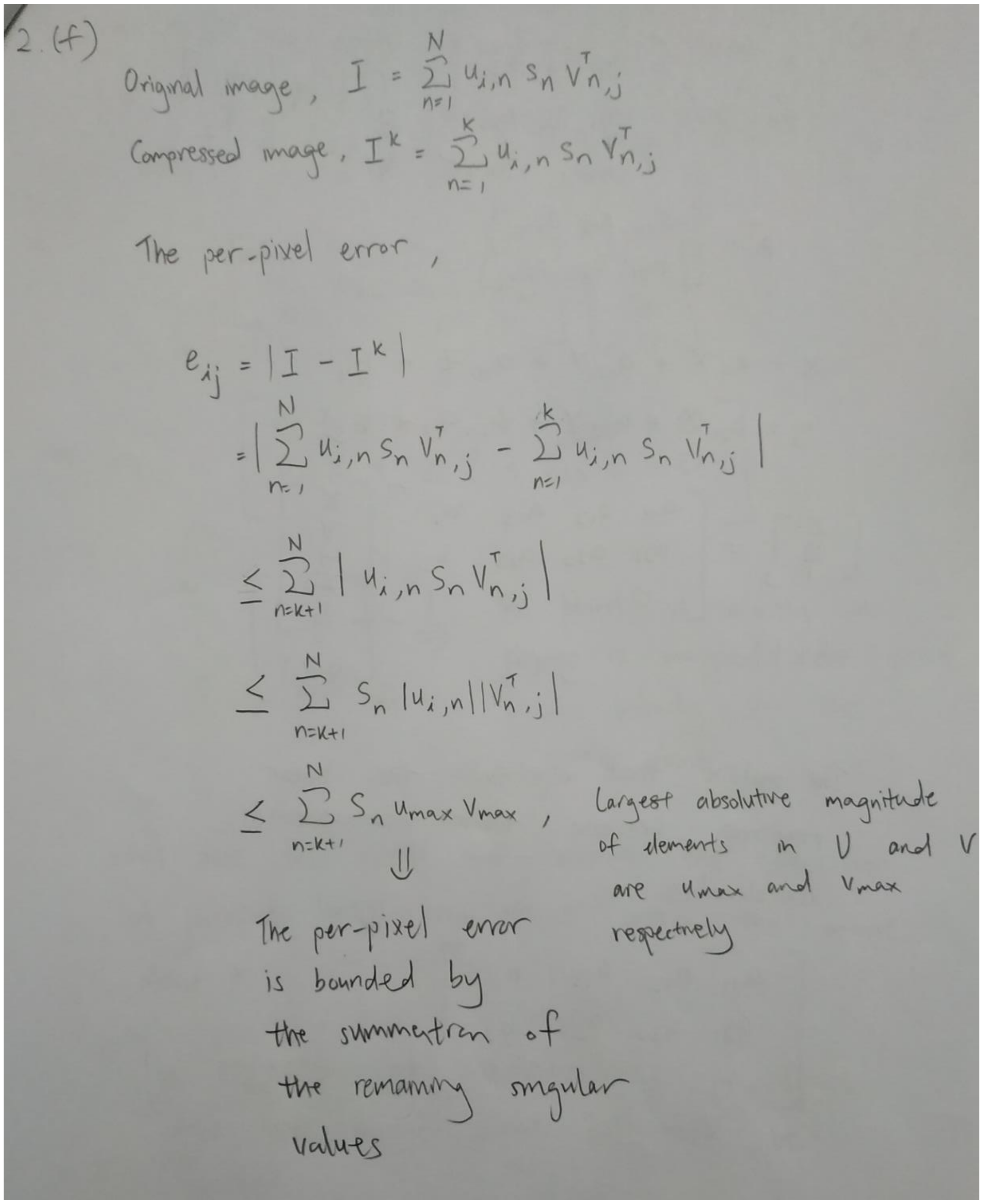
Figure 5: Compressed Image (K = 80)

With reference to the above images (figure 3, 4 and 5), when K increases, the quality of the image improves. The singular values decrease exponentially. It appears that only the first few singular values, around the first 10, and their respective eigenvectors approximate the original image well.

(e)

It is not worth transmitting when K = 100 because no bits is saved when transmitting at K = 100. When K = 100, 100 * (261+161+1) = 42300 values will be required for transmission. The original image has only 261 * 161 = 42021 values. As such, no compression is obtained when K = 100 and it is not worth transmitting at K = 100.

(f)



2.(f)

Original image, $I = \sum_{n=1}^{N} u_{i,n} s_n V^T_{n,j}$

Compressed image, $I^k = \sum_{n=1}^{k} u_{i,n} S_n V^T_{n,j}$

The per-pixel error,

$$e_{ij} = |I - I^k|$$

$$= \left| \sum_{n=1}^{N} u_{i,n} S_n V^T_{n,j} - \sum_{n=1}^{k} u_{i,n} S_n V^T_{n,j} \right|$$

$$\leq \sum_{n=k+1}^{N} | u_{i,n} S_n V^T_{n,j} |$$

$$\leq \sum_{n=k+1}^{N} S_n |u_{i,n}| |V^T_{n,j}|$$

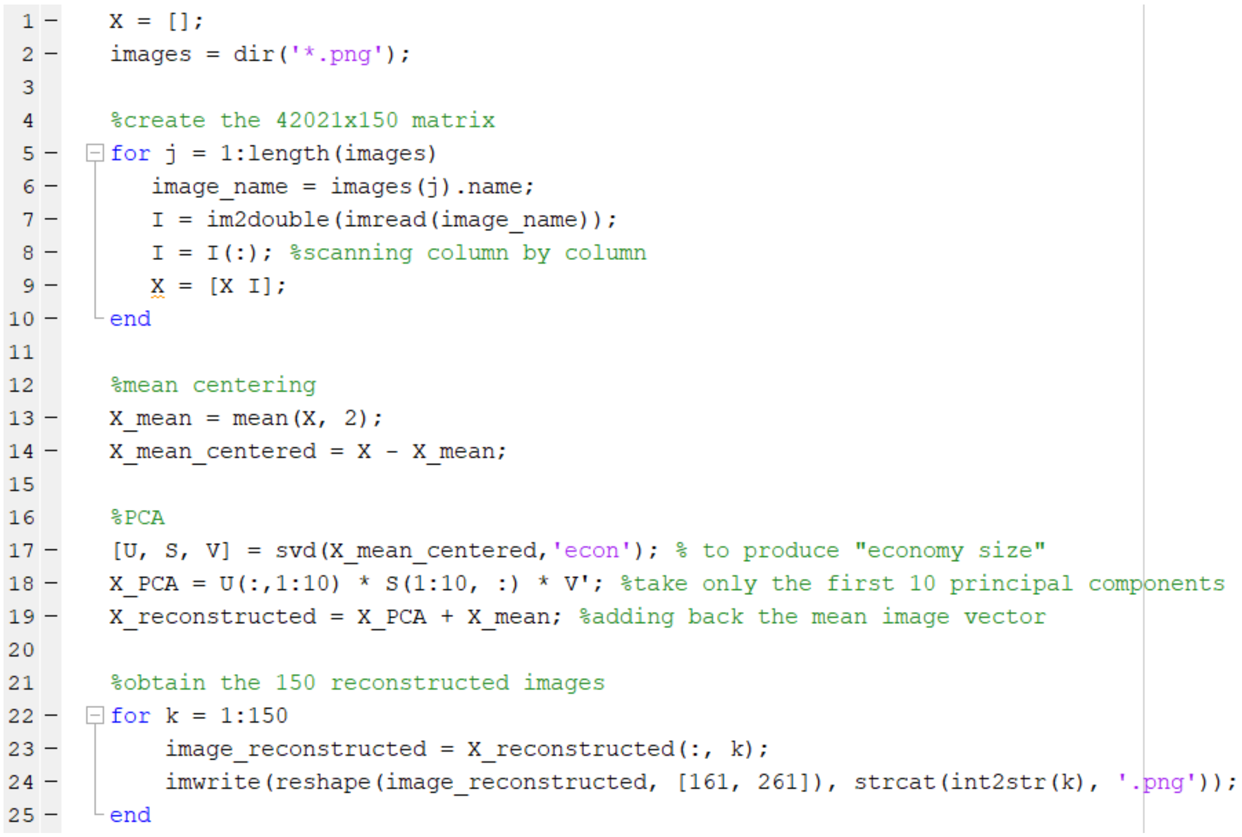
$$\leq \sum_{n=k+1}^{N} S_n u_{max} V_{max} ,$$

Largest absolute magnitude of elements in U and V are u_{max} and V_{max} respectnely

$\Downarrow$

The per-pixel error is bounded by the summatin of the remaming singular values

(g)



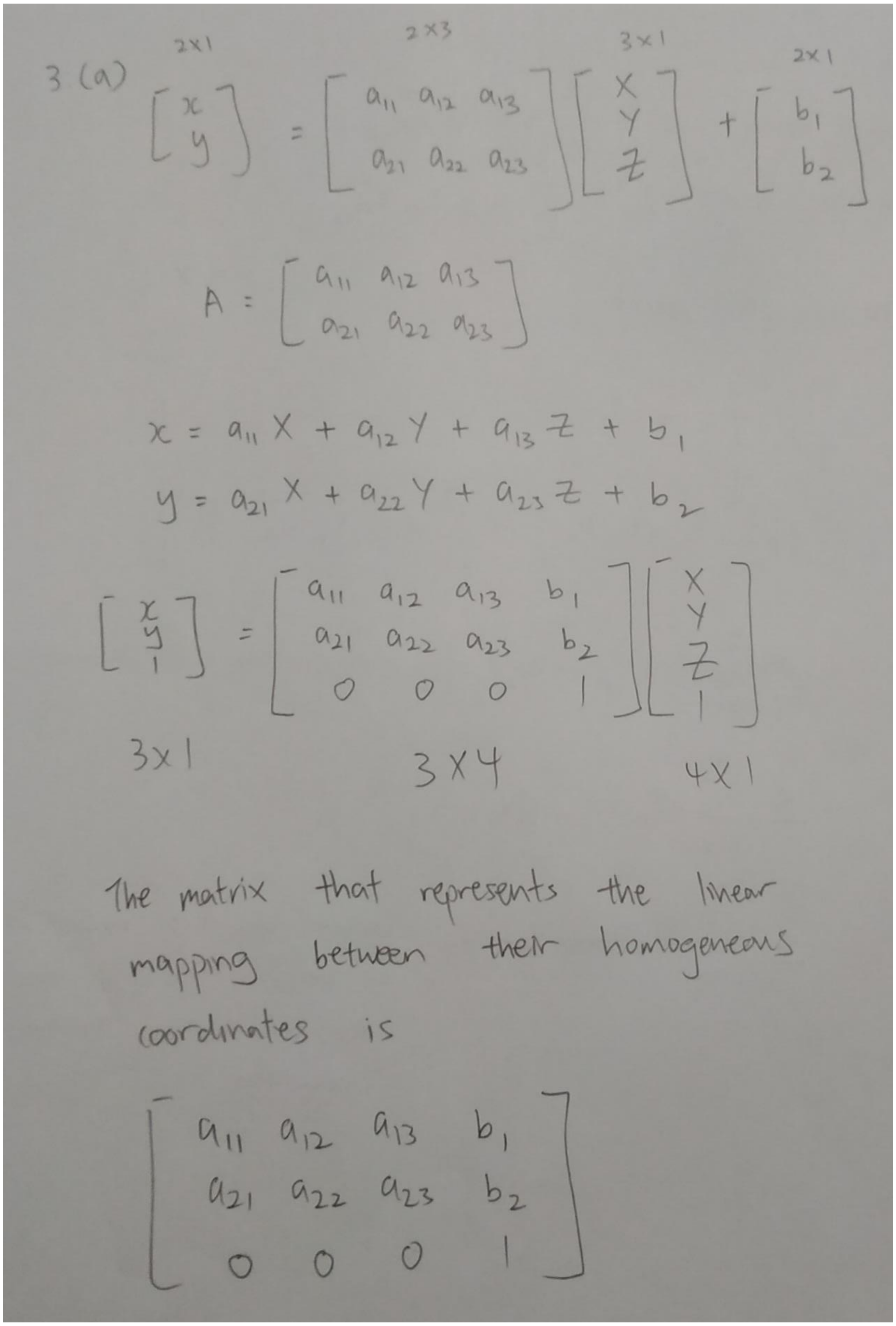
```
X = [];
images = dir('*.png');

%create the 42021x150 matrix
for j = 1:length(images)
   image_name = images(j).name;
   I = im2double(imread(image_name));
   I = I(:); %scanning column by column
   X = [X I];
end

%mean centering
X_mean = mean(X, 2);
X_mean_centered = X - X_mean;

%PCA
[U, S, V] = svd(X_mean_centered,'econ'); % to produce "economy size"
X_PCA = U(:,1:10) * S(1:10, :) * V'; %take only the first 10 principal components
X_reconstructed = X_PCA + X_mean; %adding back the mean image vector

%obtain the 150 reconstructed images
for k = 1:150
    image_reconstructed = X_reconstructed(:, k);
    imwrite(reshape(image_reconstructed, [161, 261]), strcat(int2str(k), '.png'));
end
```

In the reconstructed video, the images are blurrier compared to the original images. The movement of the clouds and the movement of the airplanes on the ground before taking off can still be observed. However, the taking off and landing of the airplanes cannot be observed in the reconstructed video.

Slow changes in the image sequence will require the rank of the reconstructed matrix to be small, while fast changes in the image sequence will require the rank of the reconstructed matrix to be large. In this assignment, only the first 10 principal components are used to reconstruct the image, fast changes in the image sequence will not be included in the reconstructed image. As such, slow changes in the image sequence such as the movement of the clouds will be preserved but fast changes in the image sequence such as taking off and landing of the planes will be blurry.

3. Projection Model; Homogeneous coordinates

(a)



3 (a)

$$\overset{2\times 1}{\begin{bmatrix} x \\ y \end{bmatrix}} = \overset{2\times 3}{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}} \overset{3\times 1}{\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}} + \overset{2\times 1}{\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

$$x = a_{11} X + a_{12} Y + a_{13} Z + b_1$$

$$y = a_{21} X + a_{22} Y + a_{23} Z + b_2$$

$$\underset{3\times 1}{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}} = \underset{3\times 4}{\begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}} \underset{4\times 1}{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}$$

The matrix that represents the linear mapping between their homogeneous coordinates is

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Show that the point at infinity in space $(X, Y, Z, 0)^T$ is mapped to point of infinity in the image plane

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 0 \end{bmatrix}$$

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

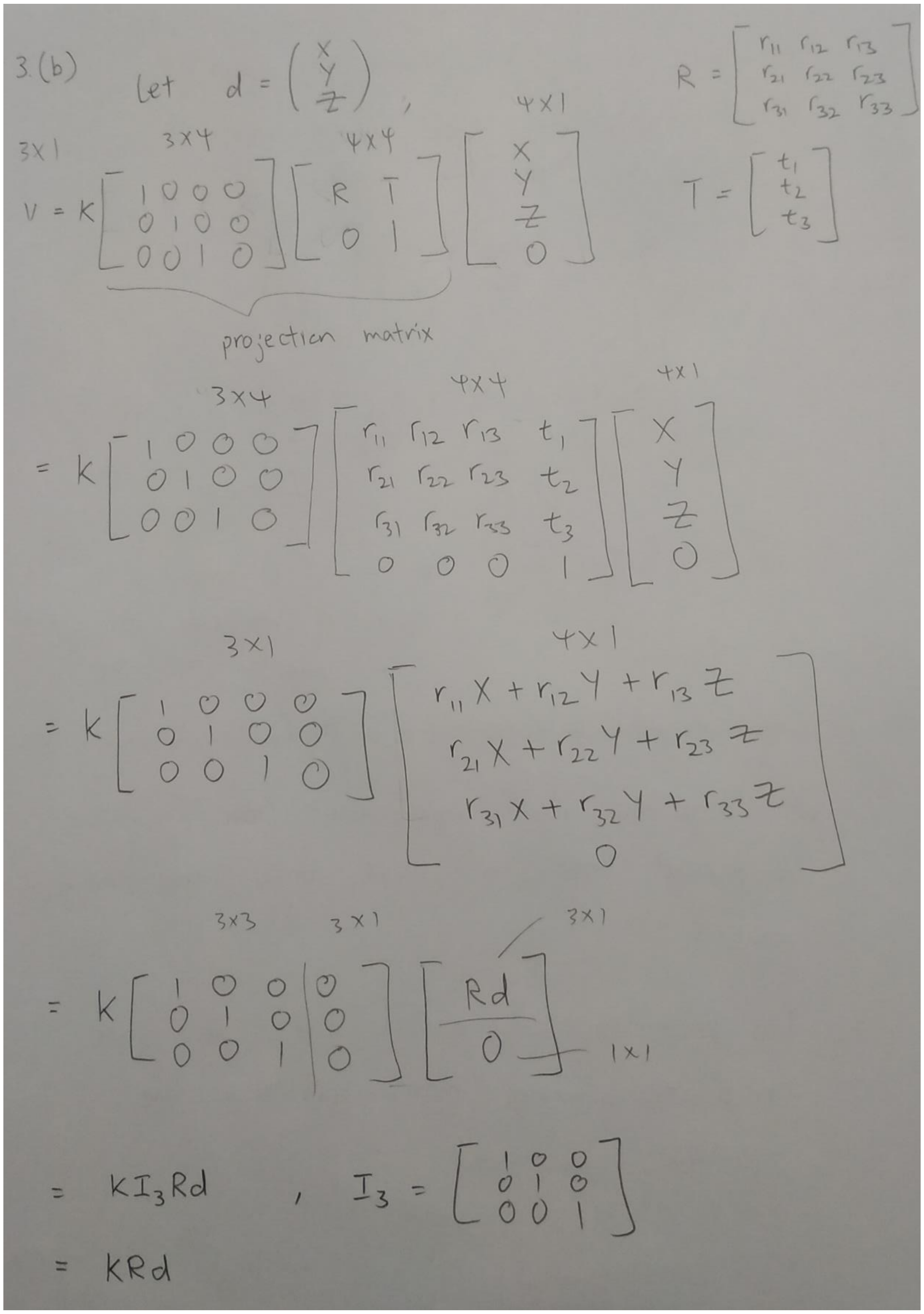
$$= \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \Rightarrow$$

The point at infinity in space is mapped to point of infinty in the image plane

What does this result imply about the projection of parallel lines in space onto the image plane?

The projection of parallel lines in space onto the image plane will also be parallel

(b)



3.(b) Let $d = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$, $R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$

$$\overset{3\times1}{V} = K \underbrace{\overset{3\times4}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}} \overset{4\times4}{\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}}}_{\text{projection matrix}} \overset{4\times1}{\begin{bmatrix} X \\ Y \\ Z \\ 0 \end{bmatrix}} \qquad T = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

$$= K \overset{3\times4}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}} \overset{4\times4}{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}} \overset{4\times1}{\begin{bmatrix} X \\ Y \\ Z \\ 0 \end{bmatrix}}$$

$$= K \overset{3\times1}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}} \overset{4\times1}{\begin{bmatrix} r_{11}X + r_{12}Y + r_{13}Z \\ r_{21}X + r_{22}Y + r_{23}Z \\ r_{31}X + r_{32}Y + r_{33}Z \\ 0 \end{bmatrix}}$$

$$= K \left[\overset{3\times3}{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}} \middle| \overset{3\times1}{\begin{matrix} 0 \\ 0 \\ 0 \end{matrix}} \right] \begin{bmatrix} \overset{3\times1}{Rd} \\ \hline \underset{1\times1}{0} \end{bmatrix}$$

$$= K I_3 R d \quad , \quad I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= KRd$$

$v = KRd$

$d = R^{-1} K^{-1} v$, K & R are non-singular (invertible)

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.

$d_1 \cdot d_2 = 0$

$\begin{pmatrix} 1 \\ 0 \\ 4 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} = 0$

$3\times1 \quad 3\times1$

$(1 \ 0 \ 4) \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} = 0$

$1\times3 \quad 3\times1$

$$d_1 \cdot d_2 = 0$$

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

$$(d_1)^T (R^T R) d_2 = 0 , \quad R^T R = I$$

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

$$(v_1 K^{-1})^T (K^{-1} v_2) = 0 , \quad Rd = K^{-1} v$$

$$v_1^T K^{-T} K^{-1} v_2 = 0$$

$$\begin{bmatrix} \frac{-1}{s_x} & k & o_x \\ 0 & \frac{-1}{s_y} & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

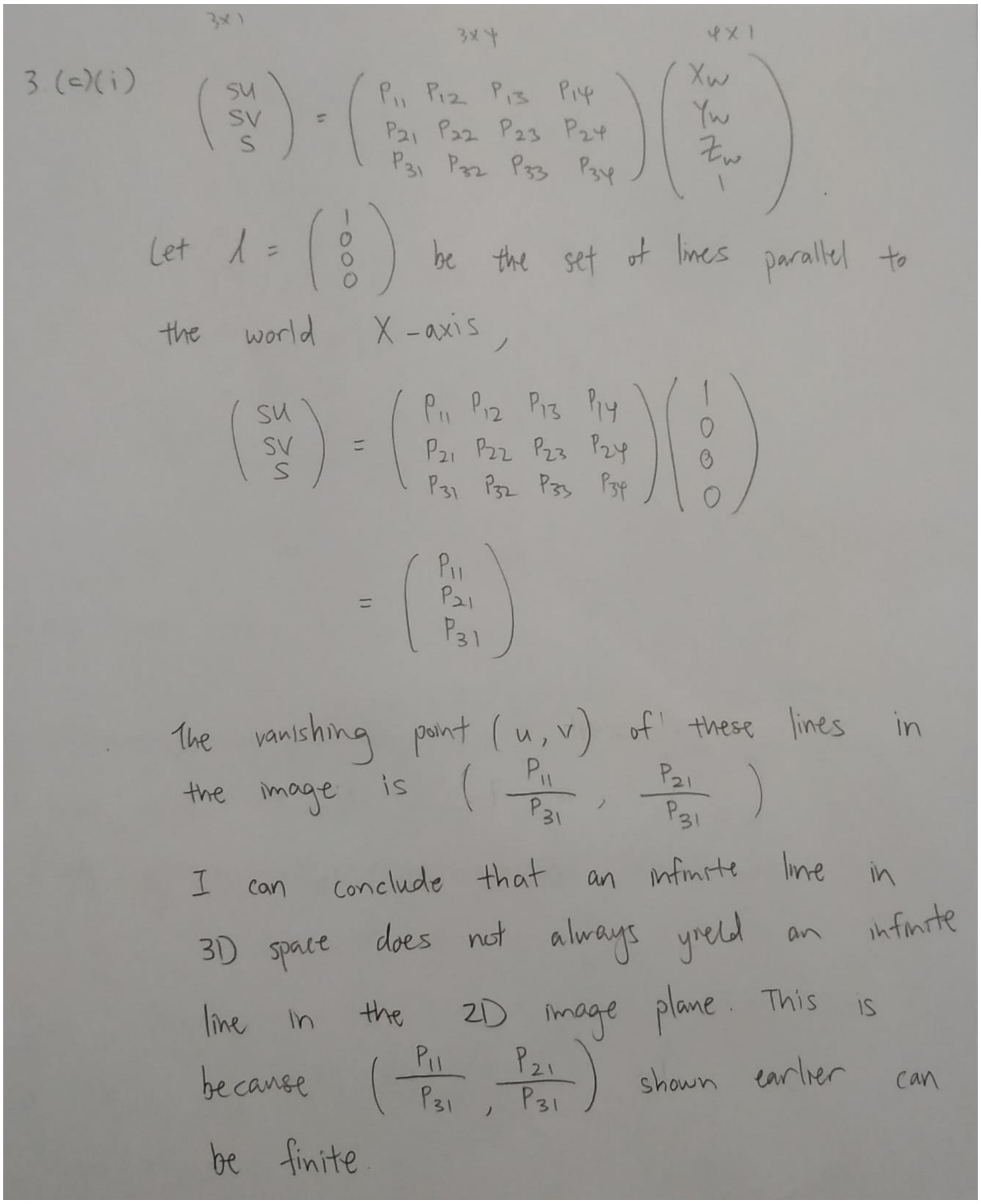
Assuming that there is zero additional skew factor k and we are looking at a square pixel,

$$k = 0$$
$$s_x = s_y$$

$$\therefore K = \begin{bmatrix} \frac{-1}{s_x} & 0 & o_x \\ 0 & \frac{-1}{s_x} & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

In the above matrix, there is 3 equations and 3 unknowns ($\frac{-1}{s_x}$, o_x, o_y), solving K is possible.

(c)



3 (c)(i)

$$\underset{3\times 1}{\begin{pmatrix} su \\ sv \\ s \end{pmatrix}} = \underset{3\times 4}{\begin{pmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{pmatrix}} \underset{4\times 1}{\begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}}.$$

Let $l = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ be the set of lines parallel to the world X-axis,

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{pmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} P_{11} \\ P_{21} \\ P_{31} \end{pmatrix}$$

The vanishing point (u, v) of these lines in the image is $\left(\frac{P_{11}}{P_{31}}, \frac{P_{21}}{P_{31}} \right)$

I can conclude that an infinite line in 3D space does not always yield an infinite line in the 2D image plane. This is because $\left(\frac{P_{11}}{P_{31}}, \frac{P_{21}}{P_{31}} \right)$ shown earlier can be finite.

(c)(ii) Let $X_w = 0$, $Y_w = 0$ & $Z_w = 0$

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

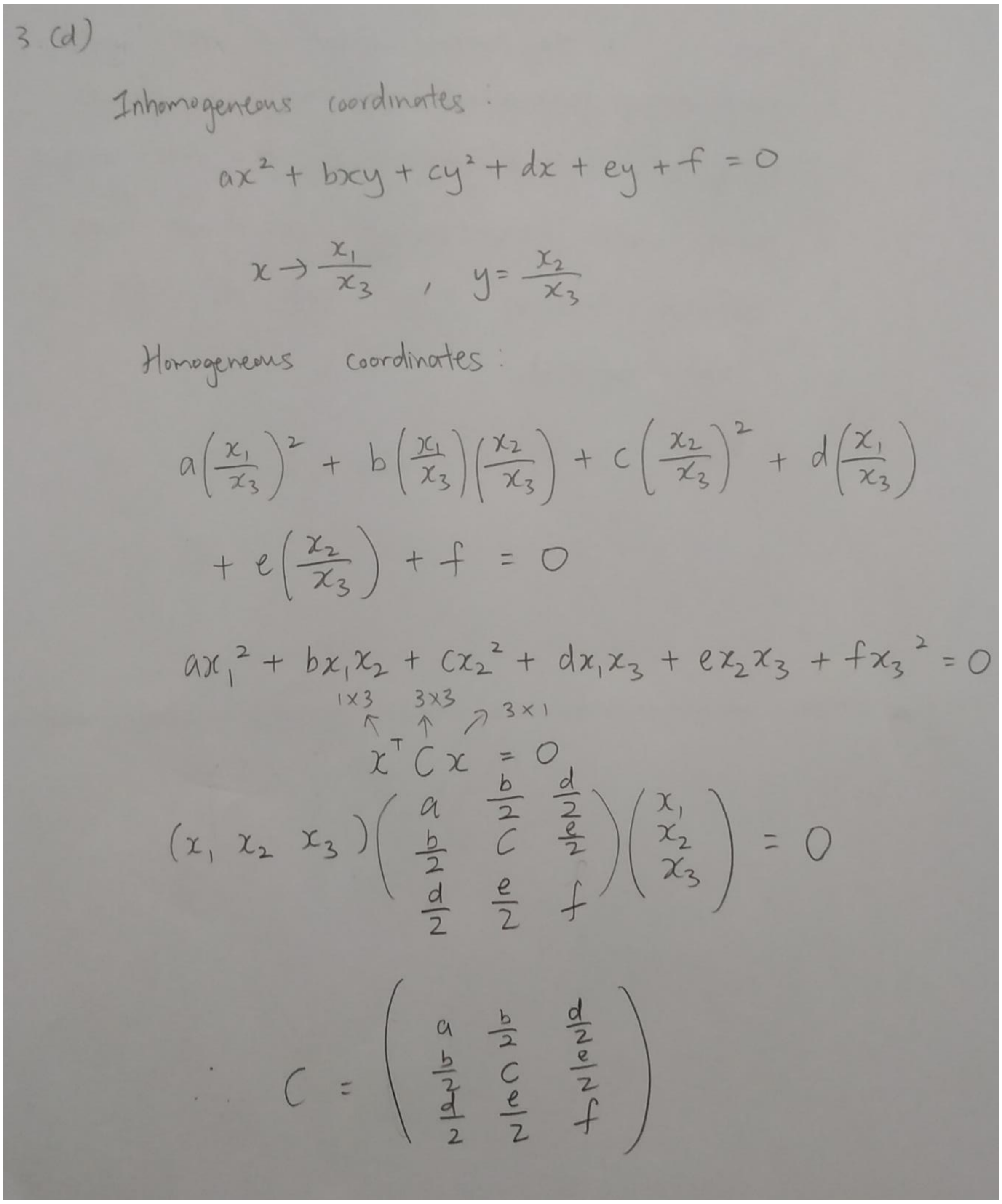
$$= \begin{pmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} P_{14} \\ P_{24} \\ P_{34} \end{pmatrix}$$

$(0,0,0,1)$ is the world point that gives rise to (P_{14}, P_{24}, P_{34}).
The significance of the image point is that it is the image point which the translation vector cuts the image plane.

(c)(iii) Point C is the camera center in the world coordinates represented as a homogeneous 4 vector. In addition, it is the unique point where its image is not well defined because the line of sight is not defined.

(d)



3. (d)

Inhomogeneous coordinates :

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

$$x \rightarrow \frac{x_1}{x_3} \quad , \quad y = \frac{x_2}{x_3}$$

Homogeneous coordinates :

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

$$+ e\left(\frac{x_2}{x_3}\right) + f = 0$$

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

$$\underset{1\times 3}{x^T}\ \underset{3\times 3}{C}\ \underset{3\times 1}{x} = 0$$

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

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

$Cx = (l m^T + m l^T)x$, $C = l m^T + m l^T$

$= l m^T x + m l^T x$

In homogeneous coordinates, $x = l \times m$ (intersection of line m and line l)

$$m^T x = 0 , \quad l^T x = 0$$

$\therefore$ $Cx = l(0) + m(0)$

$= 0 \Rightarrow x = l \times m$ is the null vector of C

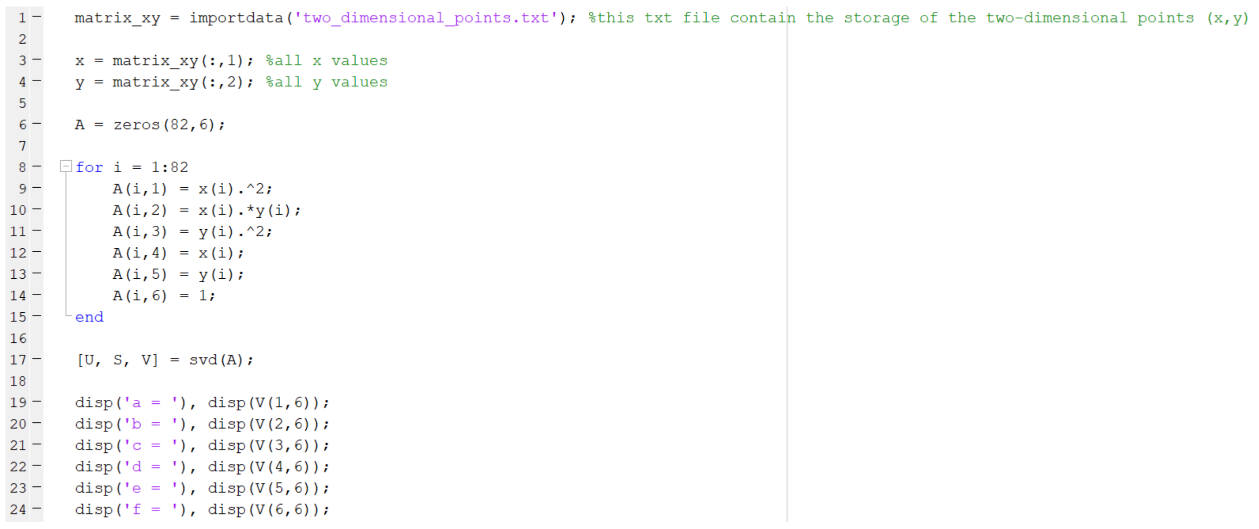
Points on l satisfy $l^T x = 0$ and points on m satisfy $m^T x = 0$,

On the conic, since

$$x^T C x = x^T l \cancelto{0}{m^T x} + x^T m \cancelto{0}{l^T x}$$

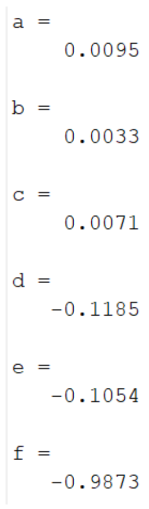
$$= 0$$

$\therefore$ Points on the lines l and m satisfy $x^T C x = 0$



```
matrix_xy = importdata('two_dimensional_points.txt'); %this txt file contain the storage of the two-dimensional points (x,y)

x = matrix_xy(:,1); %all x values
y = matrix_xy(:,2); %all y values

A = zeros(82,6);

for i = 1:82
    A(i,1) = x(i).^2;
    A(i,2) = x(i).*y(i);
    A(i,3) = y(i).^2;
    A(i,4) = x(i);
    A(i,5) = y(i);
    A(i,6) = 1;
end

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

disp('a = '), disp(V(1,6));
disp('b = '), disp(V(2,6));
disp('c = '), disp(V(3,6));
disp('d = '), disp(V(4,6));
disp('e = '), disp(V(5,6));
disp('f = '), disp(V(6,6));
```

(e)



```
a =
    0.0095

b =
    0.0033

c =
    0.0071

d =
   -0.1185

e =
   -0.1054

f =
   -0.9873
```

4. Estimate Fundamental Matrix F using 8-Point Algorithm

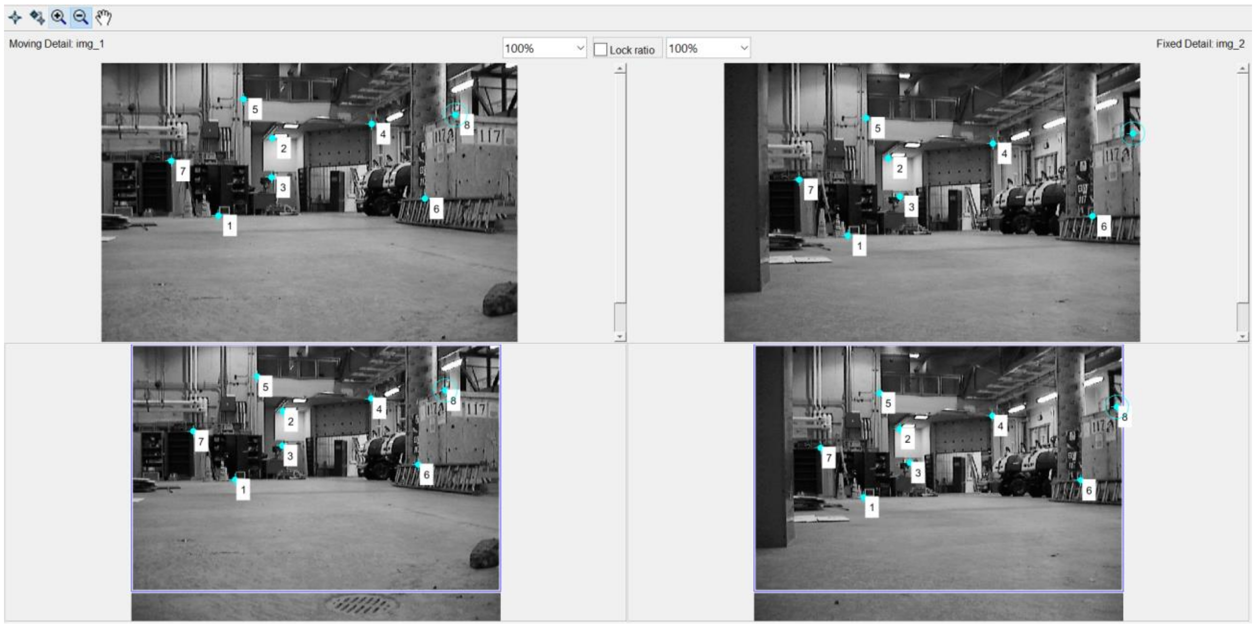
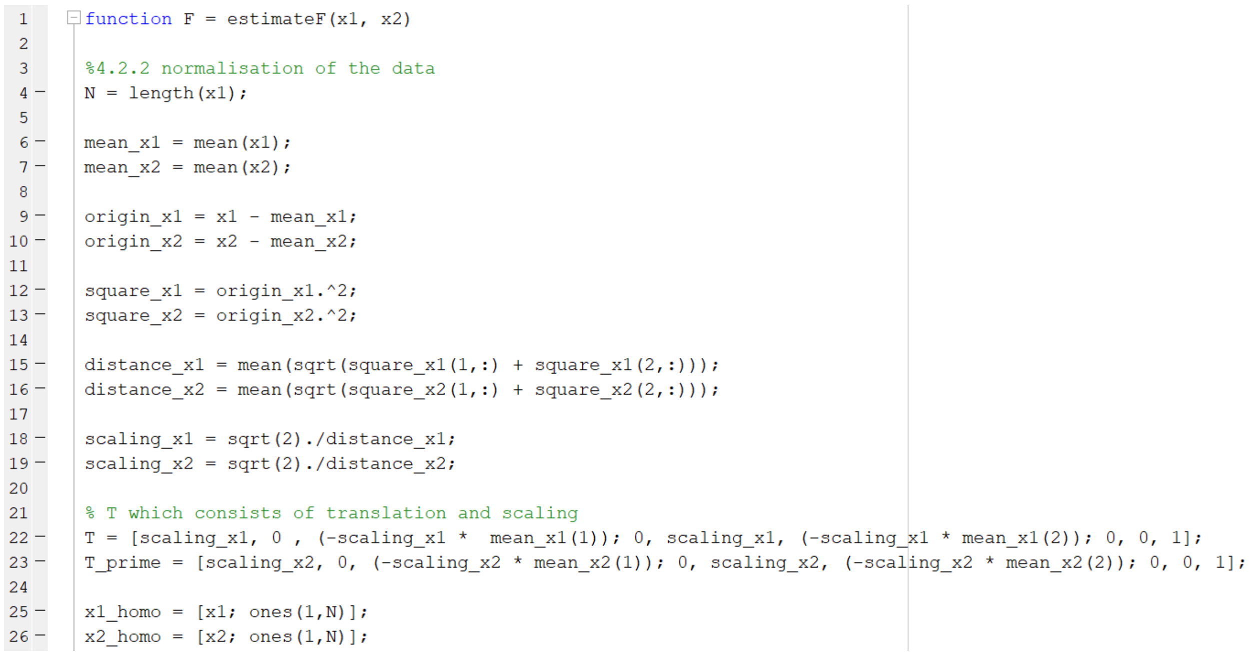
(1) 

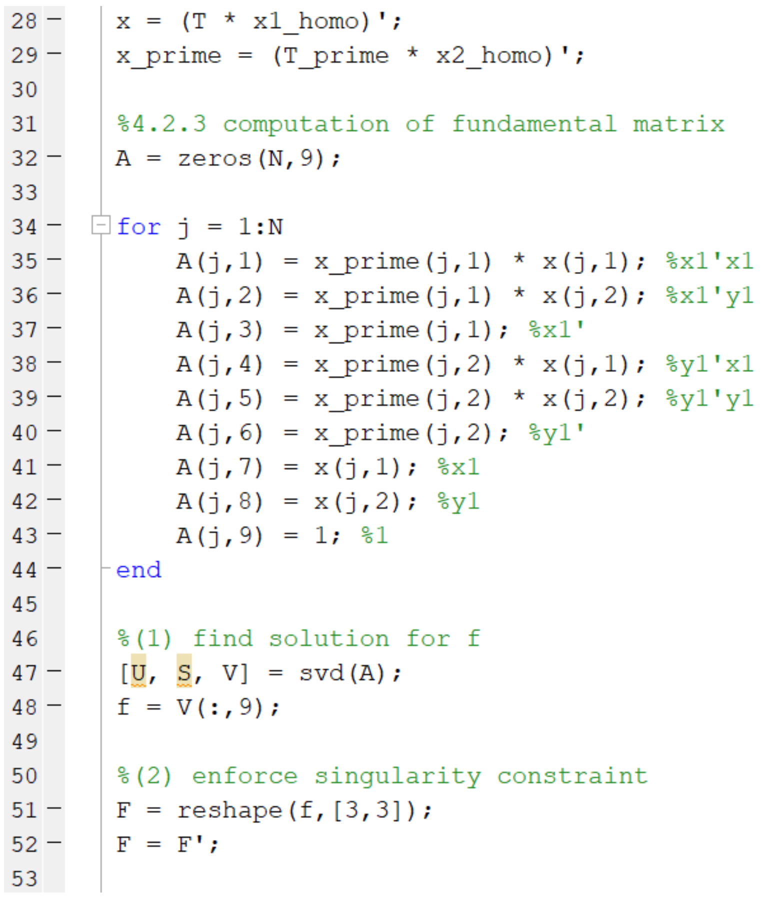
Figure 6: 8 corresponding points

The corresponding points chosen are points on corners and prominent features. These corresponding points are chosen because they are easy to locate in both images.

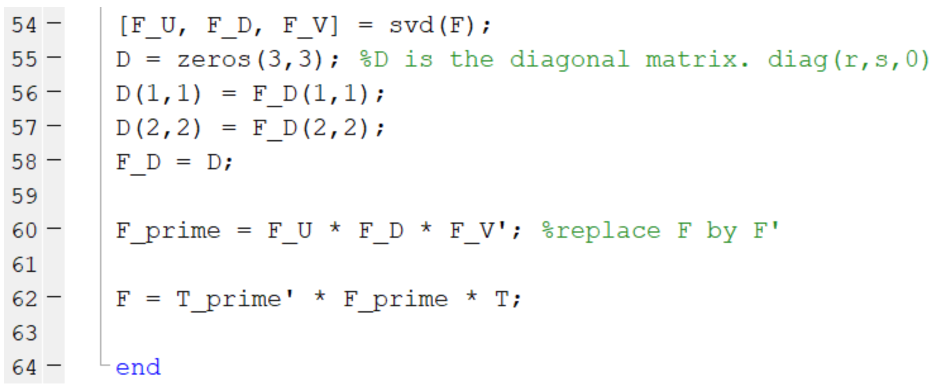
(2)
```
%4.2.1 establish point correspondence
img_1 = imread('frc1.tif');
img_2 = imread('frc2.tif');

%cpselect(img_1, img_2);

%x = cpstruct8.inputPoints'; %8 corresponding points
%x_prime = cpstruct8.basePoints'; %8 corresponding points

%x = cpstruct12.inputPoints'; %12 corresponding points
%x_prime = cpstruct12.basePoints'; $12 corresponding points

x = cpstruct16.inputPoints'; %16 corresponding points
x_prime = cpstruct16.basePoints'; %16 corresponding points

%4.2.2, 4.2.3 and 4.2.4
F = estimateF(x, x_prime);
disp("F = "), disp(F);
displayEpipolarF(img_1, img_2, F);
```



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

%4.2.2 normalisation of the data
N = length(x1);

mean_x1 = mean(x1);
mean_x2 = mean(x2);

origin_x1 = x1 - mean_x1;
origin_x2 = x2 - mean_x2;

square_x1 = origin_x1.^2;
square_x2 = origin_x2.^2;

distance_x1 = mean(sqrt(square_x1(1,:) + square_x1(2,:)));
distance_x2 = mean(sqrt(square_x2(1,:) + square_x2(2,:)));

scaling_x1 = sqrt(2)./distance_x1;
scaling_x2 = sqrt(2)./distance_x2;

% T which consists of translation and scaling
T = [scaling_x1, 0 , (-scaling_x1 *  mean_x1(1)); 0, scaling_x1, (-scaling_x1 * mean_x1(2)); 0, 0, 1];
T_prime = [scaling_x2, 0, (-scaling_x2 * mean_x2(1)); 0, scaling_x2, (-scaling_x2 * mean_x2(2)); 0, 0, 1];

x1_homo = [x1; ones(1,N)];
x2_homo = [x2; ones(1,N)];
```



```matlab
x = (T * x1_homo)';
x_prime = (T_prime * x2_homo)';

%4.2.3 computation of fundamental matrix
A = zeros(N,9);

for j = 1:N
    A(j,1) = x_prime(j,1) * x(j,1); %x1'x1
    A(j,2) = x_prime(j,1) * x(j,2); %x1'y1
    A(j,3) = x_prime(j,1); %x1'
    A(j,4) = x_prime(j,2) * x(j,1); %y1'x1
    A(j,5) = x_prime(j,2) * x(j,2); %y1'y1
    A(j,6) = x_prime(j,2); %y1'
    A(j,7) = x(j,1); %x1
    A(j,8) = x(j,2); %y1
    A(j,9) = 1; %1
end

%(1) find solution for f
[U, S, V] = svd(A);
f = V(:,9);

%(2) enforce singularity constraint
F = reshape(f,[3,3]);
F = F';
```



```
[F_U, F_D, F_V] = svd(F);
D = zeros(3,3); %D is the diagonal matrix. diag(r,s,0)
D(1,1) = F_D(1,1);
D(2,2) = F_D(2,2);
F_D = D;

F_prime = F_U * F_D * F_V'; %replace F by F'

F = T_prime' * F_prime * T;

end
```

(3)

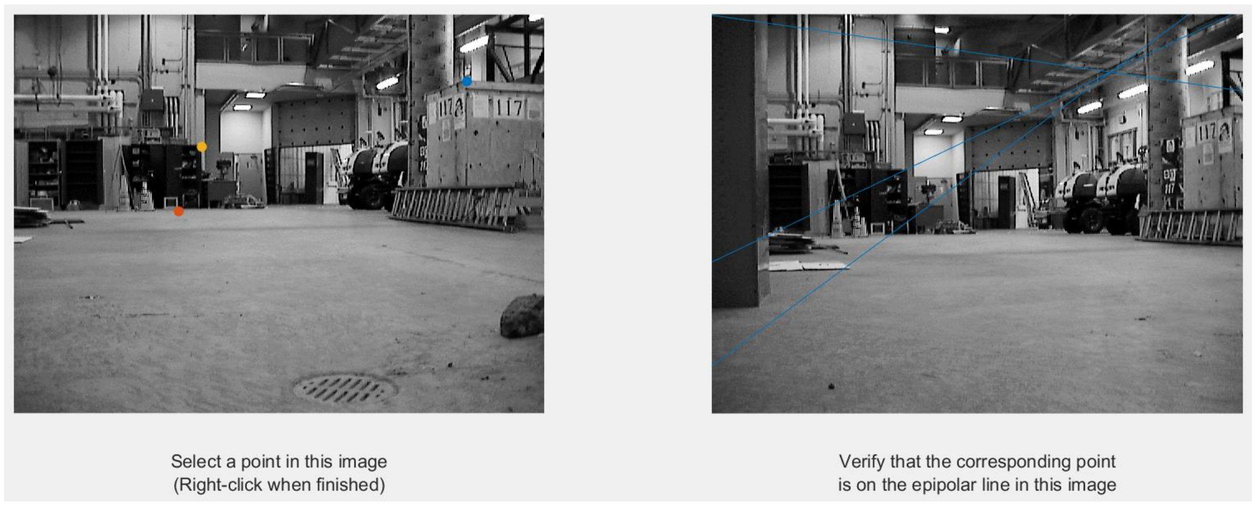
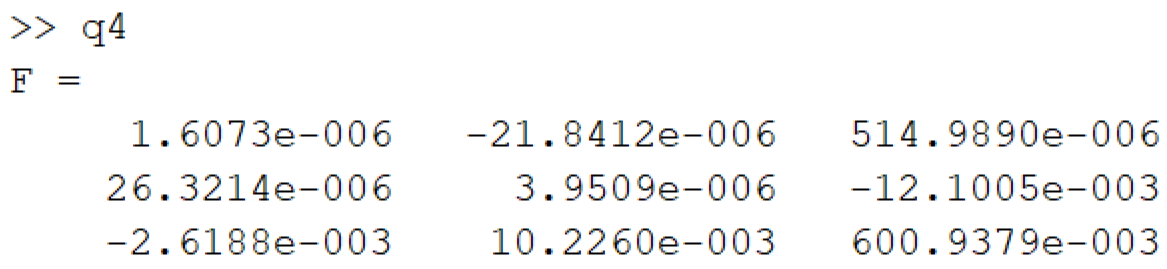

Select a point in this image
(Right-click when finished)
Verify that the corresponding point
is on the epipolar line in this image

Figure 7: Epipolar Lines for 8 Corresponding Points



```
>> q4
F =

     1.6073e-006   -21.8412e-006   514.9890e-006
    26.3214e-006     3.9509e-006   -12.1005e-003
    -2.6188e-003    10.2260e-003   600.9379e-003
```

Figure 8: Estimated Fundamental Matrix F for 8 Corresponding Points

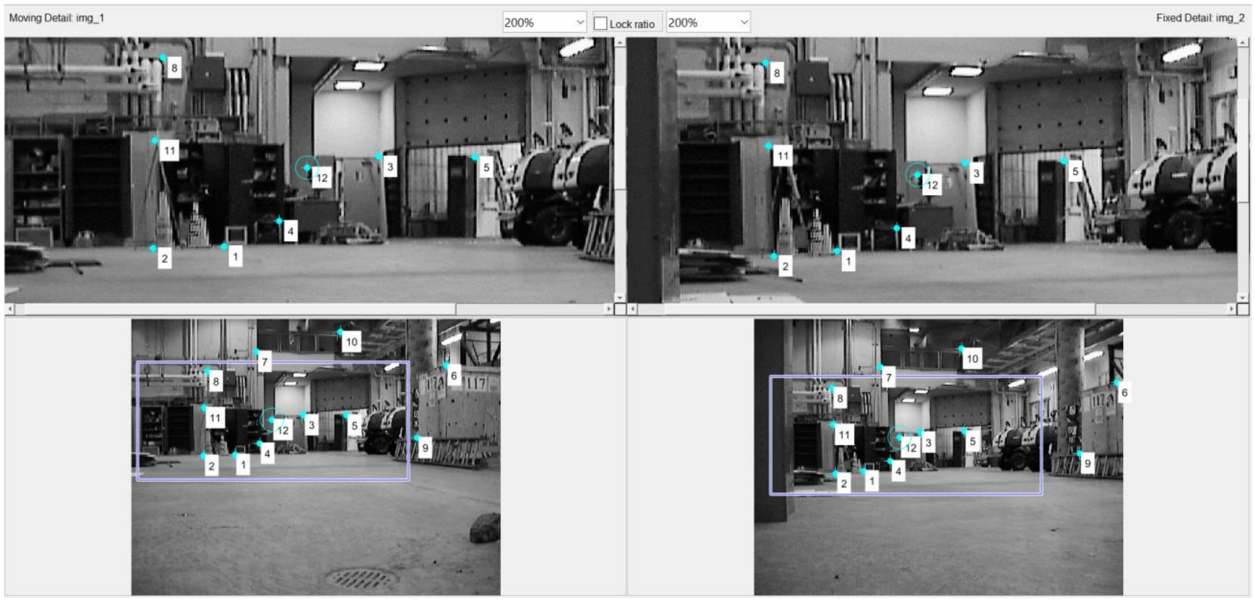
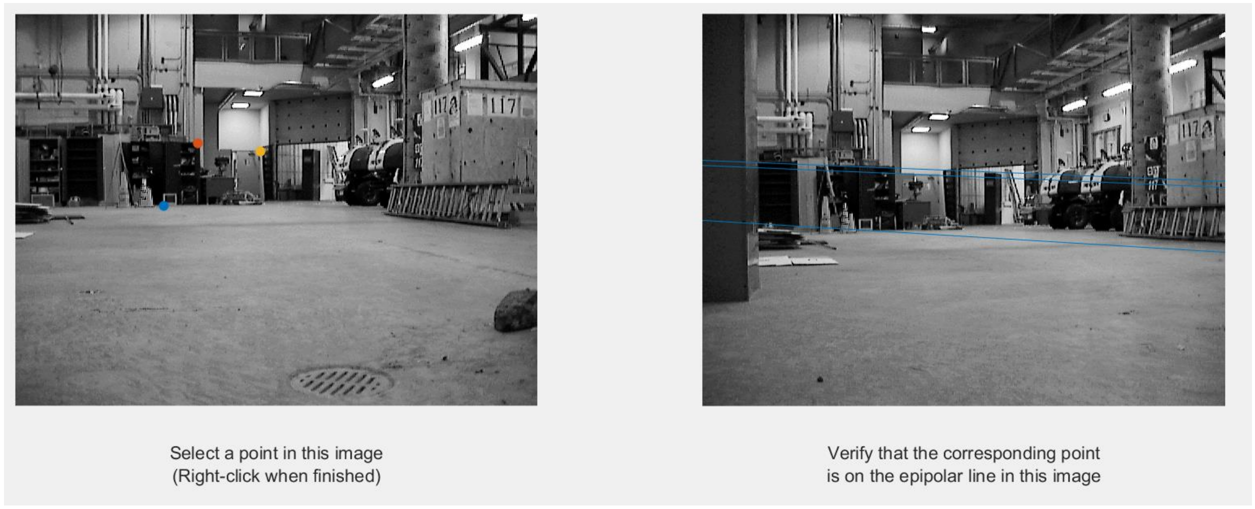


Figure 9: 12 Corresponding Points



Select a point in this image
(Right-click when finished)

Verify that the corresponding point
is on the epipolar line in this image

Figure 10: Epipolar Lines for 12 Corresponding Points

```
F =

    69.1476e-009     2.4654e-006   -44.5901e-006
    -4.1378e-006    -2.2689e-006    -7.7042e-003
   398.1067e-006     8.8254e-003   126.6449e-003
```

Figure 11: Estimated Fundamental Matrix F for 12 Corresponding Points

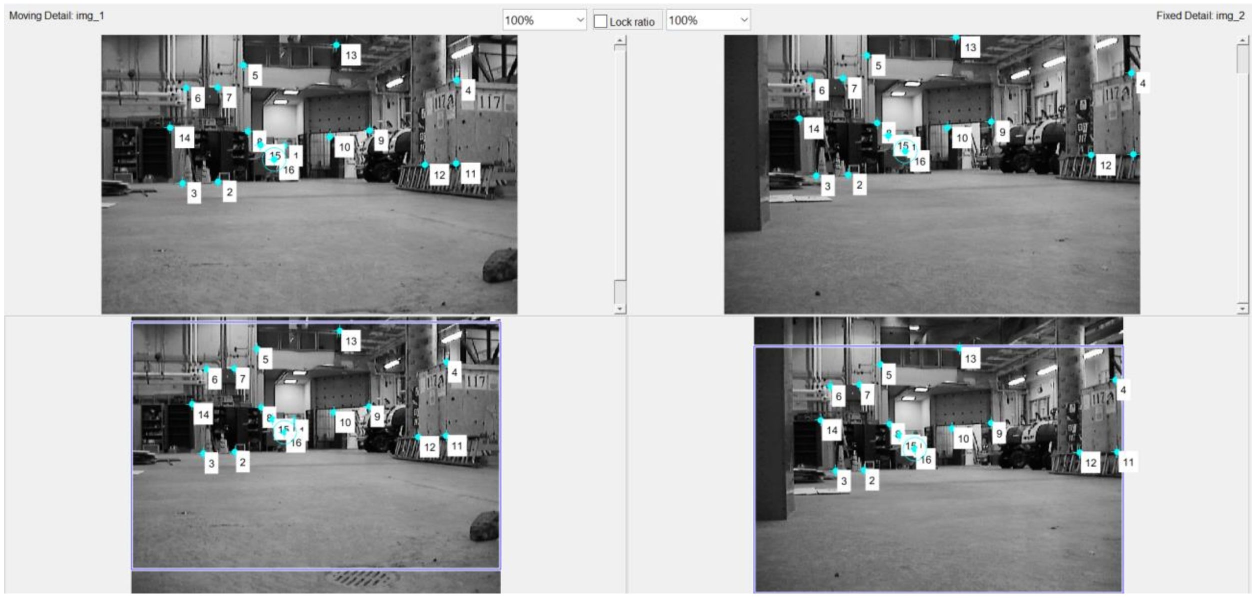
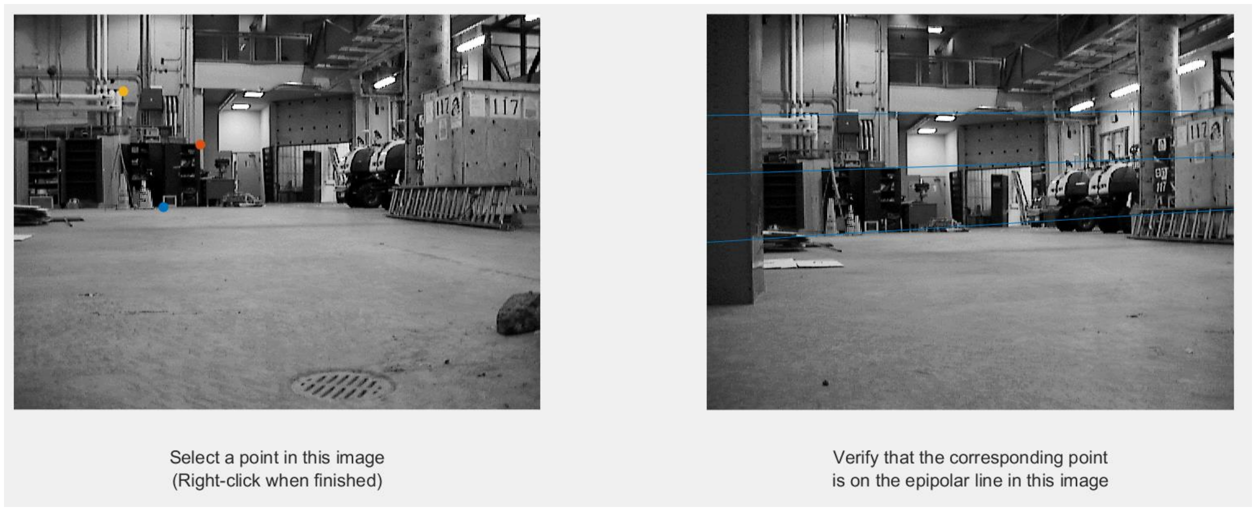


Figure 12: 16 Corresponding Points


Select a point in this image
(Right-click when finished)
Verify that the corresponding point
is on the epipolar line in this image
Figure 13: Epipolar Lines for 16 Corresponding Points

```
F =

   -11.7411e-009      3.2029e-006   -217.4278e-006
    -3.0675e-006     -1.8526e-006      9.4552e-003
   330.8584e-006     -9.0236e-003   -291.1141e-003
```

Figure 14: Estimated Fundamental Matrix F for 16 Corresponding Points

(4) In the previous question, different sets of corresponding points are used. The accuracy of the result improves when more pairs of corresponding points are selected in both images. This can be observed where the corresponding epipolar lines in the right image go through the matched points selected. In addition, the inaccuracy of the results obtained in (3) might

be due to the error in locating the corresponding points manually using cpselect function in MATLAB. Next, error will arise due to the linear least square method used which ignores non-linear constraint such as the rank 2 constraint. Although singularity constraint is enforced in the question, this does not resolve the issue. Furthermore, the structure of the scene is near a planar scene. This will result in ambiguity when selecting the corresponding points, which will lead to more serious inaccuracy of the result because of choosing the wrong corresponding points that clustered in small region. Lastly, the error is not well normalized. The residual of the matrix minimized does not contain any proper explanation.

In conclusion, to achieve better results, RANSAC can be implemented and more pairs of corresponding points should be selected.

Q5. Motion Perception

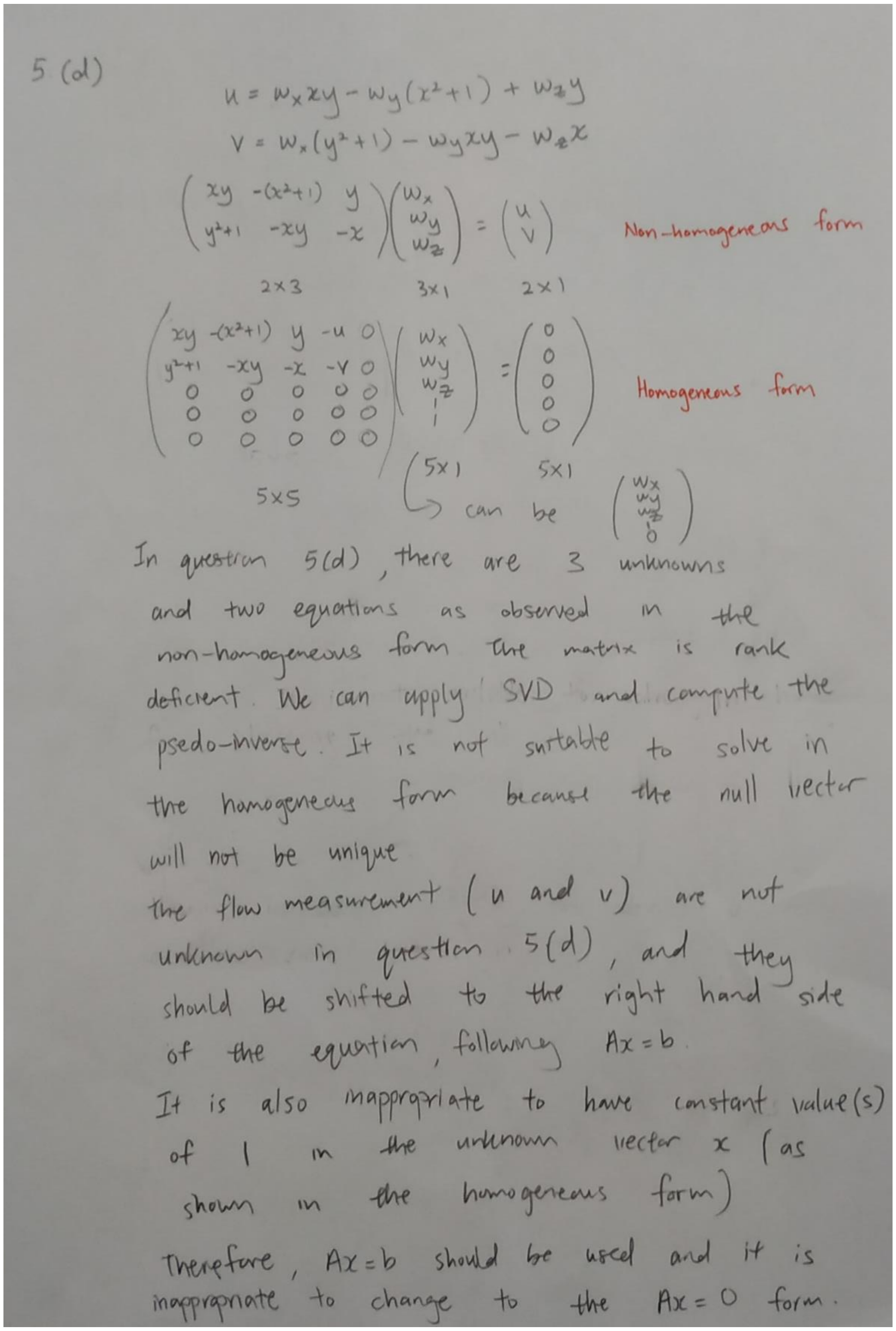
(a) At aperture 1, the perceived motion appears to move diagonally. However, this perceived motion is consistent with many other motions. As such, the perceived motion is ambiguous as a result of the aperture problem.
At aperture 2, the perceived motion is unambiguously horizontal. This is because the corners of the two moving squares are 2D features. These features help to disambiguate the motion of the edges.
At aperture 3, the perceived motion is unambiguously vertical. However, their motion is spurious as the corners of the two moving squares are not actual physical corners.

(b) In a barber-pole, the diagonal stripes are spinning between the occluders. The human visual system seems to know that the line terminators aligned with the occluders are due to occlusion by another surface instead of the end of the line itself. As such, the motion signals of these line terminations are ambiguous and, likely to become suppressed and have less influence on the perceived motion. Therefore, the perceived motion tends to be biased in the direction orthogonal to the occlusion boundary.

(c) In Figure c, when the two orthogonal bars are presented together within an occluding aperture, T-junctions are formed at the endpoints of the two orthogonal bars. These T-junctions signal a cue that the motion at the endpoints of the two orthogonal bars are spurious as a result of occlusion, and these motion at the T-junctions will have less influence on our perception. As such, the motions at the endpoints of the two orthogonal bars are ignored by the visual system. Disregarding the motions at the endpoints of the two orthogonal bars, the remaining local motions of the two orthogonal bars are consistent with a single circular motion.

In Figure d, when the occluding aperture is removed, T-junctions are not produced. The endpoints of the two orthogonal bars are clearly observed by the visual system and there is unambiguous motion which decides the perceived motion of the two orthogonal bars as seen in Figure d.

(d)



5 (d)

$$u = w_x xy - w_y(x^2+1) + w_z y$$

$$v = w_x(y^2+1) - w_y xy - w_z x$$

$$\underbrace{\begin{pmatrix} xy & -(x^2+1) & y \\ y^2+1 & -xy & -x \end{pmatrix}}_{2\times 3} \underbrace{\begin{pmatrix} w_x \\ w_y \\ w_z \end{pmatrix}}_{3\times 1} = \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{2\times 1}$$

Non-homogeneous form

$$\underbrace{\begin{pmatrix} xy & -(x^2+1) & y & -u & 0 \\ y^2+1 & -xy & -x & -v & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{5\times 5} \underbrace{\begin{pmatrix} w_x \\ w_y \\ w_z \\ 1 \\ 1 \end{pmatrix}}_{5\times 1} = \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{5\times 1}$$

Homogeneous form

$\hookrightarrow$ can be $\begin{pmatrix} w_x \\ w_y \\ w_z \\ 1 \\ 0 \end{pmatrix}$

In question 5(d), there are 3 unknowns and two equations as observed in the non-homogeneous form. The matrix is rank deficient. We can apply SVD and compute the psedo-inverse. It is not suitable to solve in the homogeneous form because the null vector will not be unique

The flow measurement (u and v) are not unknown in question 5(d), and they should be shifted to the right hand side of the equation, following $Ax = b$.

It is also inappropriate to have constant value(s) of 1 in the unknown vector x (as shown in the homogeneous form)

Therefore, $Ax = b$ should be used and it is inappropriate to change to the $Ax = 0$ form.