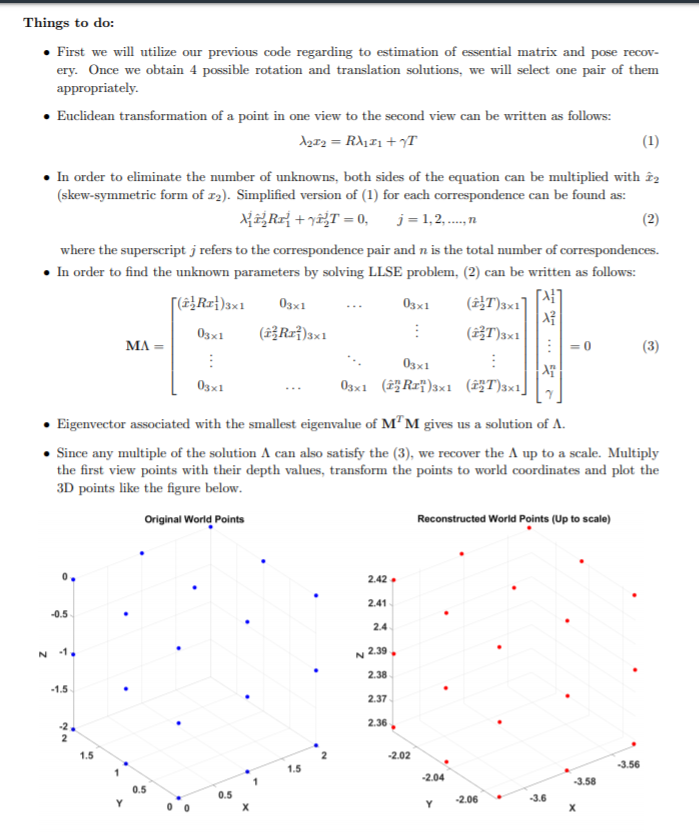
**EE417**

**POST-LAB #9 REPORT**

**Name: Nidanur GUNAY**

**ID:2423**

****

In this lab we recovered 3D structure from two view by obtaining 4 possible rotation and translation solutions and then we selected one pair of them appropriately. The steps that we need to follow were explained in detail in the lab document.

x1=p1;

x2=p2;

M=zeros(57, 20 );

* Initially, I’ve constructed 57x 20 matrix that is filled up all zeros. Since there were 19 points and each world coordinates corresponds to 3x1 vector, row size supposed to 57 and additional 1 columns for the translation vector.

for i = 1:1:19

%% skew

tempx2=x2(1:3,i:i);

skewx2=zeros(3,3);

skewx2(1,1)=0;

skewx2(2,2)=0;

skewx2(3,3)=0;

skewx2(1,2)=-(tempx2(3));

skewx2(2,1)=(tempx2(3));

skewx2(1,3)=tempx2(2);

skewx2(3,1)= -(tempx2(2));

skewx2(2,3)=-(tempx2(1));

skewx2(3,2)=(tempx2(1));

M(i\*3-2:i\*3,i) = skewx2\*R\*x1(:,i);

M(i\*3-2:i\*3,20) = skewx2\*T;

end

* In that step, I’ve filled the matrix according to given formula. Firstly I’ve calculated the skew symmetric form of x2 and then I’ve applied to formula.

[U S V] = svd(M'\*M);

l = V(:,end);

A = zeros(4,19);

for i=1:1:19

A(1,i) = x1(1,i)\*l(i);

A(2,i) = x1(2,i)\*l(i);

A(3,i) = x1(3,i)\*l(i);

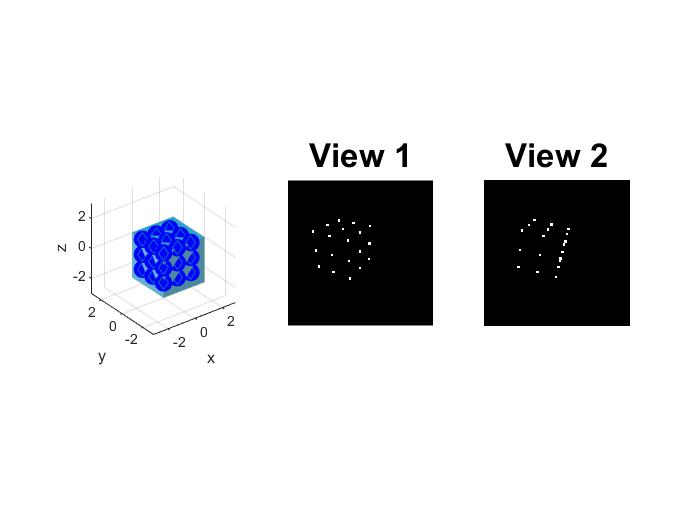
A(4,i) = 1;

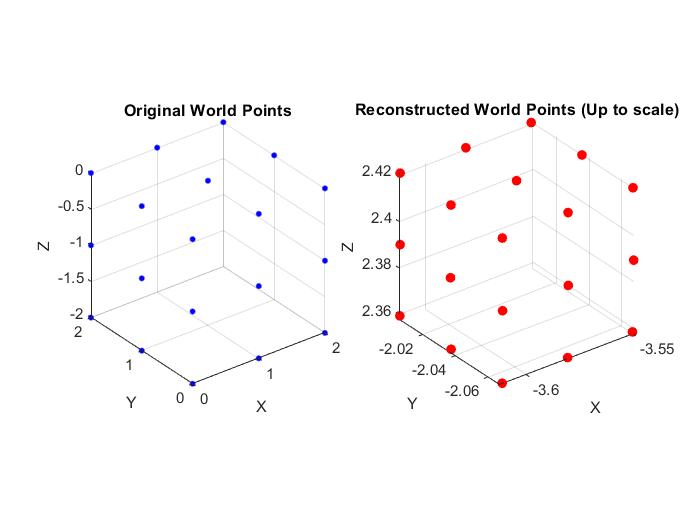
end

A = inv(Hc1)\*A;

* By the help of SVD decomposition, I’ve calculated the eigenvalues and real world coordinates were covered.

**RESULTS**

****

****

**DISCUSSION**

Since the correct scale factor is 100\3 and my implementation has given the scale factor as 1.0611, my result are pretty far from the correct solution. I thought that selecting different points might help as in the previous lab and I’ve started to selecting different combinations of these 19 points however this solution didn’t give the proper result. As a consequence of the procedure of 3D structure recovery, we could not get the accurate scale factor of the object.

**Appendix: Lab9.m**

clear all; close all; clc;

%% Definitions

rng(1);

L = 300;

I1 = zeros(L,L);

noiseCoef = 3;

f=L;

u0 = L/2;

v0 = L/2;

K = [f 0 u0;

0 f v0;

0 0 1];

DEG\_TO\_RAD = pi/180;

%% World Coordinates

P\_W=[0 2 0 1;

0 1 0 1;

0 0 0 1;

0 2 -1 1;

0 1 -1 1;

0 0 -1 1;

0 2 -2 1;

0 1 -2 1;

0 0 -2 1;

1 0 0 1;

2 0 0 1;

1 0 -1 1;

2 0 -1 1;

1 0 -2 1;

2 0 -2 1;

1 1 0 1;

2 1 0 1;

1 2 0 1;

2 2 0 1];

P\_W = P\_W';

NPTS = size(P\_W,2); %Number of points

%% Visualization

figure;

subplot(1,3,1)

wally = meshgrid(0:0.1:3);

wallz = meshgrid(-3:0.1:0);

wallx = 0\*ones(size(wallz,1));

surf(wallx, wally, wallz','FaceColor',(1/255)\*[97 178 205],'EdgeColor','none')

hold on

wallx = meshgrid(0:0.1:3);

wallz = meshgrid(-3:0.1:0);

wally = 0\*ones(size(wallz,1));

surf(wallx, wally, wallz','FaceColor',(1/255)\*[77 137 157],'EdgeColor','none')

wallx = meshgrid(0:0.1:3);

wally = meshgrid(0:0.1:3);

wallz = zeros(size(wally,1));

surf(wallx, wally', wallz,'FaceColor',(1/255)\*[45 162 200],'EdgeColor','none')

plot3(P\_W(1,:),P\_W(2,:),P\_W(3,:),'b.','MarkerSize',36);

axis equal;

grid on

axis vis3d;

axis([-3 3 -3 3 -3 3])

xlabel('x')

ylabel('y')

zlabel('z')

plot3(P\_W(1,:),P\_W(2,:),P\_W(3,:),'d');

%title('Original World Points','FontSize',20)

%% Camera Transformation for View 1

ax = 120 \* DEG\_TO\_RAD;

ay = 0 \*DEG\_TO\_RAD;

az = 60 \* DEG\_TO\_RAD;

Rx = [1 0 0;

0 cos(ax) -sin(ax);

0 sin(ax) cos(ax)];

Ry = [cos(ay) 0 sin(ay);

0 1 0;

-sin(ay) 0 cos(ay)];

Rz = [cos(az) -sin(az) 0;

sin(az) cos(az) 0;

0 0 1];

Rc1 = Rx\*Ry\*Rz;

Tc1 = [0;0;5];

M = [Rc1 Tc1];

p1 = K\*(M \* P\_W);

noise1 = noiseCoef\*rand(3,NPTS)-(noiseCoef/2);

noise1(3,:)=1;

p1 = p1 + noise1;

u1(1,:) = p1(1,:) ./ p1(3,:);

u1(2,:) = p1(2,:) ./ p1(3,:);

u1(3,:) = p1(3,:) ./ p1(3,:);

for i=1:length(u1)

x = round(u1(1,i)); y=round(u1(2,i));

I1(y-2:y+2, x-2:x+2) = 255;

end

subplot(1,3,2), imshow(I1, []), title('View 1', 'FontSize',20);

%% Camera Transformation for View 2

ax = 0 \* DEG\_TO\_RAD;

ay = -25 \*DEG\_TO\_RAD;

az = 0 \* DEG\_TO\_RAD;

Rx = [1 0 0;

0 cos(ax) -sin(ax);

0 sin(ax) cos(ax)];

Ry = [cos(ay) 0 sin(ay);

0 1 0;

-sin(ay) 0 cos(ay)];

Rz = [cos(az) -sin(az) 0;

sin(az) cos(az) 0;

0 0 1];

Rc2c1 = Rx\*Ry\*Rz;

Tc2c1 = [3;0;1];

Hc1 = [Rc1 Tc1; 0 0 0 1];

Hc2c1 = [Rc2c1 Tc2c1; 0 0 0 1];

Hc2 = Hc2c1\*Hc1;

Rc2 = Hc2(1:3,1:3);

Tc2 = Hc2(1:3,4);

M = [Rc2 Tc2];

I2 = zeros(L,L);

p2 = K\*(M\*P\_W);

noise2 = noiseCoef\*rand(3,NPTS)-(noiseCoef/2);

noise2(3,:)=1;

p2 = p2 + noise2;

u2(1,:) = p2(1,:) ./ p2(3,:);

u2(2,:) = p2(2,:) ./ p2(3,:);

u2(3,:) = p2(3,:) ./ p2(3,:);

for i=1:length(u2)

x = round(u2(1,i)); y=round(u2(2,i));

I2(y-2:y+2, x-2:x+2) = 255;

end

subplot(1,3,3), imshow(I2, []), title('View 2', 'FontSize',20);

t = Tc2c1;

T\_skew = [0 -t(3) t(2); t(3) 0 -t(1); -t(2) t(1) 0];

Etrue = T\_skew\*Rc2c1;

%% Conversion of pixel points

p1 = inv(K)\*u1;

p2 = inv(K)\*u2;

%% Estimated variables in Lab#8

E = [0.0019 -0.3304 0.0018;

-0.1104 0.0027 -0.9939;

-0.0045 0.9438 0.0040];

R = [0.9016 0.0030 -0.4325;

-0.0019 1.000 0.0029;

0.4326 -0.0018 0.9016];

T = [0.9438; 0.0030; 0.3304];

%% Lab#9 Assignment starts here.

x1=p1;

x2=p2;

M=zeros(57, 20 );

for i = 1:1:19

%% skew

tempx2=x2(1:3,i:i);

skewx2=zeros(3,3);

skewx2(1,1)=0;

skewx2(2,2)=0;

skewx2(3,3)=0;

skewx2(1,2)=-(tempx2(3));

skewx2(2,1)=(tempx2(3));

skewx2(1,3)=tempx2(2);

skewx2(3,1)= -(tempx2(2));

skewx2(2,3)=-(tempx2(1));

skewx2(3,2)=(tempx2(1));

M(i\*3-2:i\*3,i) = skewx2\*R\*x1(:,i);

M(i\*3-2:i\*3,20) = skewx2\*T;

end

[U S V] = svd(M'\*M);

l = V(:,end);

A = zeros(4,19);

for i=1:1:19

A(1,i) = x1(1,i)\*l(i);

A(2,i) = x1(2,i)\*l(i);

A(3,i) = x1(3,i)\*l(i);

A(4,i) = 1;

end

A = inv(Hc1)\*A;

%% Plot the 3D points

figure

subplot(1,2,1)

plot3(P\_W(1,:),P\_W(2,:),P\_W(3,:),'b.','MarkerSize',10)

axis equal, grid on, axis vis3d

xlabel('X')

ylabel('Y')

zlabel('Z')

title('Original World Points')

%Plot your reconstructed world points here.

subplot(1,2,2)

plot3(A(1,:),A(2,:),A(3,:),'r.','MarkerSize',20)

axis equal

grid on

axis vis3d

xlabel('X')

ylabel('Y')

zlabel('Z')

title('Reconstructed World Points (Up to scale)')