1) Load the 100 pairs of corresponding 2-D and 3-D points in the files 2Dpoints.txt and 3Dpoints.txt (the i^{th} row of both files corresponds to the i^{th} point). Use these point correspondences to solve (using Eigen-analysis) for the camera matrix P (whose rasterized vector **p** has a unit $L2$ norm). [5 pts]

Camera matrix P :

P =

-0.0021	-0.0020	-0.0017	0.9123
0.0006	-0.0001	-0.0028	0.4094
-0.0000	0.0000	-0.0000	0.0007

2) Given the computed matrix P (from Problem 1), project the 3-D homogeneous points $(X_i,Y_i,Z_i,1)$ to 2-D. Compute the sum-of-squared error (sum-of-squared distances) between the resulting 3-D-to-2-D projected points and the given 2-D points (ensure all 2-D points are inhomogeneous). [3 pts]

sum-of-squared error = 18.7461

3)The file homography.txt contains 15 corresponding 2-D points from two different images, where the first and second columns correspond to the x and y coordinates of the points in the first image and the third and fourth columns correspond to the x and y coordinates of the points in the second image. Load the 2-D point sets and use the **Normalized Direct Linear Transformation algorithm** to compute the final homography H that maps the original points ***from*** **image 1 to image 2** (i.e., make sure $P_2 = HP_1$). [5 pts]

Final homography H:

$$H = \begin{matrix} 0.3875 & 0.4842 & -21.3951 \\ -0.0609 & 0.2709 & 90.4191 \\ 0.0003 & 0.0003 & 0.4072 \end{matrix}$$

4)Plot the points from image 2 and the projected points from image 1 on the same plot. Make sure the projected points are converted into inhomogeneous form. [1 pt]

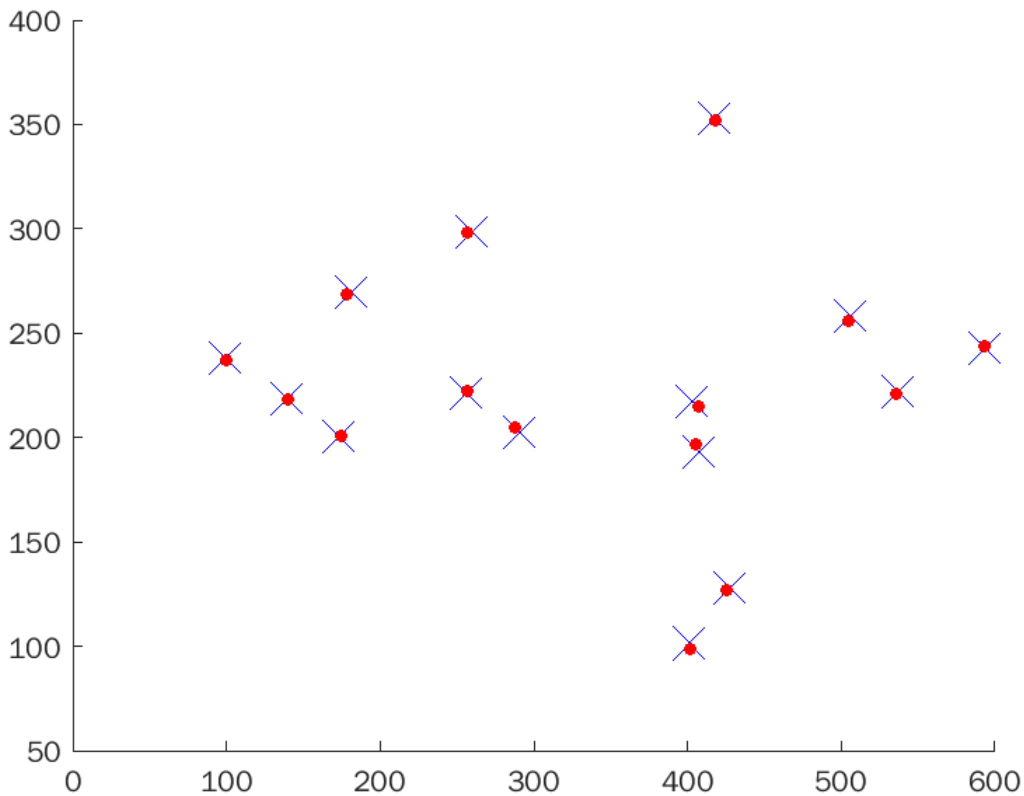

Image 1: Blue X

Image 2: Red dots

As we can see, points from image2 are very close to the corresponding projection points, which means our result is good.

5)Compute the sum-of-squared error (squared Euclidean distance) between the actual points from image 2 and the projected points from image 1. [2 pts]

sum-of-squared error = 105.9739

6)As usual, upload the standard material.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Problem 1

D2 = readtable('2Dpoints.txt');
D3 = readtable('3Dpoints.txt');

x = table2array(D2(:,1));
y = table2array(D2(:,2));

Xw = table2array(D3(:,1));
Yw = table2array(D3(:,2));
Zw = table2array(D3(:,3));

length = size(x);
z = 1;
for i = 1:length(1)
    for j = 1:2
        if mod(j, 2) ~= 0
            A(z,1:12) = [Xw(i), Yw(i), Zw(i), 1, 0, 0, 0, 0, -1*Xw(i)*x(i),
-1*Yw(i)*x(i), -1*Zw(i)*x(i), -1*x(i)];
        else
            A(z,1:12) = [0,0,0,0, Xw(i), Yw(i), Zw(i), 1, -1*Xw(i)*y(i),
-1*Yw(i)*y(i), -1*Zw(i)*y(i), -1*y(i)];
        end
        z = z+1;
    end
end

B = A' * A;

[eVec, eVal] = eig(B);
m = min(eVal(eVal ~= 0));
[minRow,minColumn] = find(eVal == m);
pPrev = eVec(1:12,minColumn);
p = reshape(pPrev,[4,3])';
disp("P = ");
disp(p);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Problem 2
vector1(1:length(1)) = 1;
d3h = [Xw'; Yw';Zw';vector1];
D2h = p * d3h;
D2 = table2array(D2);
for i = 1:length(1)
```

```
    D2P(i,1) = D2h(1,i)/D2h(3,i);
    D2P(i,2) = D2h(2,i)/D2h(3,i);
    distance(i) = (D2P(i,1) - D2(i,1)) ^ 2 + (D2P(i,2) - D2(i,2)) ^2;

end
sumOfSquared = sum(distance, 'all');
disp("sum-of-squared error = ");
disp(sumOfSquared);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Problem 3
clear; close all;
points = readtable('homography.txt');

x1 = table2array(points(:,1));
y1 = table2array(points(:,2));

x2 = table2array(points(:,3));
y2 = table2array(points(:,4));

x1_mean = mean(x1);
y1_mean = mean(y1);

x2_mean = mean(x2);
y2_mean = mean(y2);

length = size(x1);
for i = 1:length(1)

    distance1(i) = sqrt((x1(i) - x1_mean) ^ 2 + (y1(i) - y1_mean) ^2);
    distance2(i) = sqrt((x2(i) - x2_mean) ^ 2 + (y2(i) - y2_mean) ^2);

end

s1 = sqrt(2) / mean(distance1);
s2 = sqrt(2) / mean(distance2);

z = 1;
for i = 1:length(1)

    for j = 1:2
        if mod(j, 2) ~= 0
            A(z,1:9) = [s1*(x1(i)-x1_mean), s1*(y1(i)-y1_mean), 1, 0, 0, 0,
-1*s1*(x1(i)-x1_mean)*s2*(x2(i)-x2_mean), -1*s1*(y1(i)-y1_mean)*s2*(x2(i)-
x2_mean), -1*s2*(x2(i)-x2_mean)];
        else
            A(z,1:9) = [0, 0, 0, s1*(x1(i)-x1_mean), s1*(y1(i)-y1_mean), 1,
-1*s1*(x1(i)-x1_mean)*s2*(y2(i)-y2_mean), -1*s1*(y1(i)-y1_mean)*s2*(y2(i)-
y2_mean), -1*s2*(y2(i)-y2_mean)];
        end
        z = z+1;
    end
end

T1 = [s1, 0, -1*s1*x1_mean;
      0 , s1, -1*s1*y1_mean;
      0, 0, 1];

T2 = [s2, 0, -1*s2*x2_mean;
```

```
      0 , s2, -1*s2*y2_mean;
      0, 0, 1];

B = A' * A;

[eVec, eVal] = eig(B);
m = min(eVal(eVal ~= 0));
[minRow,minColumn] = find(eVal == m);
hPrev = eVec(1:9,minColumn);
h_hat = reshape(hPrev,[3,3])';
H = inv(T2) * h_hat * T1;
disp("H = ");
disp(H);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Problem 4
vector1(1:length(1)) = 1;

d2h = [x1'; y1';vector1];
D2h = H * d2h;

for i = 1:length(1)
    x2_p(i) = D2h(1,i)/D2h(3,i);
    y2_p(i) = D2h(2,i)/D2h(3,i);

    distance(i) = (x2_p(i) - x2(i)) ^ 2 + (y2_p(i) - y2(i)) ^2;

end

figure;
hold on;
plot(x2_p,y2_p,'bx','MarkerSize',15);
plot(x2,y2,'r.','MarkerSize',15);
hold off;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Problem 5
sumOfSquared = sum(distance, 'all');
disp("sum-of-squared error = ");
disp(sumOfSquared);
```