Yukun Duan
Computer Vision for HCI AU'22
Homework Assignment #9

1.

Here is the camera matrix I solved:

```
  -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.

The calculated SSE is: 18.7461.

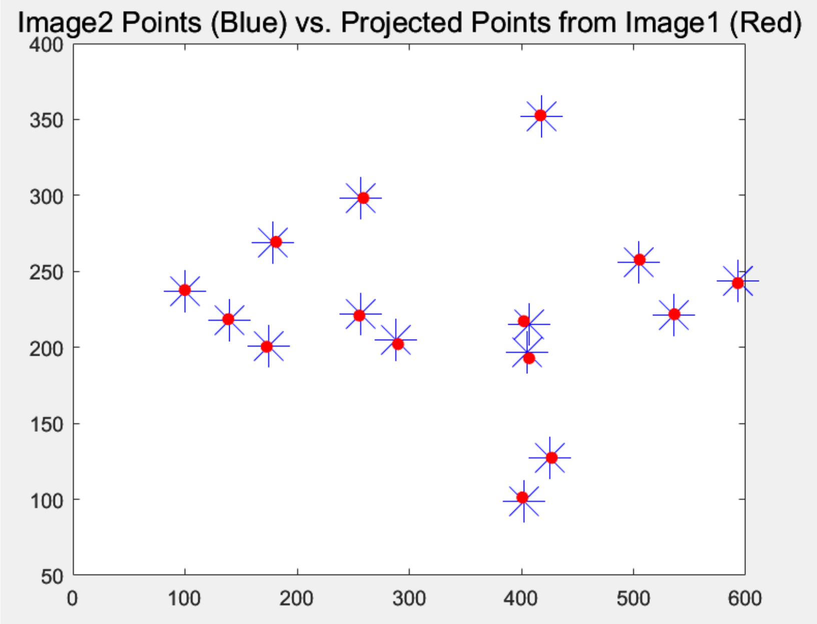
3.

The calculated H value is:

```
   0.3875     0.4842   -21.3951
  -0.0609     0.2709    90.4191
   0.0003     0.0003     0.4072
```

4.

Here is the projected point from image1 and the origin image2. According to the image below, the points are mostly lying on origin points from image2.



5.

The calculated SSE is 105.9739.

Attached Code:

```
% Yukun Duan
% CSE5524 - HW9
% 10/29/2022


%% Problem 1
% load 2d and 3d data
data2d = load('2Dpoints.txt');
data3d = load('3Dpoints.txt');
x = data2d(:,1); y = data2d(:,2);
X = data3d(:,1); Y = data3d(:,2); Z = data3d(:,3);
one = ones(size(data2d,1), 1); zero = zeros(size(data2d, 1), 1);

% construct matrix A
A = zeros(2*size(data2d,1), 12);
A(1:2:end, :) = [X,Y,Z,one,zero,zero,zero,zero,-X.*x, -Y.*x, -Z.*x, -x];
A(2:2:end, :) = [zero,zero,zero,zero,X,Y,Z,one,-X.*y, -Y.*y, -Z.*y, -y];

% calculate p value
[V, D] = eig(A'*A);
p = V(:, 1); % eigenvector with smallest egvalue
p = reshape(p, 4, [])';
disp(p)

%% Problem 2
% calculate homogeneous 3d array and 2d array
origin3dArr = [data3d, one]';
result2dArr = p * origin3dArr;

% switch result to inhomogeneous and reshape to same size as input 2d array
result2d = [result2dArr(1,:)./result2dArr(3,:);
result2dArr(2,:)./result2dArr(3,:)];
result2d = reshape(result2d, [], size(data2d,1))';

% calculate sum of squared error
error = sum((result2d - data2d).^2, 'all');
disp(error)

%% Problem 3
% load data for im1x, im1y, im2x, im2y
data = load('homography.txt');
im1x = data(:,1); im1y = data(:,2); im2x = data(:,3); im2y = data(:,4);
one = ones(size(data,1), 1); zero = zeros(size(data, 1), 1);
```

```
% calculate Ta and Tb
sa = sqrt(2)/(mean(sqrt((im1x-mean(im1x)).^2 + (im1y-
mean(im1y)).^2),'all'));
sb = sqrt(2)/(mean(sqrt((im2x-mean(im2x)).^2 + (im2y-
mean(im2y)).^2),'all'));
Ta = [sa,0,-sa*mean(im1x); 0,sa,-sa*mean(im1y); 0,0,1];
Tb = [sb,0,-sb*mean(im2x); 0,sb,-sb*mean(im2y); 0,0,1];

% get origin data transformed using calculated s
im1x = sa*(im1x-mean(im1x));
im1y = sa*(im1y-mean(im1y));

im2x = sb*(im2x - mean(im2x));
im2y = sb*(im2y - mean(im2y));

% construct A
A = zeros(2*size(im1x, 1), 9);
A(1:2:end, :) = [im1x, im1y, one, zero, zero, zero, -im1x.*im2x, -
im1y.*im2x, -im2x];
A(2:2:end, :) = [zero, zero, zero, im1x, im1y, one, -im1x.*im2y, -
im1y.*im2y, -im2y];

% calculate homography h
[V, D] = eig(A'*A);
h = V(:, 1); % eigenvector with smallest egvalue
h = reshape(h, 3, [])';

% calculate H
H = inv(Tb)* h * Ta;
disp(H)



%% Problem 4 and 5
% construct projected result
origin2dArr = [data(:,1), data(:,2), one]';
result2dArr = H * origin2dArr;
result2d = [result2dArr(1,:)./result2dArr(3,:);
result2dArr(2,:)./result2dArr(3,:)];
result2d = reshape(result2d, [], size(data,1))';

% plot projected from image1 and origin value of image2
```

```
plot(data(:,3), data(:,4), 'b*', result2d(:,1), result2d(:,2), 'r.',
'MarkerSize', 20)
title('Image2 Points (Blue) vs. Projected Points from Image1
(Red)','FontSize', 14)

% calculate error
error = sum((result2d - data(:, 3:4)).^2, 'all');
disp(error)
```