

Name: __Tristan Roman (.198)__

Score: /16

CSE 5524

Computer Vision for HCI

SP'22

Homework Assignment #9

Due: Thursday 3/31

```
% Tristan Roman
% CSE 5524 - HW9
% 3/31/2022
```

%% Problem 1 - Camera Calibration

```
P =
    0.0024    0.0024    0.0019   -0.9331
   -0.0007    0.0002    0.0032   -0.3596
    0.0000   -0.0000    0.0000   -0.0004
```

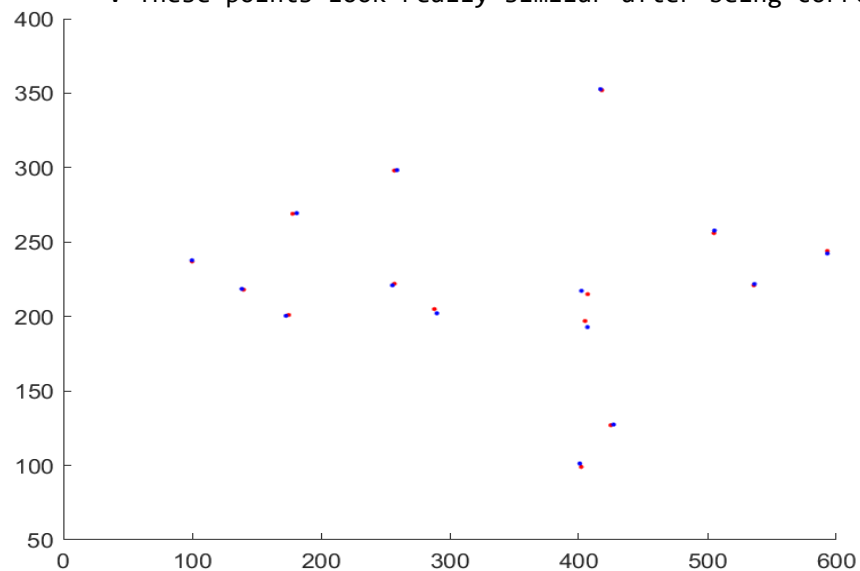
^ These seem appropriate based on the equations observed on the slides.

V The SSE seems not terribly high when considering the goal is to reduce the error of the given points.

SSE = 47.2487

%% Problem 2 - Homography

V These points look really similar after being corrected - success!



SSE = 105.9739

^ This seems high after seeing the graph, but possibly this is because there are so many points present?

Code displayed below:

```

%% Problem 1 - Camera Calibration
% load data
D2 = readmatrix('2Dpoints.txt');
D3 = readmatrix('3Dpoints.txt');

% construct matrix A
A = zeros(2,12);
N = 8;
for i = 1:N
    A = add2rows(A,i,D2,D3);
end

A = A(3:2*N+2,:);

B = transpose(A)*A;

[V,D] = eig(B);
p = V(:,1);
p1 = p(1:4);
p2 = p(5:8);
p3 = p(9:12);

P = [p1';p2';p3'];
D3 = transpose(D3);
D3 = [D3;ones(1,100)];
D2test = P*D3;
% compute Sum of squared errors
D2Result = [D2test(1,:)./D2test(3,:);D2test(2,:)./D2test(3,:)];
D2 = D2';
SSE = 0;
for i = 1:2
    for j = 1:100
        SSE = SSE + (D2(i,j)-D2Result(i,j))^2;
    end
end
SSE
pause();
%% Problem 2 - Homography
clear;
clc;
data = readmatrix('homography.txt');
A = data(:,1:2);
B = data(:,3:4);

sa = computeS(A);
sb = computeS(B);

Ta = computeT(A,sa);
Tb = computeT(B,sb);

% transform image A and B to Pa and Pb with column for each point
A = [A';ones(1,15)];
Pa = Ta*A;
B = [B';ones(1,15)];

```

```

Pb = Tb*B;

% solve for h
AA = zeros(2,9);
for i = 1:15
    AA = add2row(AA,i,Pa,Pb);
end
AA = AA(3:32,:);
[v,d] = eig(AA'*AA);
h = v(:,1);

h1 = h(1:3);
h2 = h(4:6);
h3 = h(7:9);
h = [h1';h2';h3'];

H = inv(Tb)*h*Ta; % answer for part 1

%% plot
temp = H*A;
result = temp(1:2,:)./temp(3,:);
figure;
hold on
X = data(:,3);
Y = data(:,4);
plot(X,Y,'r. ');
result = result';
plot(result(:,1),result(:,2),'b. ');
hold off
%% compute sum of square error
sse = 0;
B = data(:,3:4);
for i = 1:15
    for j = 1:2
        sse = sse + (result(i,j) - B(i,j))^2;
    end
end

%%
function result = add2row(AA,i,Pa,Pb)
    result = zeros(2,9);
    result(1,1) = Pa(1,i);
    result(1,2) = Pa(2,i);
    result(1,3) = 1;
    result(1,7) = -Pa(1,i)*Pb(1,i);
    result(1,8) = -Pa(2,i)*Pb(1,i);
    result(1,9) = -Pb(1,i);
    result(2,4) = Pa(1,i);
    result(2,5) = Pa(2,i);
    result(2,6) = 1;
    result(2,7) = -Pa(1,i)*Pb(2,i);
    result(2,8) = -Pa(2,i)*Pb(2,i);
    result(2,9) = -Pb(2,i);

    result = [AA;result];

```

end

```
function result = computeT(M,s)
    result = zeros(3,3);
    meanM = mean(M);
    result(1,1) = s;
    result(1,3) = -s*meanM(1);
    result(2,2) = s;
    result(2,3) = -s*meanM(2);
    result(3,3) = 1;
end
```

```
function result = computeS(M)
    meanM = mean(M);
    result = 0;
    for i = 1:15
        result = result + sqrt((M(i,1)-meanM(1))^2+(M(i,2)-meanM(2))^2);
    end
    result = sqrt(2)/(result/15);
end
```

```
function result = add2rows(A,i,D2,D3)
    % first new row
    newRows = zeros(2,12);
    newRows(1,1) = D3(i,1);
    newRows(1,2) = D3(i,2);
    newRows(1,3) = D3(i,3);
    newRows(1,4) = 1;
    newRows(1,9) = -D3(i,1)*D2(i,1);
    newRows(1,10) = -D3(i,2)*D2(i,1);
    newRows(1,11) = -D3(i,3)*D2(i,1);
    newRows(1,12) = -D2(i,1);

    % second new row
    newRows(2,5) = D3(i,1);
    newRows(2,6) = D3(i,2);
    newRows(2,7) = D3(i,3);
    newRows(2,8) = 1;
    newRows(2,9) = -D3(i,1)*D2(i,2);
    newRows(2,10) = -D3(i,2)*D2(i,2);
    newRows(2,11) = -D3(i,3)*D2(i,2);
    newRows(2,12) = -D2(i,2);

    % stack the matrices
    result = [A;newRows];
end
```