# 3. Exercise in Estimating View Geometry

## 1. Camera Model

### 1.1. Setting up the Camera

First, we create the rotation for every camera, which is the identity matrix. Identity matrix should be of size *n x n* with ones on the main diagonal and zeros elsewhere.

* I = eye(3); % Identity Matrix

After that, we create the translations for the 3 cameras:

* t1 = [0 0 0]’;
* t2 = [-5 0 2]’;
* t3 = [0.1 0 0.1];

Now that we have the translation and rotation for every camera, we implement the camera matrices:

* P1 = [I t1];
* P2 = [I t2];
* P3 = [I t3];

Camera matrices are *3 x 4* matrices

We have the 3D point, Q = [2 4 10 1]’; and we project it to all three cameras:

* Q = [2 4 10 1]’;
* q1 = P1 \* Q1;
* q2 = P2 \* Q1;
* q3 = P3 \* Q1;

q1, q2, q3 are *3 x 1* vectors, which represent the coordinates of each point on image in homogeneous coordinates.

We normalize the points:

q1(1,:)=q1(1,:)./q1(3,:);

q1(2,:)=q1(2,:)./q1(3,:);

q2(1,:)=q2(1,:)./q2(3,:);

q2(2,:)=q2(2,:)./q2(3,:);

q3(1,:)=q3(1,:)./q3(3,:);

q3(2,:)=q3(2,:)./q3(3,:);

Give a verbal/intuitive explanation of the 3 cameras location and viewing direction.

**Verification of Q1:**

q11 = [0.2 0.3 1]’;

q12 = [-0.25 1/3 1]’;

q13 = [0.2079 0.3960 1]’;

[Q1, B] = Est3D(q11, P1, q12, P2);

The Q1 output from this function is:

|  |
| --- |
| 0.1818 |
| 0.3636 |
| 0.9091 |
| 0.0909 |

If we compare with the 3D point Q1 = [2 4 10 1]’; we will see that the first is a scaled version of the other.

### 1.2. 3D Solver

1. We have the projection points,

* q21 = [-0.1667, 0.3333, 1]’;
* q22 = [-0.50000, 0.2857, 1]’;

in cameras 1 and 2 respectively. We use the function Est3D to estimate the 3D point Q2.

q21 = [-0.1667 0.3333 1]';

q22 = [-0.5 0.2857 1]';

[Q2,B]=Est3D(q21,P1,q22,P2);

Q2 output:

|  |
| --- |
| -0.1557 |
| 0.3114 |
| 0.9342 |
| 0.0778 |

1. Calculate q23, the projection of Q2 into camera 3.

* q23 = P3 \* Q2;

q23 output:

|  |
| --- |
| -0.1479 |
| 0.3114 |
| 0.9420 |

### 1.3. Accuracy

1. Adding 0.1 to all coordinates of q22 and q23

* q22n = [q22(1:2) + 0.1; q22(3)];
* q23n = [q23(1:2) + 0.1; q23(3)];

1. Re-estimating the 3D position of Q2 based on q21 and q22n, using cameras 1 and 2 (P1 and P2)

* [Q2, B] = Est3D(q21, P1, q22n, P2);

Q2 output:

|  |
| --- |
| -0.1618 |
| 0.3502 |
| 0.9214 |
| 0.0474 |

1. Re-estimating the 3D position of Q2 based on q21 and q23n

* [Q2, B] = Est3D(q21, P1, q23n, P2)

Q2 output:

|  |
| --- |
| -0.1593 |
| 0.3358 |
| 0.9281 |
| -0.0243 |

1. Compare the estimates from 2 and 3 with the ‘ground truth’ based on q21 and q22. Explain!

### 1.4 Real Images

load TwoImageData.mat

figure('name','Q12 view 1');

imshow(im1);

g1 = [ginput(1) 1]';

figure('name','Q12 view 2');

imshow(im2);

g2 = [ginput(1) 1]';

format long e

[Q3,B]=Est3D(g1,P1,g2,P2);

## 2. 3D Inference from a Plane

|  |
| --- |
| %% 3D Inference from a Plane  clear; close all;  im = imread('petergade.png');  imagesc(im);    q = ginput(4);  q = [q, ones(4,1)];  qp = [0, 0 1; 610, 0, 1; 0, 1340 1; 610, 1340 1];    %%    X1 = q';    % Normalizing q  Mean1=mean(X1')';  X1(1,:)=X1(1,:)-Mean1(1);  X1(2,:)=X1(2,:)-Mean1(2);  S1=mean(sqrt(diag(X1'\*X1)))/sqrt(2);  X1(1:2,:)=X1(1:2,:)/S1;  T1=[eye(2)/S1,-Mean1(1:2)/S1;0 0 1];    % Normalizing qp  X1p = qp';  Mean1=mean(X1p')';  X1p(1,:)=X1p(1,:)-Mean1(1);  X1p(2,:)=X1p(2,:)-Mean1(2);  S1p=mean(sqrt(diag(X1p'\*X1p)))/sqrt(2);  X1p(1:2,:)=X1p(1:2,:)/S1p;  T1p=[eye(2)/S1p,-Mean1(1:2)/S1p;0 0 1];    % Finding H  b1 = kron(X1(:,1)', CrossOp(X1p(:,1)));  b2 = kron(X1(:,2)', CrossOp(X1p(:,2)));  b3 = kron(X1(:,3)', CrossOp(X1p(:,3)));  b4 = kron(X1(:,4)', CrossOp(X1p(:,4)));  B = [b1;b2;b3;b4];    [~,~,H] = svd(B);  H = H(:,end);  H = reshape(H,3,3);  H = inv(T1p)\*H\*T1;    %% Warping image  figure(2)  Tr = maketform('projective',H');  WarpIm = imtransform(im,Tr,'Xdata',[-100, 710], 'Ydata',[-200, 1440]);  imagesc(WarpIm)    %% Estimating players position  close all;  imagesc(im);  q = ginput(2);  q = [q, ones(2,1)];  qp = H\*q';  qp(:,1) = qp(:,1)./qp(3,1);  qp(:,2) = qp(:,2)./qp(3,2); |

What is the position of the two players on the court?