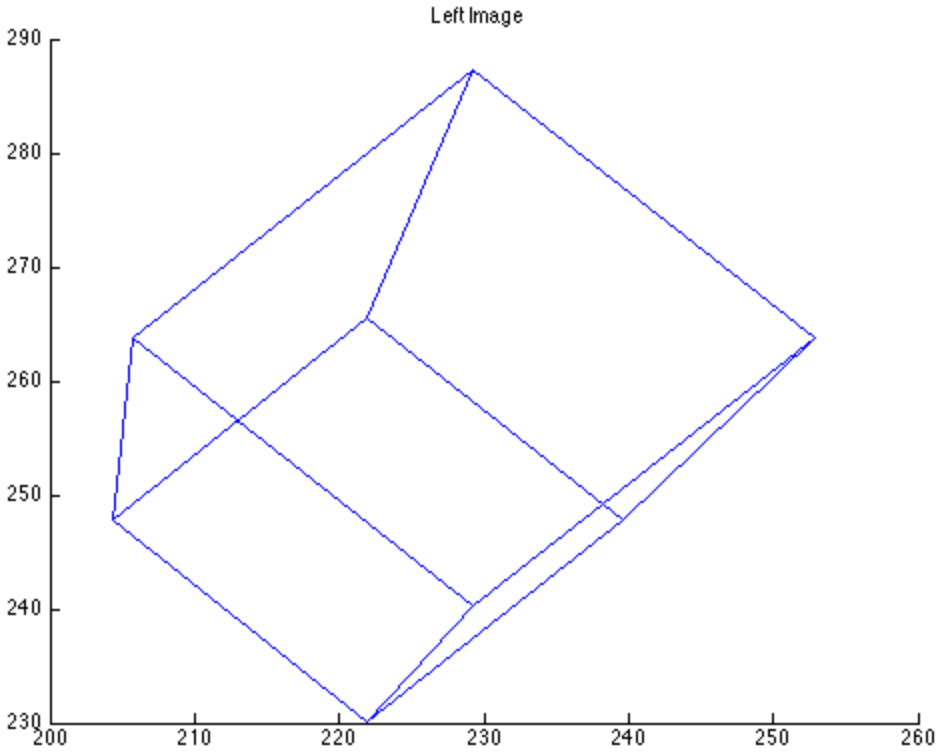
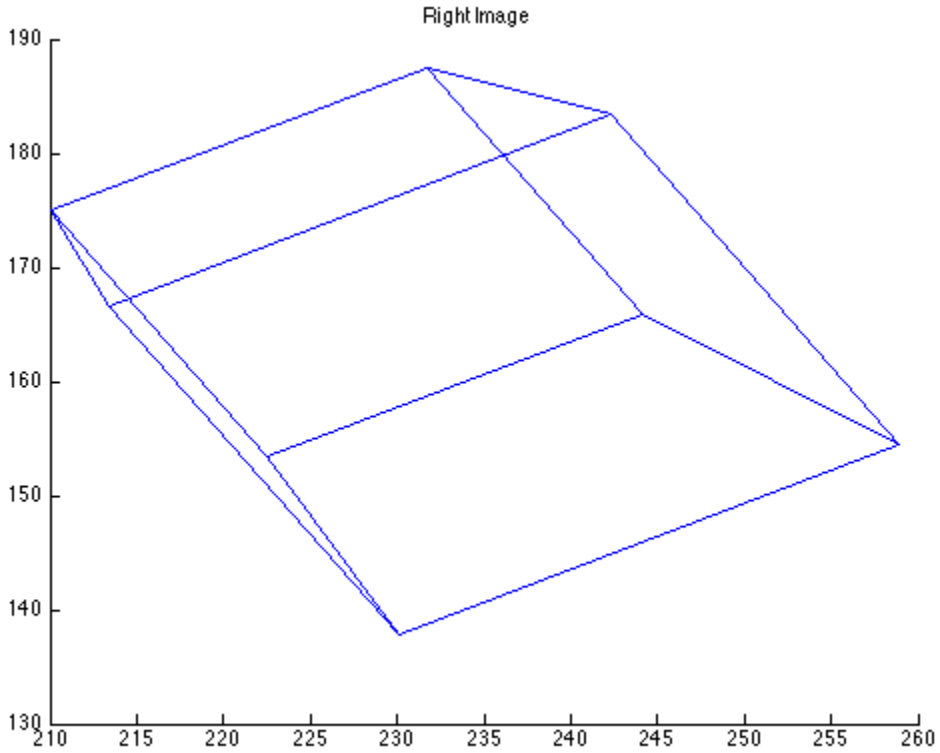
```
% intrinsic param
K = [-100 0 200 ;
    0 -100 200 ;
    0 0 1];

Mextleft = [ 0.707 0.707 0 -3 ;-0.707 0.707 0 -0.5; 0 0  1 3];
Mextright = [ 0.866 -0.5 0 -3 ;0.5 0.866 0 -0.5; 0 0 1 3];

pts = [ 2 0 0 ;
  3 0 0;
  3 1 0;
  2 1 0;
  2 0 1 ;
  3 0 1;
  3 1 1;
  2 1 1;
  2.5 0.5 2];
%for i = 1:9,
 %   pts(i,:) = rand(1,3);
%end;

%pts = [ 0 0 0 ; 0 1 0; 1 1 0; -1 1 0;1 0 1 ;1 0 1;1 1 1;1 1 1;1.5 0.5 2];

NN = 9;
pix = zeros(NN,3);
for i = 1:NN,
    pixels = K*Mextleft * [pts(i,1) pts(i,2) pts(i,3) 1]';
    leftpix(i,:) = pixels./pixels(3);
    pixels = K*Mextright * [pts(i,1) pts(i,2) pts(i,3) 1]';
    rightpix(i,:) = pixels./pixels(3);
end

% rightpix and leftpix are the list of corresponding points (attainable by
% ginput also

figure(1);clf;
drawmyobject(leftpix); title('Left Image');
figure(2);clf;
drawmyobject(rightpix); title('Right Image');
```


Left Image
290
280
270
260
250
240
230
200
210
220
230
240
250
260


Right Image
190
180
170
160
150
140
130
210
215
220
225
230
235
240
245
250
255
260

From pixels to rays

```
rightray = inv(K)*[rightpix(:,1) rightpix(:,2) rightpix(:,3)]';
leftray = inv(K)*[leftpix(:,1) leftpix(:,2) leftpix(:,3)]';
```

STEREO RECONSTRUCTION With known camera matrices

```
Trw = [Mextright ; 0 0 0 1];
Tlw = [Mextleft; 0 0 0 1];
Twr = inv(Trw); % can be done using transpose
Twl = inv(Tlw); % can be done using transpose

Tlr = Tlw*Twr;
% Rotation from right to left coordinate frame
Rlr = Tlr(1:3,1:3);
    % translation
tlr = Tlr(1:3,4);

reconpts = reconstruct3d(leftray,rightray,Rlr,tlr,Twl);

figure(3);clf;view(3)
drawmy3dobject(pts(:,1:3));title('Original 3D Points');

figure(4);clf;view(3)
drawmy3dobject(reconpts(:,1:3));title('Euclidian Reconstruction');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Now assume that no parameters are known and only point correspondences
% are given
% Reconstruct F from point correspondences.
%
% Using the epipolar constraint between the point correspondences, F can be
% found


for i = 1:NN
tt=leftpix(i,:)' * rightpix(i,:); % 3 x3 matrix
%form the matrix for Aq = 0, where q is 9x1 the elements of
A(i,:) = [tt(1,:) tt(2,:) tt(3,:)];
end;

[U,S,V] = svd(A);
lastcol = V(:,9);
F(1,1) = lastcol(1); F(1,2) = lastcol(2); F(1,3) = lastcol(3);
F(2,1) = lastcol(4); F(2,2) = lastcol(5); F(2,3) = lastcol(6);
F(3,1) = lastcol(7); F(3,2) = lastcol(8); F(3,3) = lastcol(9);
```

```
% Compare this to the built in call
% F = estimateFundamentalMatrix(righpix(:,1:2),leftpix(:,1:2),'Method','Norm8Point



% Get Camera Matrix From F
Ktemp = eye(3);
[U,S,V] = svd(F');
lastcol = V(:,3);
epipole = lastcol;
e1 = epipole(1);
e2 = epipole(2);
e3 = epipole(3);
ecross = [0 -e3 e2; e3 0 -e1; -e2 e1 0];
% camera matrix from F
cameramatrix = [ecross*F epipole];


% Reconstruct with triangulation
Rlr = cameramatrix(1:3,1:3);
tlr = cameramatrix(1:3,4);

reconpts = reconstruct3d(leftray,rightray,Rlr,tlr,Twl)
% Result is up to a projective transformation
figure(5);clf;view(3);
drawmy3dobject(reconpts(:,1:3));title('Reconstruction up to a Projective Transform
figure(6);clf;view(2);
drawmy3dobject(reconpts(:,1:3));title('Different View, Reconstruction up to a Proj


%%%%%%%%% Find Essential Matrix
% Now assume you know K and find E.
E = K'*F*K;

% Get camera matrix from E , use standard method
W = [ 0 -1 0; 1 0 0 ; 0 0 1];
Z = [0 1 0; -1 0 0 ; 0 0 0];
[U,S,V] = svd(E);
% four cases S1,R1 S2,R1  S1,R2  S2,R2
S1 = -U*Z*U';
S2 = U*Z*U';
R1 = U*W'*V';
R2 = U*W*V';

foundit = 0;

%case 1
if (~foundit)
    S = S1;R=R1;
    tlr= [ S(3,2) S(1,3) -S(1,2)]';
    reconpts = reconstruct3d(leftray,rightray,R,tlr,eye(4))
 if (min(reconpts(:,3))>0) foundit = 1;
    end
end
```

```
%case 2
if (~foundit)
    S = S2;R=R1;
    tlr= [ S(3,2) S(1,3) -S(1,2)]';
    reconpts = reconstruct3d(leftray,rightray,R,tlr,eye(4))
    if (min(reconpts(:,3))>0) foundit = 1;
    end
end


%case 3
if (~foundit)
    S = S1;R=R2;
    tlr= [ S(3,2) S(1,3) -S(1,2)]';
    reconpts = reconstruct3d(leftray,rightray,R,tlr,eye(4))
    if (min(reconpts(:,3))>0) foundit = 1;
    end
end


%case 4
if (~foundit)
    S = S2;R=R2;
    tlr= [ S(3,2) S(1,3) -S(1,2)]';
    reconpts = reconstruct3d(leftray,rightray,R,tlr,eye(4))
   if (min(reconpts(:,3))>0) foundit = 1;
    end
end


figure(7);clf;view(3);
drawmy3dobject(reconpts(:,1:3));title('Reconstruction knowing only K');
```

```
NN =

     9


q =

   -0.8706
    0.4637
   -0.1644


q =

   -0.8726
    0.4648
    0.1504
```

```
q =

   -0.8564
    0.4562
    0.2419


q =

   -0.8805
    0.4690
   -0.0693


q =

   -0.8758
    0.4665
   -0.1240


q =

   -0.8769
    0.4671
    0.1134


q =

   -0.8676
    0.4621
    0.1838


q =

   -0.8814
    0.4695
   -0.0520


q =

   -0.8823
    0.4700
    0.0248


NN =

     9
```

```
q =

   -0.3677
   -0.6836
   -0.6305


q =

   -0.3510
   -0.6518
   -0.6723


q =

   -0.4073
   -0.7592
   -0.5077


q =

   -0.4236
   -0.7906
   -0.4422


q =

   -0.4038
   -0.7526
   -0.5202


q =

   -0.3910
   -0.7281
   -0.5630


q =

   -0.4318
   -0.8063
   -0.4044


q =

   -0.4425
   -0.8270
   -0.3468
```

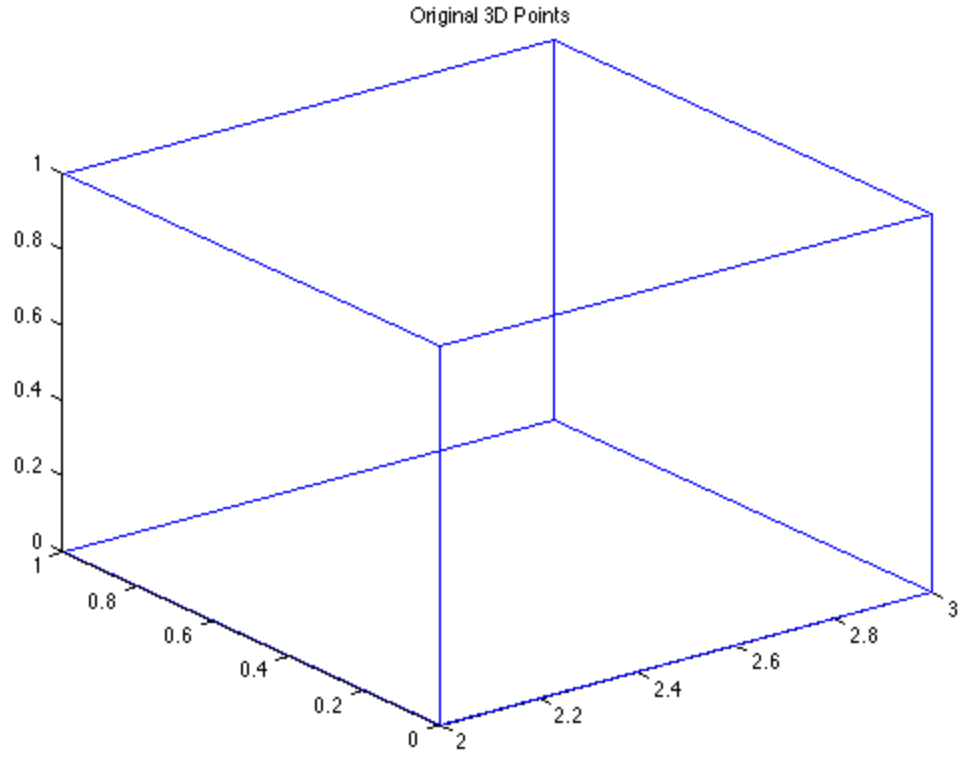
```
q =

   -0.4351
   -0.8127
   -0.3876


reconpts =

    1.6435    3.1679   -3.2451    1.0000
    1.4851    3.1481   -3.2485    1.0000
    1.4814    3.0454   -3.2182    1.0000
    1.6210    3.0554   -3.1985    1.0000
    1.6369    3.0996   -3.2224    1.0000
    1.5263    3.0901   -3.2324    1.0000
    1.5331    3.0119   -3.1846    1.0000
    1.6216    3.0163   -3.1627    1.0000
    1.5926    3.0231   -3.1786    1.0000


NN =

     9


q =

    0.8706
   -0.4637
    0.1644


q =

    0.8725
   -0.4648
   -0.1504


q =

   -0.8567
    0.4557
    0.2416


q =

   -0.8805
    0.4690
   -0.0693
```

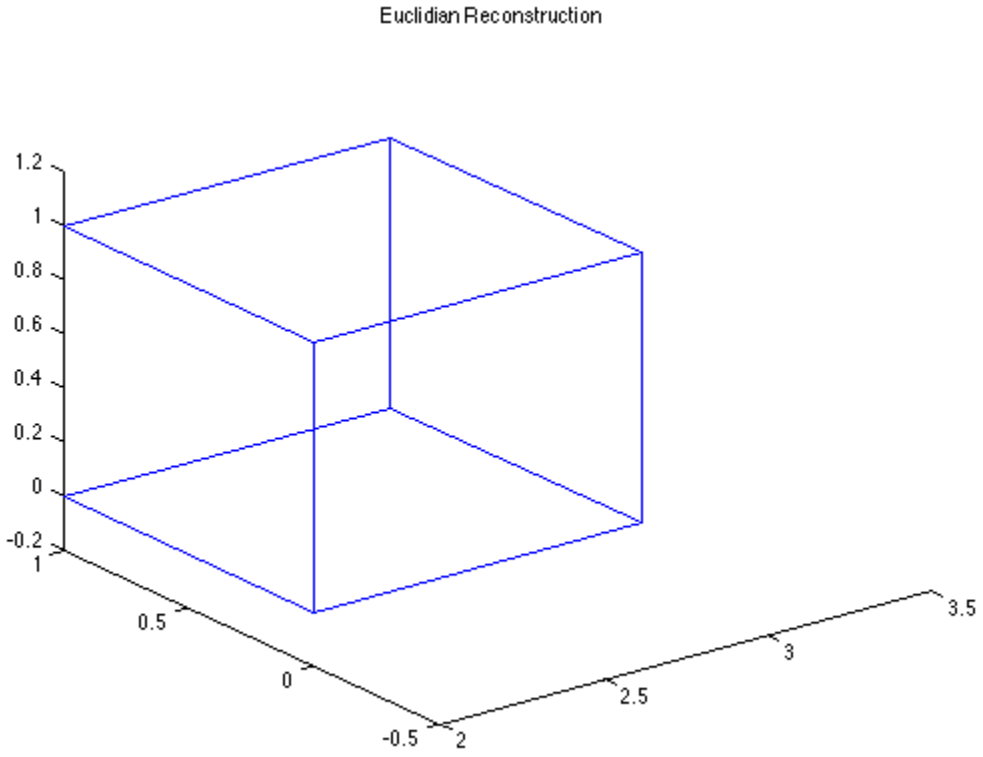
```
q =

    0.8758
   -0.4665
    0.1241


q =

    0.8769
   -0.4671
   -0.1134


q =

   -0.8679
    0.4616
    0.1836


q =

   -0.8814
    0.4695
   -0.0520


q =

    0.8823
   -0.4700
   -0.0248


reconpts =

   -1.3591   -1.6402    2.5708    1.0000
   -0.5022   -1.4976    1.7141    1.0000
    1.0571   11.7660  -18.4417    1.0000
    1.1784    1.6181   -4.0217    1.0000
   -1.3591   -1.6402    3.4278    1.0000
   -0.5022   -1.4976    2.2855    1.0000
    1.0570   11.7655  -24.5880    1.0000
    1.1784    1.6181   -5.3623    1.0000
   -1.7506   -3.8118    9.9578    1.0000


NN =

     9
```

```
q =

    0.8706
   -0.4637
    0.1644


q =

    0.8725
   -0.4648
   -0.1504


q =

   -0.8567
    0.4557
    0.2416


q =

   -0.8805
    0.4690
   -0.0693


q =

    0.8758
   -0.4665
    0.1241


q =

    0.8769
   -0.4671
   -0.1134


q =

   -0.8679
    0.4616
    0.1836


q =

   -0.8814
    0.4695
   -0.0520
```

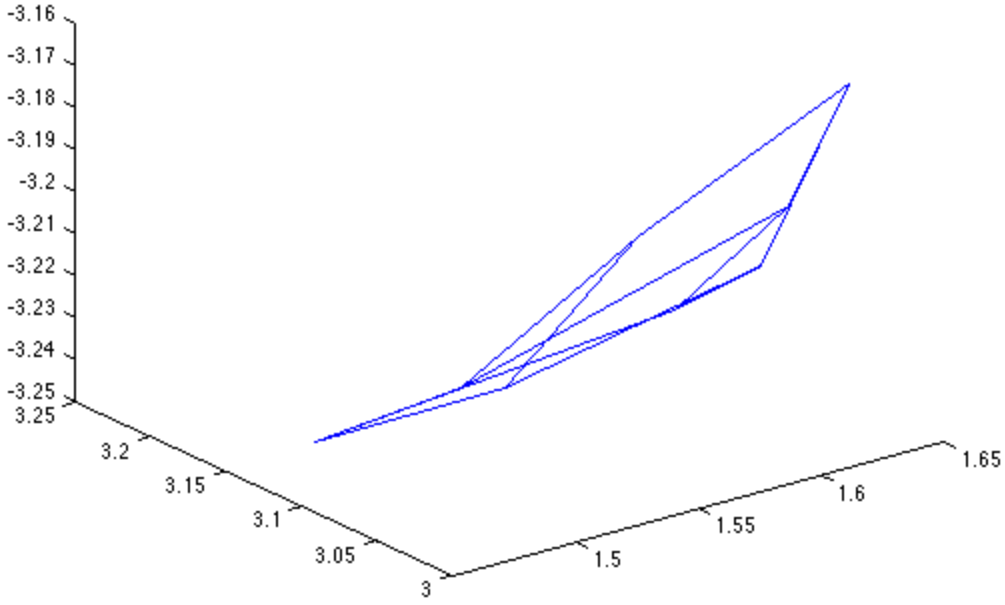
```
q =

    0.8823
   -0.4700
   -0.0248


reconpts =

    1.3591    1.6402   -2.5708    1.0000
    0.5022    1.4976   -1.7141    1.0000
   -1.0571  -11.7660   18.4417    1.0000
   -1.1784   -1.6181    4.0217    1.0000
    1.3591    1.6402   -3.4278    1.0000
    0.5022    1.4976   -2.2855    1.0000
   -1.0570  -11.7655   24.5880    1.0000
   -1.1784   -1.6181    5.3623    1.0000
    1.7506    3.8118   -9.9578    1.0000


NN =

     9


q =

   -0.8706
    0.4637
   -0.1644


q =

   -0.8725
    0.4648
    0.1504


q =

   -0.8564
    0.4562
    0.2419


q =

   -0.8805
    0.4690
   -0.0693
```

```
q =

   -0.8758
    0.4665
   -0.1240


q =

   -0.8769
    0.4671
    0.1134


q =

   -0.8675
    0.4621
    0.1838


q =

   -0.8814
    0.4695
   -0.0520


q =

   -0.8823
    0.4700
    0.0248


reconpts =

   -0.4283   -0.5169    0.8102    1.0000
   -0.2374   -0.7078    0.8102    1.0000
   -0.0464   -0.5169    0.8102    1.0000
   -0.2374   -0.3259    0.8101    1.0000
   -0.4283   -0.5169    1.0802    1.0000
   -0.2374   -0.7078    1.0802    1.0000
   -0.0464   -0.5169    1.0802    1.0000
   -0.2374   -0.3259    1.0802    1.0000
   -0.2374   -0.5169    1.3503    1.0000
```

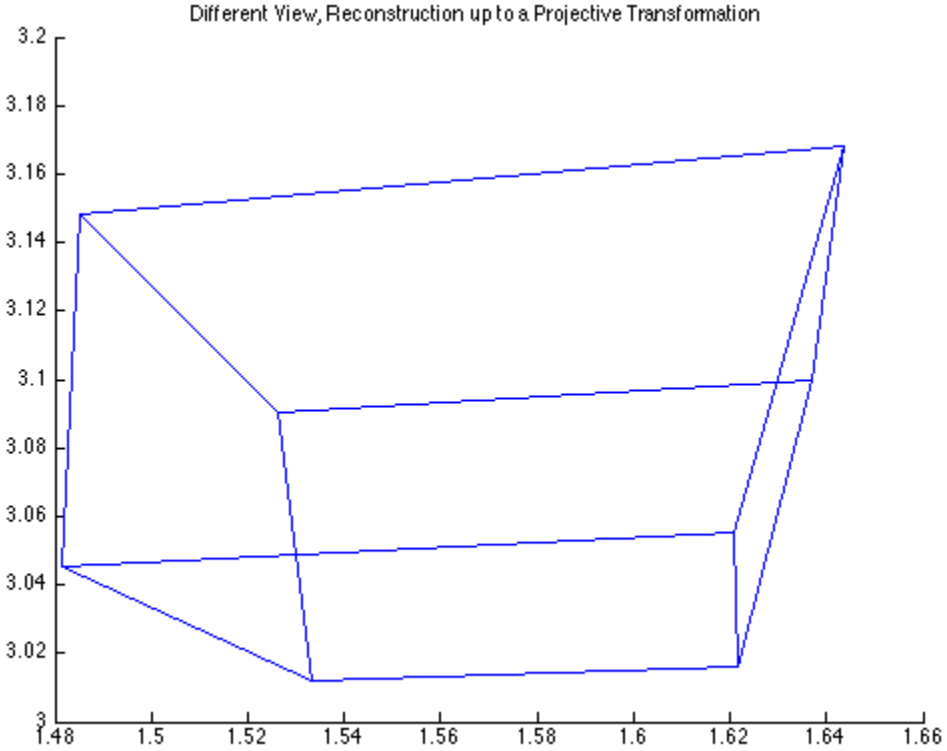

Original 3D Points
1
0.8
0.6
0.4
0.2
0
1
0.8
0.6
0.4
0.2
0
2
2.2
2.4
2.6
2.8
3

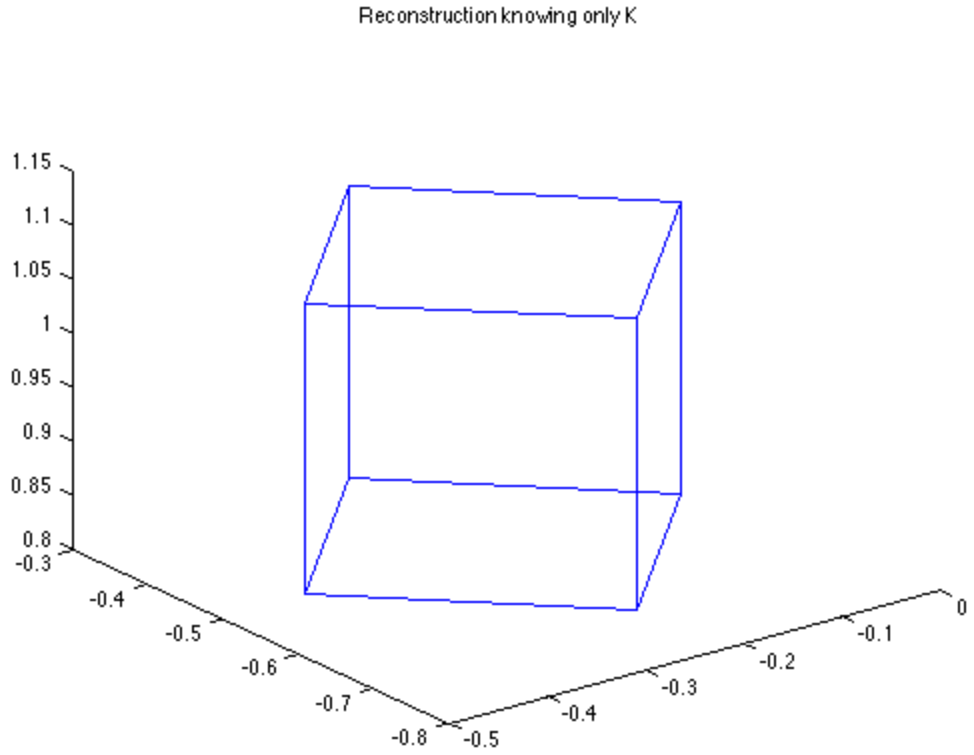

Euclidian Reconstruction
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
1
0.5
0
-0.5
2
2.5
3
3.5

Reconstruction up to a Projective Transformation



Different View, Reconstruction up to a Projective Transformation





Published with MATLAB® R2013b