Lund University

Computer Vision

FMAN95

Assignment 1

Nils Broman

31/01/2021

Contents

2 Points in Homogeneous Coordinates

2.1 E1

$$x_1 = \begin{bmatrix} 2 \\ -1 \end{bmatrix} \qquad x_2 = \begin{bmatrix} -3 \\ 2 \end{bmatrix} \qquad x_3 = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

x_4 can be interpreted as a point infinitely far away.

2.2 CE1

Matlab code found in appendix.

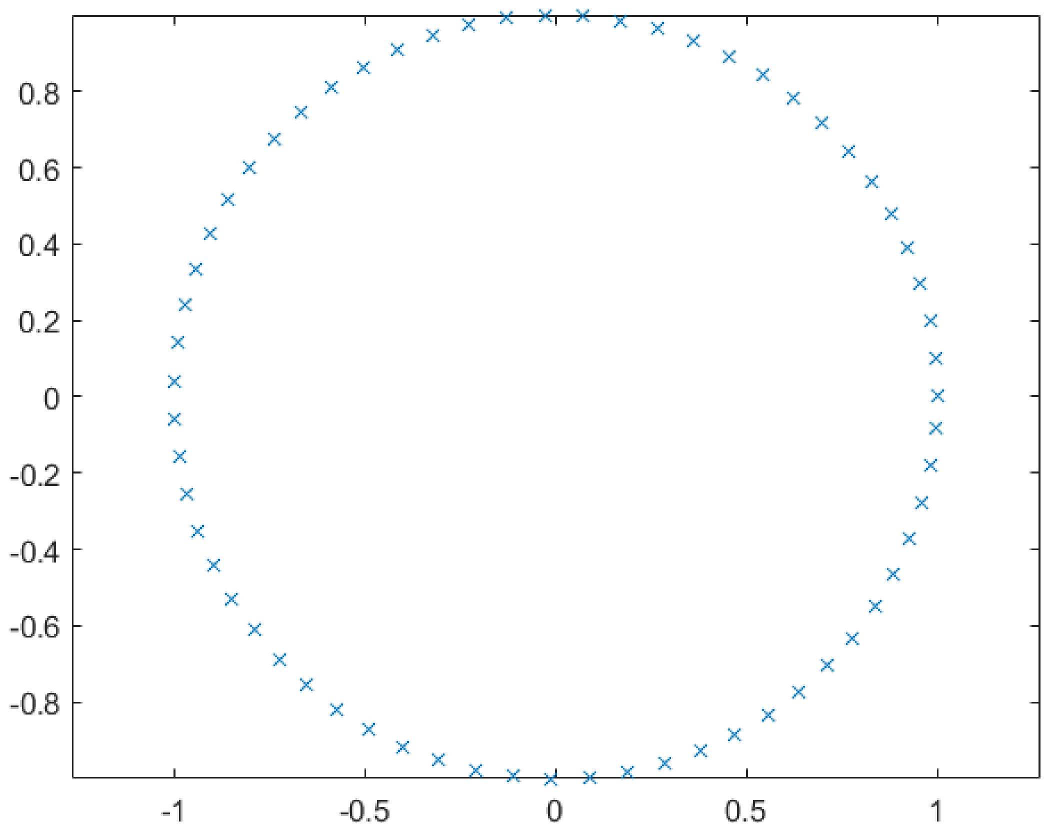

Figure 1

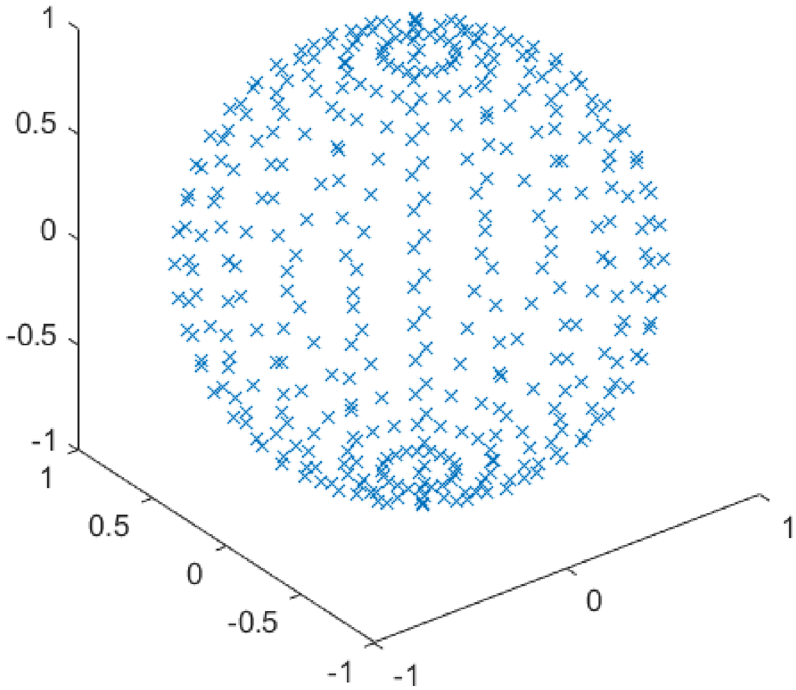

Figure 2

3 Lines

3.1 E2

$$\ell_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad \ell_2 = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \tag{1}$$

Point found where the scalar products $\ell^T \mathbb{x} = 0$ for both lines, leading to the sys. of eq. where we also let $z = t$

$$\begin{cases} x + y + z = 0 \\ 3x + 2y + z = 0 \\ z = t \end{cases} \Rightarrow \begin{cases} x = t \\ y = -2t \\ z = t \end{cases} \tag{2}$$

and deviding by z we find the intersection point $(1, -2)$ in $\mathbb{R}^2$.

For the lines

$$\ell_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \qquad \ell_2 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \tag{3}$$

it is easy to see that the sys. of eq. only solves for $z = 0$, meaning that the lines intersect infinitely far away (same idea as in E1).

Computing the line through $x_1 = (1, 1)$ and $x_2 = (3, 2)$ using the same idea as in 2, and since the coordinates are the same (given $z = 1$) we find the line

$$\ell_5 = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \tag{4}$$

3.2 E3

From the definition of the nullspace of a matrix we get the same sys. of eq. as in E2, meaning the full nullspace is

$$\mathcal{N}(A) = \{t(1, -2, 1), \forall\, t \in \mathbb{R}\} \tag{5}$$

which obviously contains the intersection point.

3.3 CE2

Matlab code in Appendix.

Distance $d = 8.2695$ px. Considering the resolution of the image, the distance is fairly close to zero.

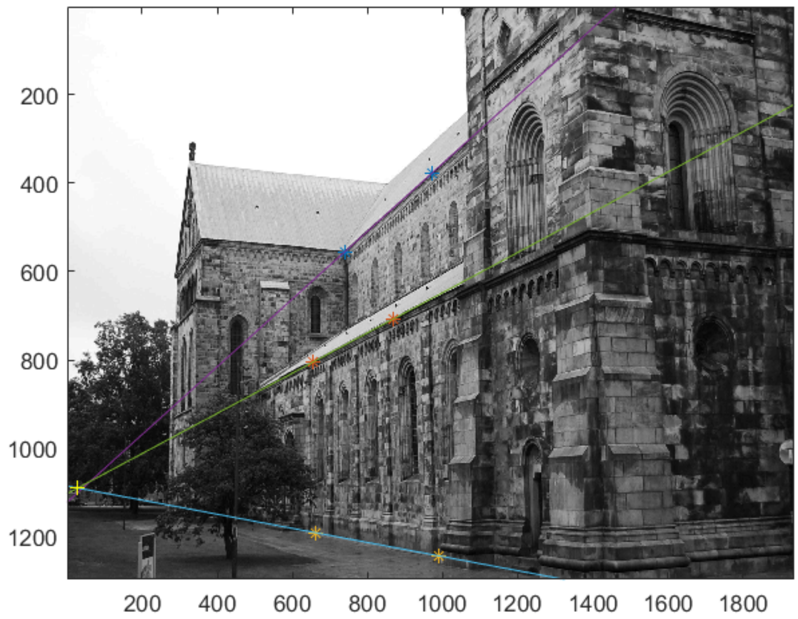

Figure 3

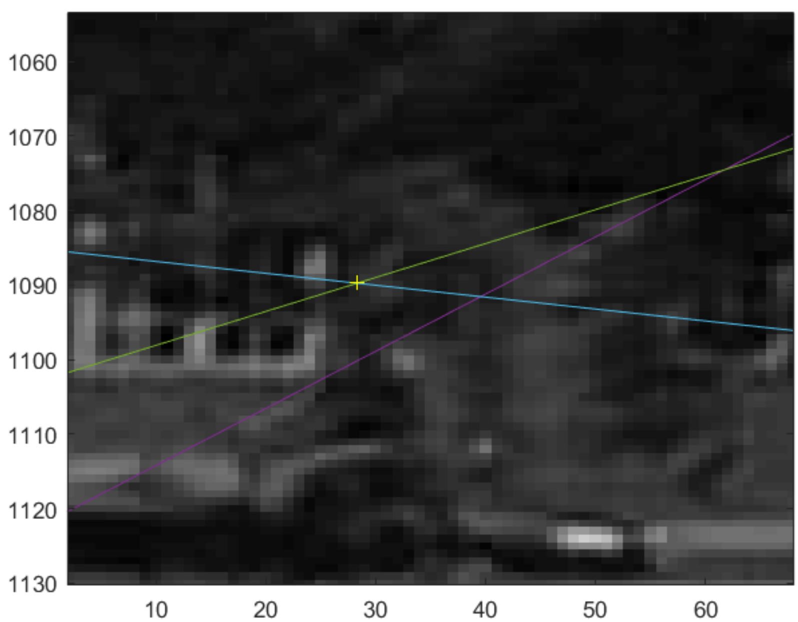

Figure 4

4 Projective Transformations

4.1 E4

$$\vec{y}_1 \sim H\vec{x}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \vec{y}_2 \sim H\vec{x}_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{6}$$

$$\ell_1 = t\begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} \qquad \ell_2 = t\begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \tag{7}$$

$$H^{-1} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix} \tag{8}$$

so

$$(H^{-1})^T \ell_1 = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} = \ell_2 \tag{9}$$

Proof that transformations presernve lines:

$$0 = \ell_1^T \vec{x} = \ell_1^T H^{-1} H \vec{x} = ((H^{-1})^T \ell_1)^T \vec{y} \;\Rightarrow$$

$$if\ \ell_1^T \vec{x} = 0\ \exists\, \ell_2 = (H^{-1})^T \ell_1\ \ such\ that\ \ \ell_2^T \vec{y} = 0$$

$\square$

4.2 CE3

H_1 preserves length between points. H_1 and H_2 preserve angles. H_1 , H_2 and H_3 maps parallel lines to parallel lines.

Euclidean :	H_1
Similarity :	H_1
Affine :	H_1 , H_2 and H_3
Projective :	H_4

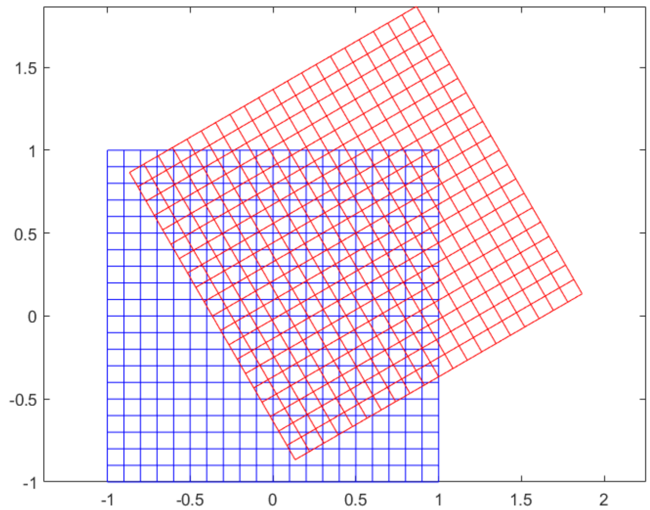

Figure 5

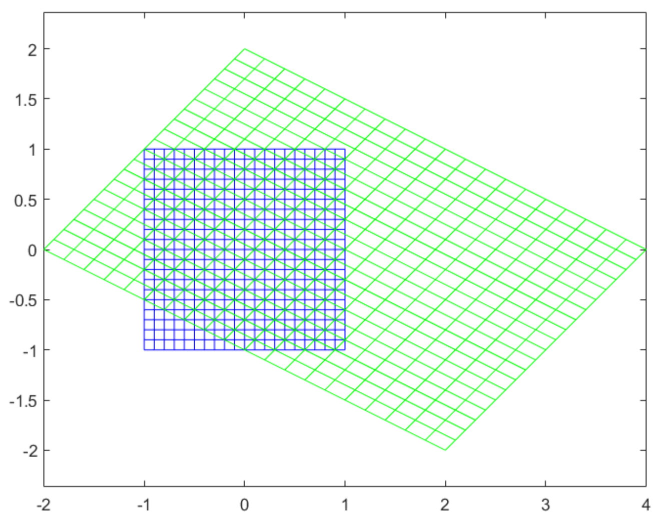

Figure 6

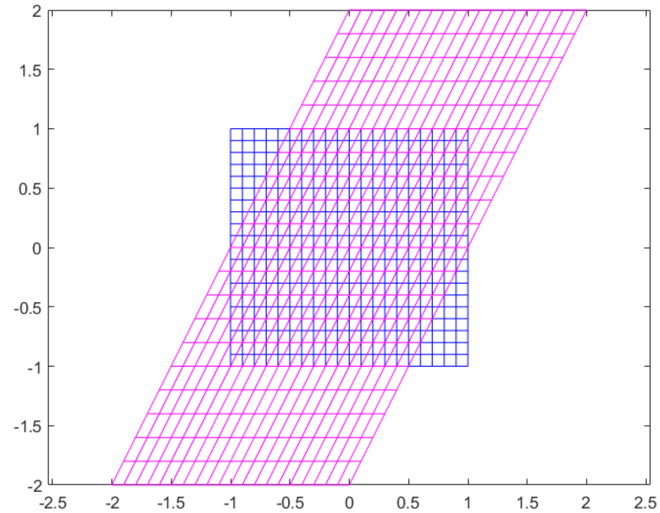

Figure 7

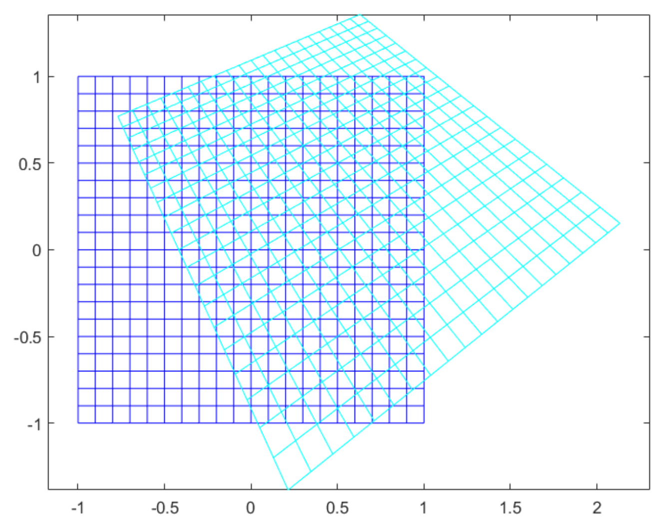

Figure 8

5 The Pinhole Camera

5.1 E5

Projections

$$X_1 : \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix} \qquad X_2 : \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} \qquad X_3 : \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \tag{10}$$

Geometric interpretation of the third projection is that the point is in the plane, containing the camera center, that is parallel to the image plane, therefore it's viewing ray will also be parallel to the image plane.

$$\text{Camera center} : \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \qquad \text{Principal axis} : \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

5.2 CE4

$$\begin{array}{rcc} \text{Camera} : & 1 & 2 \\ \text{Center} : & \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 6.6352 \\ 14.8460 \\ -15.0691 \end{bmatrix} \\ \text{Principial axis} : & \begin{bmatrix} 0.3129 \\ 0.9461 \\ 0.0837 \end{bmatrix} & \begin{bmatrix} 0.0319 \\ 0.3402 \\ 0.9398 \end{bmatrix} \end{array}$$

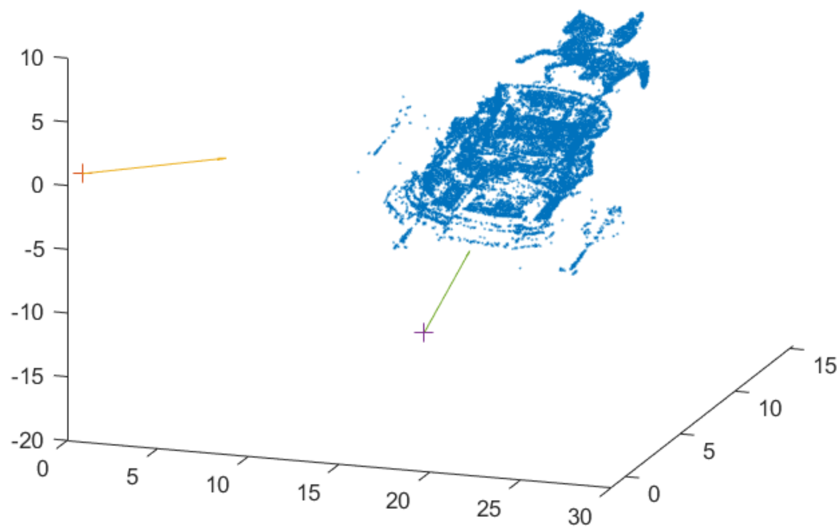

Figure 9

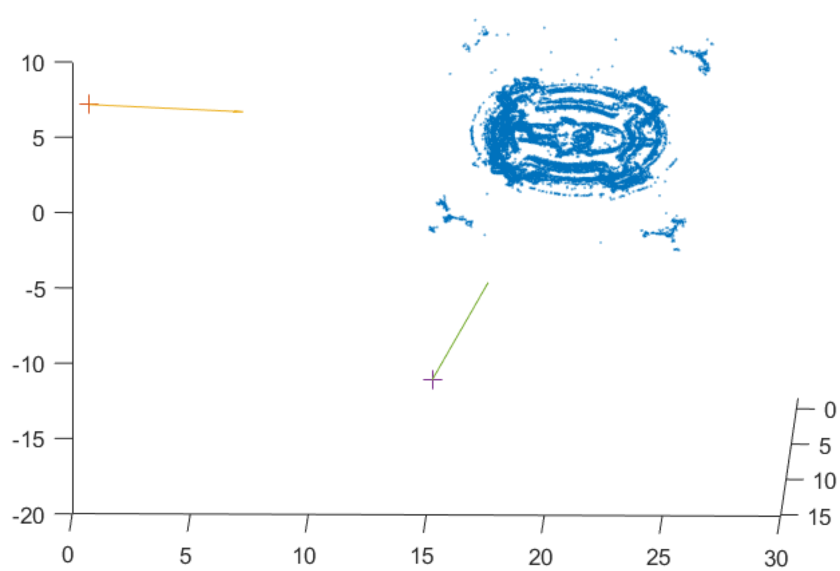

Figure 10

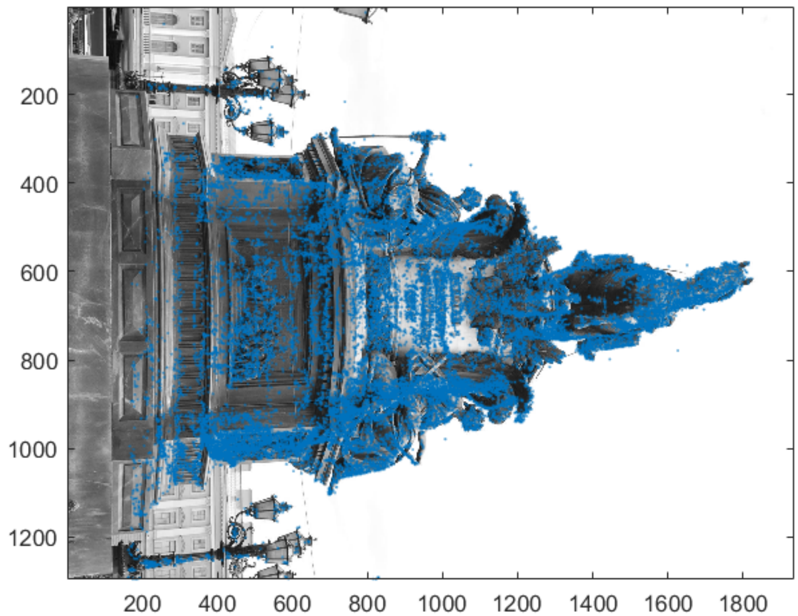

Figure 11

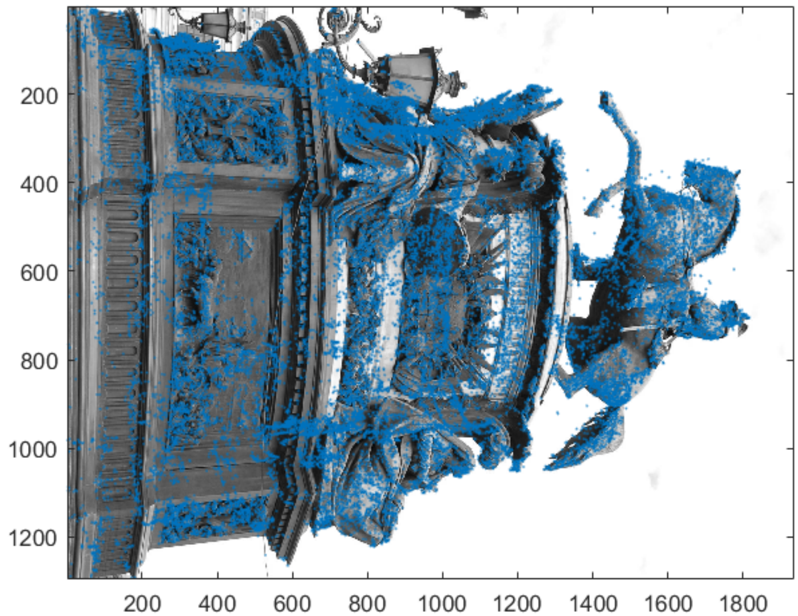

Figure 12

6 Appendix

Listing 1: main.m

```

%% CE 1
load('compEx1.mat')

%
flatmat2 = pflat(x2D);
figure(1)
plot(flatmat2(1,:), flatmat2(2,:),'x')
axis equal

flatmat3 = pflat(x3D);
figure(2)
plot3(flatmat3(1,:), flatmat3(2,:), flatmat3(3,:), 'x')
axis equal

%% CE 2

load('compEx2.mat')
im = imread('compEx2.JPG');

figure(3)
imagesc(im)
colormap gray
hold on

plot(p1(1,:),p1(2,:),'*')
plot(p2(1,:),p2(2,:),'*')
plot(p3(1,:),p3(2,:),'*')

l1 = linsolve([p1' ; 0 0 1],[0;0;1]);
l2 = linsolve([p2' ; 0 0 1],[0;0;1]);
l3 = linsolve([p3' ; 0 0 1],[0;0;1]);

rital(l1)
rital(l2)
rital(l3)

% Not parallel in 3D since not in 2D

```

```
intsec = linsolve([l2'; l3'; 0 0 1],[0;0;1]);
plot(intsec(1), intsec(2), '+', 'color', 'yellow')

d = abs(l1(1)*intsec(1) + l1(2)*intsec(2) + l1(3))/sqrt(l1(1)^2+l1(2)^2)

% Close to zero? Why/why not?

hold off

%% CE 3
clear all
load('compEx3.mat');

H1 = [sqrt(3) -1 1 ; 1 sqrt(3) 1 ; 0 0 2];
H2 = [1 -2 1 ; 1 1 0 ; 0 0 1];
H3 = [1 1 0 ; 0 2 0 ; 0 0 1];
H4 = [sqrt(3) -1 1 ; 1 sqrt(3) 1 ; 1/4 1/2 2];

start1 = pflat(H1*[startpoints ; ones(1,42)]);
end1 = pflat(H1*[endpoints ; ones(1,42)]);

start2 = pflat(H2*[startpoints ; ones(1,42)]);
end2 = pflat(H2*[endpoints ; ones(1,42)]);

start3 = pflat(H3*[startpoints ; ones(1,42)]);
end3 = pflat(H3*[endpoints ; ones(1,42)]);

start4 = pflat(H4*[startpoints ; ones(1,42)]);
end4 = pflat(H4*[endpoints ; ones(1,42)]);


figure(1)
plot([startpoints(1,:); endpoints(1,:)], ...
    [startpoints(2,:); endpoints(2,:)], 'b-');
hold on
plot([start1(1,:); end1(1,:)], [start1(2,:); end1(2,:)], 'r-');
axis equal
hold off

figure(2)
plot([startpoints(1,:); endpoints(1,:)], ...
    [startpoints(2,:); endpoints(2,:)], 'b-');
hold on
```

```
plot([start2(1,:); end2(1,:)], [start2(2,:); end2(2,:)], 'g-');
axis equal
hold off

figure(3)
plot([startpoints(1,:); endpoints(1,:)], ...
    [startpoints(2,:); endpoints(2,:)], 'b-');
hold on
plot([start3(1,:); end3(1,:)], [start3(2,:); end3(2,:)], 'm-');
axis equal
hold off

figure(4)
plot([startpoints(1,:); endpoints(1,:)], ...
    [startpoints(2,:); endpoints(2,:)], 'b-');
hold on
plot([start4(1,:); end4(1,:)], [start4(2,:); end4(2,:)], 'c-');
axis equal
hold off

%% CE 4

clear all
close all

load('compEx4.mat')

I1 = imread('compEx4im1.jpg');
I2 = imread('compEx4im2.jpg');

figure(1)
imagesc(I1)
colormap gray

figure(2)
imagesc(I2)
colormap gray

%%

camCent1 = pflat(null(P1))
camCent2 = pflat(null(P2))

```

```
camDir1 = P1(3,1:3)
camDir2 = P2(3,1:3)

Uc = pflat(U);

figure(3)
plot3(Uc(1,:), Uc(2,:), Uc(3,:), '.', 'Markersize',2)

hold on

plot3(camCent1(1), camCent1(2), camCent1(3), '+', 'Markersize', 8)
quiver3(camCent1(1),camCent1(2),camCent1(3),camDir1(1),camDir1(2),camDir1
    (3),7)

plot3(camCent2(1), camCent2(2), camCent2(3), '+', 'Markersize', 8)
quiver3(camCent2(1),camCent2(2),camCent2(3),camDir2(1),camDir2(2),camDir2
    (3),7)

hold off

%%

proj1 = pflat(P1*U);
figure(4)
imagesc(I1)
colormap gray
hold on
plot(proj1(1,:), proj1(2,:), '.', 'Markersize', 2)


proj2 = pflat(P2*U);
figure(5)
imagesc(I2)
colormap gray
hold on
plot(proj2(1,:), proj2(2,:), '.', 'Markersize', 2)
```

Listing 2: pflat.m

```
function proj_points = pflat(points)
%-----------------------------------------------------------------
% PURPOSE
%
%   Project points from given matrix
%
% INPUT:   points:              Matrix [m x n]
%
% OUTPUT   proj_points:         Matrix [m x n-1]
%-----------------------------------------------------------------

%   Nils Broman, 2021-01-22
%-----------------------------------------------------------------

proj_points=points(1:end-1,:)./points(end,:);

end
```