CMSC 733: Assignment #1

Due on Thursday, October 1, 2015

Aloimonos, Yiannis 11:00am

Kanishka Ganguly

09/28/2015

Contents

Problem 1

Listing 1: Image Projection

```
clear all
close all
clc

%{
GIVEN THAT

Calibrated Camera with f = 1 and origin at (0, 0, -3)
Camera rotated at -45 deg
%}

%%% World Coordinates of Cube %%%
P1 = [0 -0.5 0];
P2 = [1 -0.5 0];
P3 = [1 -0.5 1];
P4 = [0 -0.5 1];
P5 = [0 0.5 0];
P6 = [1 0.5 0];
P7 = [1 0.5 1];
P8 = [0 0.5 1];

%%% Making array of points above %%%
world = [P1 1;P2 1;P3 1;P4 1;P5 1;P6 1;P7 1;P8 1];

%%% Camera Rotation Matrix (Homogenous) %%%
R = [cosd(-45) 0 sind(-45) 0; 0 1 0 0; -sind(-45) 0 cosd(-45) 0; 0 0 0 1];

%%% Camera Translation Matrix (Homogenous) %%%
T = [1 0 0 0; 0 1 0 0; 0 0 1 -3; 0 0 0 1];

%%% Camera Focus Matrix (Homogenous) %%%
f = 1; % focus f = 1 %
F = [f 0 0 0; 0 f 0 0; 0 0 1 0];

%%%  Q1(a) Projection Matrix %%%
Pr = F * R * T;
fprintf('Ans. 1(a) The projection matrix is\n\n')
Pr

%%-----------------------------------------------------------------------%

%%% Converting WORLD to IMAGE coordinates %%%
fprintf('Ans. 1(b)\n')
for n = 1:8
    p = Pr * transpose(world(n,:));
    p_x = p(1)/p(3);
    p_y = p(2)/p(3);

    %%% Q1(b) Image Coordinates p5, p6, p7, p8 (Homogenous and Non-Homogenous) %%%
    switch(n)
```

```
        case {6, 8}
            fprintf('The Camera Coordinate p_%d (Non-Homogenous) is (%.3f, %.3f, %.3f) and p_%d (Homogenous) is (%.3f, %.3f)\n',n,p(1),p(2),p(3),n,p_x,p_y)
        case{5}
            fprintf('The Camera Coordinate p_%d (Non-Homogenous) is (%.3f, %.3f, %.3f) and p_%d (Homogenous) is (%.3f, %.3f)\n',n,p(1),p(2),p(3),n,p_x,p_y)
            % Store Non-Homogenous Coordinates of P5 %
            p_5 = [p(1) p(2) p(3)];
        case{7}
            fprintf('The Camera Coordinate p_%d (Non-Homogenous) is (%.3f, %.3f, %.3f) and p_%d (Homogenous) is (%.3f, %.3f)\n',n,p(1),p(2),p(3),n,p_x,p_y)
            % Store Non-Homogenous Coordinates of P7 %
            p_7 = [p(1) p(2) p(3)];
        otherwise
            fprintf('')
    end
end

%%------------------------------------------------------------------------%

%%% The first 3 columns Projection Matrix gives us the vanishing points of the 3 parallel lines %%%

%%% Q1(c) %%%
fprintf('\nAns. 1(c)\n')
fprintf('The non-homogenous coordinate of vanishing point 1 is (%.3f,%.3f,%.3f)\n',(Pr(1,1)), (Pr(2,1)), (Pr(3,1)))
fprintf('The non-homogenous coordinate of vanishing point 2 is (%.3f,%.3f,%.3f)\n',(Pr(1,2)), (Pr(2,2)), (Pr(3,2)))
fprintf('The non-homogenous coordinate of vanishing point 3 is (%.3f,%.3f,%.3f)\n',(Pr(1,3)), (Pr(2,3)), (Pr(3,3)))

%%------------------------------------------------------------------------%

%%% Q1(d) %%%
fprintf('\nAns. 1(d)\n')
%%% The Non-Homogenous Camera Coordinates of P5 and P7 are p_5 and p_7 %%%

% Finding Line Equation of p_5 - p_7 %
fprintf('X Equation: (x - x_1) / (x_2 - x_1) = t\n')
fprintf('Y Equation: (y - y_1) / (y_2 - y_1) = t\n')
fprintf('Z Equation: (z - z_1) / (z_2 - z_1) = t\n\n')

% Vector v parallel to line %
fprintf('v = <0.283, 0,-1.776>\n')
fprintf('r = <2.553, 0.5, -1.756> + <0.283,0,-1.776>t\n\n')

% In Vector Form %
fprintf('x = 2.553 + 0.283t\n')
fprintf('y = 0.5 + 0t\n')
```

```
94  fprintf('z = -1.756 - 1.776t\n\n')
95
96  % Taking Homogenous Form %
97  fprintf('As t --> infinity, we have\n')
98  fprintf('x" = x/z ~ 0.283/1.776 = 0.1273\n')
99  fprintf('y" = y/z ~ 0/1.776 = 0\n')
100 fprintf('z" = z/z = 1\n')
101
102 % Answer %
103 fprintf('The Coordinates of Vanishing Point of P5-P7 are (%.3f, %.3f,
        %.3f)\n\n',0.1273,0,1)
104
105 %%-----------------------------------------------------------------------%
```

(a)

We know that the camera has **f = 1**, **origin at (0, 0, -3)** and is rotated at **-45 °**
In above code, lines *13* to *20* define the world coordinates of the cube and line *26* takes into account the rotation of the camera at *-45 °*.
Line *29* takes care of the translation of the camera with origin at (0, 0, -3).
Also, we have the focal length of the camera as $f = 1$ which is used in Line *33*.
This gives us the projection matrix:

$$Pr = \begin{bmatrix} 0.5253 & 0 & -0.8509 & 2.5527 \\ 0 & 1.0000 & 0 & 0 \\ 0.8509 & 0 & 0.5253 & -1.5760 \end{bmatrix}$$

(b)

Using projection matrix `Pr` we convert each of the coordinates in `world` in Line *23* to get the non-homogenous image coordinates as:

- $p_5 = (2.553, 0.500, -1.576)$
- $p_6 = (3.078, 0.500, -0.725)$
- $p_7 = (2.227, 0.500, -0.200)$
- $p_8 = (1.702, 0.500, -1.051)$

Now, we divide the x, y and z coordinates by the z coordinate to get the homogenous image coordinates as:

- $p_{5homogenous} = (-1.620, -0.317, 1)$
- $p_{6homogenous} = (-4.245, -0.690, 1)$
- $p_{7homogenous} = (-11.150, -2.503, 1)$
- $p_{8homogenous} = (-1.620, -0.476, 1)$

(c)

Geometrically, the first three columns of the projection matrix gives us the 3 vanishing points corresponding to the 3 parallel lines.
So, we have:

$$Pr = \begin{bmatrix} 0.5253 & 0 & -0.8509 & 2.5527 \\ 0 & 1.0000 & 0 & 0 \\ 0.8509 & 0 & 0.5253 & -1.5760 \end{bmatrix}$$

with the red, green and blue columns representing the x, y and z coordinates of the vanishing points of the 3 parallel lines respectively.

(d)

Let $p_5 = (2.553, 0.500, -1.576)$ and $p_7 = (2.227, 0.500, -0.200)$ be the camera coordinates of P_5 and P_7.
To compute the vanishing point of line $P5 - P7$, we have the following equation for a line in 3-D:

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1} = t \tag{1}$$

where t is some parameter.

We take a vector $\vec{v}$ that is parallel to the line and we can write it as:

$$\vec{v} =< 0.283, 0, -1.776 > \tag{2}$$

In vector form, the line is:

$$\vec{r} =< 2.553, 0.5, -1.756 > + < 0.283, 0, -1.776 > t \tag{3}$$

$$\text{Or, we have}$$

$$x = 2.553 + 0.283t \tag{4}$$

$$y = 0.5 + 0t \tag{5}$$

$$z = -1.756 - 1.776t \tag{6}$$

We know, by definition, that the vanishing point of a line is the projection of that point at infinity. So, as parameter $t \to \infty$, we have the following homogenous coordinates:

$$\frac{x}{z} \approx 0.283/1.776 = 0.1593 \tag{7}$$

$$\frac{y}{z} \approx 0/1.776 = 0 \tag{8}$$

$$\frac{z}{z} = 1 \tag{9}$$

Thus, the homogenous coordinates of the vanishing point of line $P_5 - P_7$ are (0.1237, 0, 1).

(e)

To obtain two ideal vanishing points, the optical axis of the camera should be in line with any of the principal axes of the world coordinate, namely the X, Y or Z axis.
Essentially, when the camera is placed such that the image plane is parallel to any one face of the cube (only a square is visible), then we can obtain two ideal vanishing points.

Problem 2

Given that P is a 3×4 camera projection matrix and C is the camera center (in projective coordinates). We know that

$$P = [M| - MC] \tag{10}$$
$$\implies P = M[I| - C] \tag{11}$$

So, multiplying by C on both sides,

$$P \times C = M[I| - C] \times C \tag{12}$$
$$\implies P \times C = M[0] = 0 \tag{13}$$
$$\implies \underline{\underline{PC = 0}} \tag{14}$$

Problem 3

We know that P is a 3×4 camera matrix and

$$P = K \times R \times [I| - C] \tag{15}$$

Here,

$$K = \left.\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}\right\} \textbf{Calibration Matrix} \tag{16}$$

$$C = \left.\begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix}\right\} \textbf{Camera Center}$$

$$R = \left.\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}\right\} \textbf{Rotation Matrix}$$

Now, take the equation of the line joining the origin O $\equiv (0, 0, 0)$ and the camera center (world coordinate system) C $\equiv (C_x, C_y, C_z)$:

$$x = C_x t \tag{17}$$
$$y = C_y t \tag{18}$$
$$z = C_z t \tag{19}$$

Let us consider any line parallel to this line joining OC.
On it, any general point has a coordinate as $X \equiv (C_x t + x', C_y t + y', C_z t + z')$
The image coordinate for above general point can be obtained using projection matrix P as follows:

$$X_{img} = PX \tag{20}$$
$$X_{img} = K \times R \times [I| - C] \times X \tag{21}$$

$$X_{img} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & -C_x \\ 0 & 1 & 0 & -C_y \\ 0 & 0 & 1 & -C_z \end{bmatrix} \times \begin{bmatrix} C_x t + x' \\ C_y t + y' \\ C_z t + z' \end{bmatrix} \tag{22}$$

This gives us:

$$\begin{bmatrix} f(r_{11}(C_x t + x' - C_x)) + r_{12}(C_y t + y' - C_y) + r_{13}(C_z t + z' - C_z) \\ f(r_{21}(C_x t + x' - C_x)) + r_{22}(C_y t + y' - C_y) + r_{23}(C_z t + z' - C_z) \\ (r_{31}(C_x t + x' - C_x)) + r_{32}(C_y t + y' - C_y) + r_{33}(C_z t + z' - C_z) \end{bmatrix}$$

As parameter $t \to \infty$, the vanishing point V of the general point (in homogenous coordinates) becomes

$$V \equiv \begin{bmatrix} f(r_{11}C_x + r_{12}C_y + r_{13}C_z) \\ f(r_{21}C_x + r_{22}C_y + r_{23}C_z) \\ (r_{31}C_x + r_{32}C_y + r_{33}C_z) \end{bmatrix} \tag{23}$$

We know that the last (4^{th}) column of projection matrix is $P = -K \times R \times C$, or

$$\begin{bmatrix} -f(r_{11}C_x + r_{12}C_y + r_{13}C_z) \\ -f(r_{21}C_x + r_{22}C_y + r_{23}C_z) \\ -(r_{31}C_x + r_{32}C_y + r_{33}C_z) \end{bmatrix} \tag{24}$$

which is equivalent to the matrix denoted by Eqn. 23 when taken in homogenous coordinate form.

Or, we can say that the 4^{th} column of the projection matrix denotes, geometrically, the vanishing point of any line parallel to the line joining the camera center to the origin.

Extending that logic to a line parallel to the X-axis $\equiv (C_x t + x', y', z')$, Y-axis $\equiv (x', C_y t + y', z')$ or the Z-axis $\equiv (x', y', C_z t + z')$, we get, in homogenous coordinates, the first 3 columns of the projection matrix `P` respectively, which represent the vanishing points for the line.

Problem 4

Listing 2: Calibration

```
clear all
close all
clc
format shortE
format loose

%%% Declare Variables %%%
R_ = zeros();
K = zeros();

%%% CAMERA MATRIX %%%
fprintf('The Camera Matrix is')
P = [3.53*(10^2), 3.39*(10^2), 2.77*(10^2), 1.44*(10^6); ...
    -1.03*(10^2), 2.33*(10^1), 4.59*(10^2), -6.32*(10^5); ...
    7.07*(10^-1), -3.53*(10^-1), 6.12*(10^-1), -9.18*(10^2)
    ]
%%% We know that P = [M | -MC] %%%
%%% So, we discard last column of P %%%
M = P(:,(1:3));
MC = P(:,4);

%%% r-q Decomposition on M %%%
[R, Q] = rq(M);

%%% Now, K = R and R_ = Q matrices for M = K x R_ %%%
K = R
R_ = Q;

%%% To get C (camera center) %%%
C = -(-MC)\M;

%%% So, we have the camera parameters as follows %%%
fprintf('Focal Lengths (pixels): Ax = %.3f and Ay = %.3f\n',K(1,1),K(2,2))
fprintf('Image Center (pixels): X0 = %.3f and Y0 = %.3f\n', K(1,3),K(2,3))
fprintf('Skew Parameter: S = %.3f\n', K(1,2))
fprintf('Camera Center: (%f, %f, %f)\n', C(1), C(2), C(3))
```

We are given a 3×4 camera matrix as

$$P = \begin{bmatrix} 3.53\text{E}2 & 3.39\text{E}2 & 2.77\text{E}2 & 1.44\text{E}6 \\ -1.03\text{E}2 & 2.33\text{E}1 & 4.59\text{E}2 & -6.32\text{E}5 \\ 7.07\text{E}{-1} & -3.53\text{E}{-1} & 6.12\text{E}{-1} & -9.18\text{E}2 \end{bmatrix}$$

We know that

$$P = [M| - MC] \tag{25}$$

where M is defined by the 3×3 matrix in blue and $-MC$ is defined by the 1×3 matrix in red.

Problem 4 continued on next page...

Also, we know that

$$M = K \times R \tag{26}$$

where K is the calibration matrix and R is the rotation matrix.
Now, we perform RQ-Decomposition on matrix M to get back the matrices K and R. RQ-Decomposition gives us two matrices Q and R, one orthogonal and one upper-triangular in nature.
So, let $K = R_{QR}$ and let $R_M = Q$

Also, the matrix $-MC$ contains the camera center C which we retrieve by solving:

$$AX = B \tag{27}$$

$$\text{where AX = -MC and B = M}$$

$$\implies C = -(-MC)/M \tag{28}$$

Now,

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where α_x and α_y are the "Focal Lengths" in pixels, x_0 and y_0 are the coordinates of the image center in pixels and s is the skew parameter.
So, comparing with the matrix K obtained after RQ-Decomposition, we have:

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -467.165 & -90.936 & -299.578 \\ 0 & -426.437 & -199.962 \\ 0 & 0 & 1 \end{bmatrix}$$

Thus, we have the following:

$$\textbf{Calibration Parameters} \begin{cases} \textbf{Focal Lengths} \quad (\alpha_x, \alpha_y) = (-467.165, -426.437) \\ \textbf{Image Centers} \quad (x_0, y_0) = (-299.578, -199.962) \\ \textbf{Skew Parameter} \quad s = -90.936 \end{cases} \tag{29}$$

$$\textbf{Camera Center} \begin{cases} (C_x, C_y, C_z) = (0.000232, 0.000191, 0.000044) \end{cases} \tag{30}$$

Problem 5

It is given that three known points $\{A, B, C\}$ are collinear on a 3-D line and their images formed are $\{a, b, c\}$. Assume D to be the vanishing point of the 3-D line ABC and its image is d.
Let us say that:

$$A = (A_x, A_y, A_z) \tag{31}$$
$$B = (B_x, B_y, B_z) \tag{32}$$
$$C = (C_x, C_y, C_z) \tag{33}$$

are the coordinates of A, B and C respectively.

Now, let f be the focal length of the camera and by convention, the optical axis lies along Z-axis.
So, we have:

$$a = \left(f\frac{A_x}{A_z}, f\frac{A_y}{A_z}, f\right) \tag{34}$$
$$b = \left(f\frac{B_x}{B_z}, f\frac{B_y}{B_z}, f\right) \tag{35}$$
$$c = \left(f\frac{C_x}{C_z}, f\frac{C_y}{C_z}, f\right) \tag{36}$$

So, the line equation for the image points can be written as:

$$x = f\frac{C_x}{C_z} + \left(f\frac{C_x}{C_z} - \frac{B_x}{B_z}\right)t \tag{37}$$
$$y = f\frac{C_y}{C_z} + \left(f\frac{C_y}{C_z} - \frac{B_y}{B_z}\right)t \tag{38}$$
$$\tag{39}$$
$$\implies t = \frac{cd}{bc} \tag{40}$$
$$\implies t = \frac{(u+1)q - (u+1)}{u+1-q} \tag{41}$$

where $u = \frac{ab}{bc}$ and $q = \frac{AC}{BC}$

Thus, vanishing point d is:

$$\left(f\frac{C_x}{C_z} + \left(f\frac{C_x}{C_z} - \frac{B_x}{B_z}\right)t_1, f\frac{C_y}{C_z} + \left(f\frac{C_y}{C_z} - \frac{B_y}{B_z}\right)t_1, f\right) \tag{42}$$

Problem 6

(a)

We have two sets of four known coplanar points, namely $\{A, B, C, D\}$ and $\{E, F, G, H\}$ and origin O. Considering first set of points $\{O, A, B, C, D\}$, we have 4 lines OA, OB, OC and OD passing through O. The cross-ratio for the angles represented by above lines can be given as:

$$CR_O(A, B, C, D) = \frac{sin(OA, OC)sin(OB, OD)}{sin(OA, OD)sin(OB, OC)} \tag{43}$$

In a projective transformation, the cross-ratio remains invariant.
So, assuming that the point M in the scene is projected to a point M' coplanar to the points A, B, C, D, then we can find M' using the cross-ratio as:

$$CR_O(b, c, d, m) = CR_A(B, C, D, M') \tag{44}$$
$$CR_O(a, c, d, m) = CR_B(A, C, D, M') \tag{45}$$
$$CR_O(a, b, d, m) = CR_C(A, B, D, M') \tag{46}$$

Now, from the problem, since it is given that we know A, B, C, D and a, b, c, d, we can calculate the coordinates of M'.

Similarly, assuming that point M is projected to a point M'' coplanar to the points E, F, G, H, then we find M'' using cross-ratio as:

$$CR_O(f, g, h, m) = CR_E(F, G, H, M'') \tag{47}$$
$$CR_O(e, g, h, m) = CR_F(E, G, H, M'') \tag{48}$$
$$CR_O(e, f, h, m) = CR_G(E, F, H, M'') \tag{49}$$

Thus, the viewing ray Om can be determined by the points M' and M''.

(b)

Consider the projection A' of point A on the plane E, F, G, H. We can obtain the point A' as:

$$CR_e(f, g, h, a) = CR_E(F, G, H, A') \tag{50}$$
$$CR_f(e, g, h, a) = CR_F(E, G, H, A') \tag{51}$$
$$CR_g(e, f, h, a) = CR_G(E, F, H, A') \tag{52}$$

From Subsection 6(a), we have M' and M'' and also we now have A and A'. This gives us two rays AA' and $M'M''$ from the camera center.
Thus the intersection of $M'M''$ and AA' gives us the camera center O.

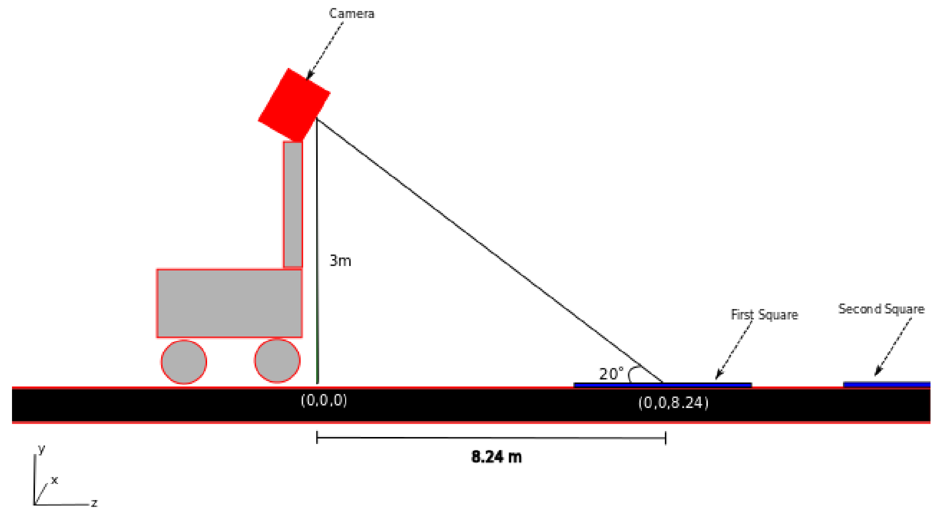
Problem 7

Listing 3: Robot Vehicle

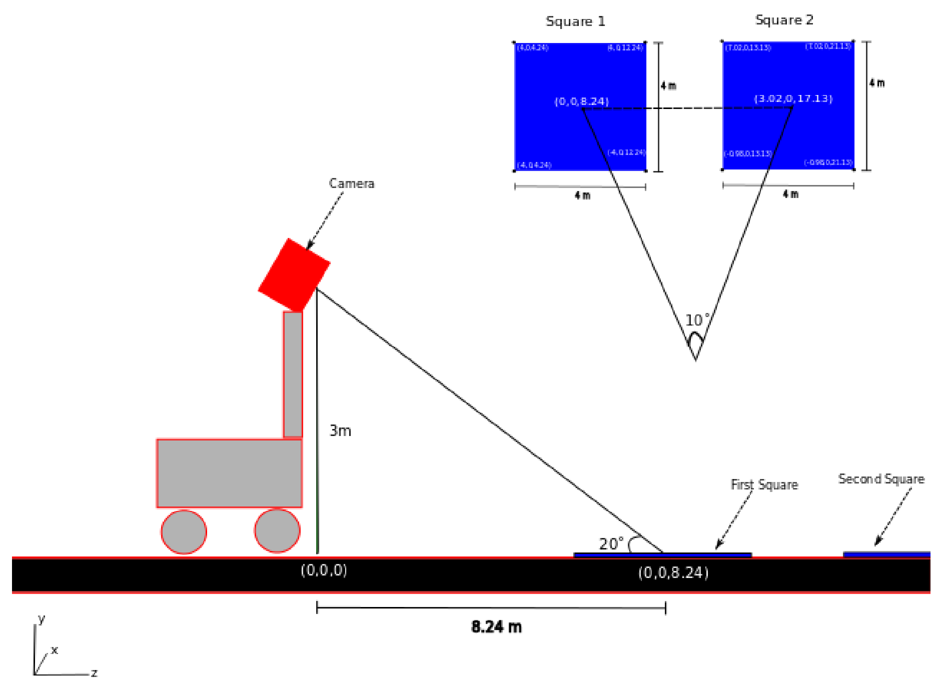
```
clear all
close all
clc
format shortE
format loose

%%% 7(a) - Compute Pixel Positions %%%

% Calibration Parameters %
f_x = 600/2; % x focal length in pixels %
f_y = 500/2; % y focal length in pixels %
x_0 = -690; % x center of image in pixels %
y_0 = 690; % y center of image in pixels %
s = 0; % skew parameter %

% Calibration Matrix K %
K = [f_x s x_0 0 ; 0 f_y y_0 0; 0 0 0 1];

% Camera Tilted at 20 degrees %
R = [1 0 0 0; 0 cosd(20) sind(20) 0; 0 -sind(20) cosd(20) 0; 0 0 0 1];

% Camera Translated at 3m along y-axis %
T = [1 0 0 0; 0 1 0 -3; 0 0 1 0; 0 0 0 1];

% Projection Matrix %
P = K*R*T;

% Finding Coordinates of corners of regular squares %

% Square 1 %
% P_1 center = 3/tand(20);%
P1_center = [0 0 8.24];
P1_1 = [-2 0 6.24 1];
P1_2 = [-2 0 10.24 1];
P1_3 = [2 0 10.24 1];
P1_4 = [2 0 6.24 1];

originalX1 = [P1_1(1),P1_2(1),P1_3(1),P1_4(1),P1_1(1)];
originalY1 = [P1_1(3),P1_2(3),P1_3(3),P1_4(3),P1_1(3)];
figure
plot(originalX1, originalY1);
hold on
% Square 2 %
% We get distance between two centers of squares at 17.4 %
% Since road is curved, we get 17.4*cosd(10) and 17.4*sind(10) as coords %

P2_center = [-3.02 0 25.37]';

% Rotate each point since road is curved %
R_2 = [cosd(-10) 0 sind(-10); 0 1 0; -sind(-10) 0 cosd(-10)];
```

```
L_1 = [2 0 -2]';
L_2 = [2 0 2]';
L_3 = [-2 0 2]';
L_4 = [-2 0 -2]';

P2_1 = (R_2 * L_1) + P2_center;
P2_2 = (R_2 * L_2) + P2_center;
P2_3 = (R_2 * L_3) + P2_center;
P2_4 = (R_2 * L_4) + P2_center;

originalX2 = [P2_1(1),P2_2(1),P2_3(1),P2_4(1),P2_1(1)];
originalY2 = [P2_1(3),P2_2(3),P2_3(3),P2_4(3),P2_1(3)];
plot(originalX2, originalY2);
hold off

% Applying Projection Matrix P on Coordinates of Squares %

% Square 1 %
p1_1 = P * P1_1';
p1_1 = p1_1((1:2))

p1_2 = P * P1_2';
p1_2 = p1_2((1:2))

p1_3 = P * P1_3';
p1_3 = p1_3((1:2))

p1_4 = P * P1_4';
p1_4 = p1_4((1:2))

X1_points = [p1_1(1),p1_2(1),p1_3(1),p1_4(1),p1_1(1)];
Y1_points = [p1_1(2),p1_2(2),p1_3(2),p1_4(2),p1_1(2)];

% Square 2 %
p2_1 = P * [P2_1;1];
p2_1 = p2_1((1:2))

p2_2 = P * [P2_2;1];
p2_2 = p2_2((1:2))

p2_3 = P * [P2_3;1];
p2_3 = p2_3((1:2))

p2_4 = P * [P2_4;1];
p2_4 = p2_4((1:2))

X2_points = [p2_1(1),p2_2(1),p2_3(1),p2_4(1),p2_1(1)];
Y2_points = [p2_1(2),p2_2(2),p2_3(2),p2_4(2),p2_1(2)];


%%% 7(b) - Plot Pixel Positions %%%
figure
plot(X1_points, Y1_points)
```

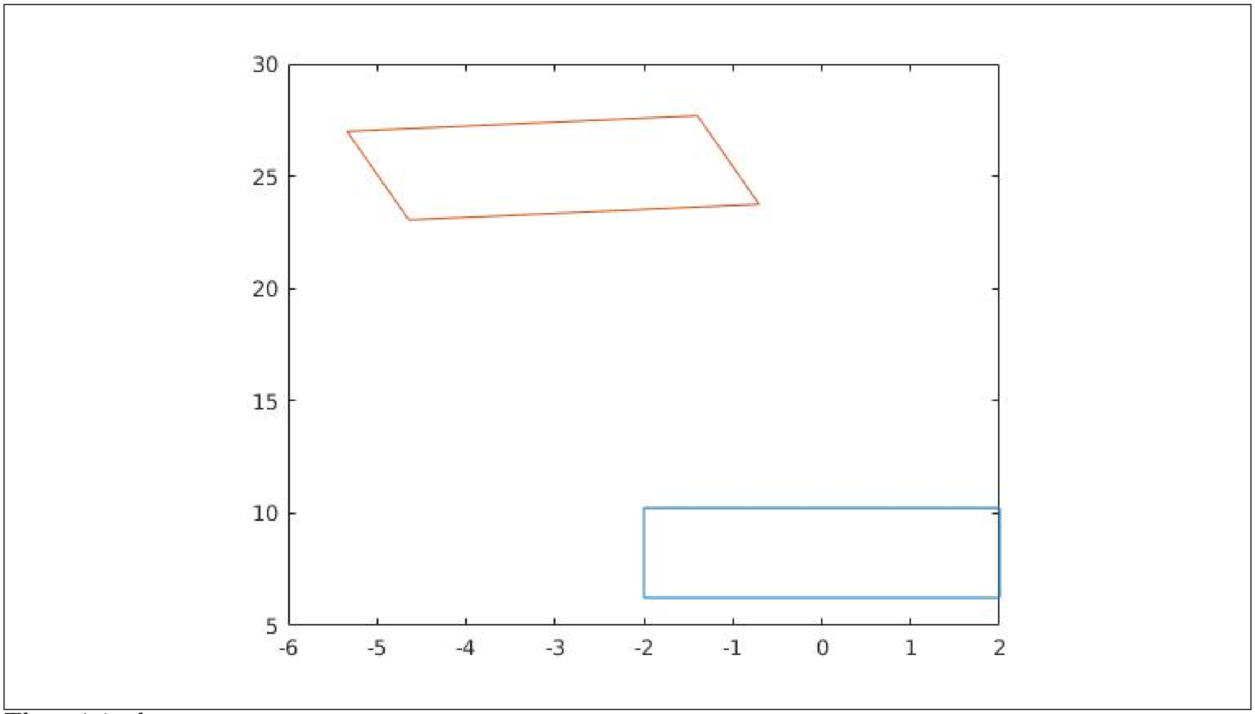
```
hold on
plot(X2_points, Y2_points)
```



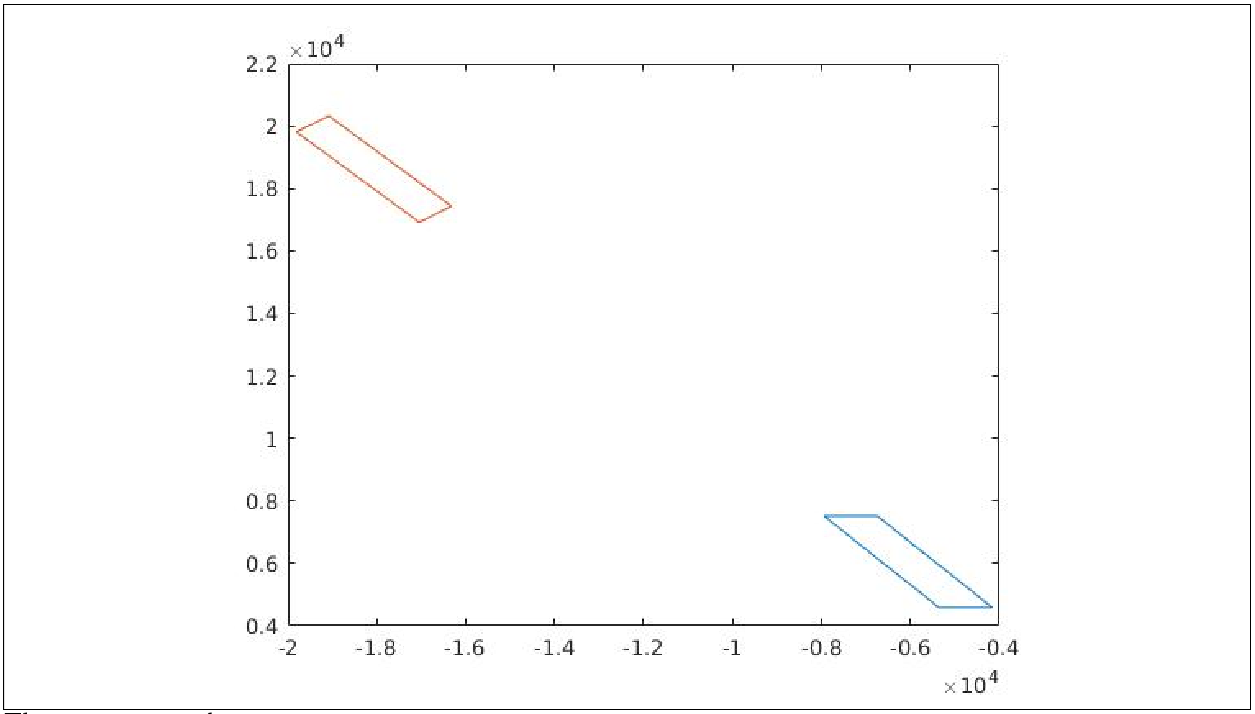
The basic setup.



The dimensions of the squares.

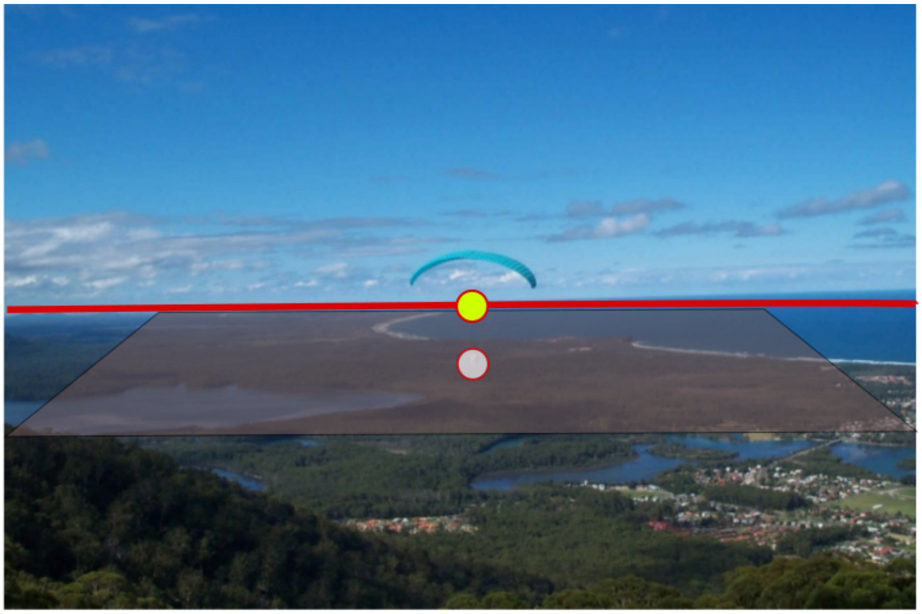


The original squares.



The reconstructed squares.

Problem 8



As can be seen from the modified figure, the brown plane shows the plane of the optical axis, which meets at the horizon to give the vanishing point in green.
All lines on this plane converge at the vanishing point.
The white circle represents the parachuter, who is lower than the vanishing point.
Thus, the parachuter is lower than the person taking the picture