Robotic Vision - Assignment 2

Author:Raghavendra Sriram
Date : 10/6/2013

Q 1) a)
Matlab code:

```
% Asgnmnt#2 - Ex1
% Author: Raghavendra Sriram
% Date  : 10/6/2013

clc;
clear all;
close all;
imtool close all;

%%Given Data
ku = 10^4; % Initialize the intrinsic camera parameters
kv = 10^4;
f = 10^(-2);
u = [470,395];
v = [270,255];
u0 = 320;
v0 = 240;
d = 1;

syms D Z1;
L1 = (f*ku*D/Z1) + u0 - u(1);
L2 = (f*ku*D/(Z1+d)) + u0 - u(2);
%% Solve the expressions and print the values of D and Z1
A = solve(L1,L2,D,Z1);
fprintf('D = %f\t\t Z1 = %f\n\n',double(A.D),double(A.Z1));
clear L1;
clear L2;
%% Give the second set of expressions
syms Y1 Y2;
L1 = (f*kv*Y1/A.Z1) + v0 - v(1);
L2 = (f*kv*Y2/(A.Z1 + d)) + v0 - v(2);
B = solve(L1,L2,Y1,Y2);

K = [f*ku  0    u0;
      0   f*kv  v0;
      0    0     1];

Xc = double(A.D);%   3-D point co-ordinates
Yc = double(B.Y1);
Zc = double(A.Z1);
C_X = [ Xc; Yc; Zc ];

Camera_3D(C_X,K,u0,v0);%   Generate the camera and the point L1

Zc = double(A.Z1) + d;
C_X = [ Xc; Yc; Zc ];

Camera_3D(C_X,K,u0,v0);%   Generate the camera and the point L2


%   The  program above simulates a pin-hole camera model using
%   the EGT. It calculates the distance between the camera and the street edge. It
also
%   calculates the distance between the street and the camera itself. Both
%   these calculated values (D and Z1) are displayed on the command window. The
%   function "Camera_3D.m" simulates the camera and the projections on
%   points L1 and L2 on the side of the street.
```

Function : Camera_3D

```
function Camera_3D(C_X,K,u0,v0)
%   The following MATLAB function generates the 3D projections of points
%   and generates the camera and frame of reference using the "Epipolar
%   Geometry Toolbox".
%% Create the figures and set their properties
figure(1);
xlabel('x'); ylabel('y');  zlabel('z');
title('Camera Placement & Perspective Projection');
axis equal;  axis([-1 3 -1 3 -1 1]);
view(48,42);
%% Figure on the Image Plane
figure(2);
title('Image Plane');
axis([0 u0*2 0 v0*2]);
grid on; hold on;
%% Placement of Camera and plot of points
figure(1);  hold on;
W_X = rotox(-pi/2)*C_X;
plot3(W_X(1), W_X(2), W_X(3), 'r+');
%% Left Camera position parameters
R = eye(3);
t = [0,0,0]';
H = f_Rt2H(R,t);
scale = 1/2;
f_3Dframe(H,'b',scale,'_{c}');
f_3Dcamera(H,'b',scale);
%% Perspective Projection for Left Camera
U_L = f_perspproj(W_X,H,K,2);
figure(2);
hold on;
plot(U_L(1), U_L(2), 'rO');
```

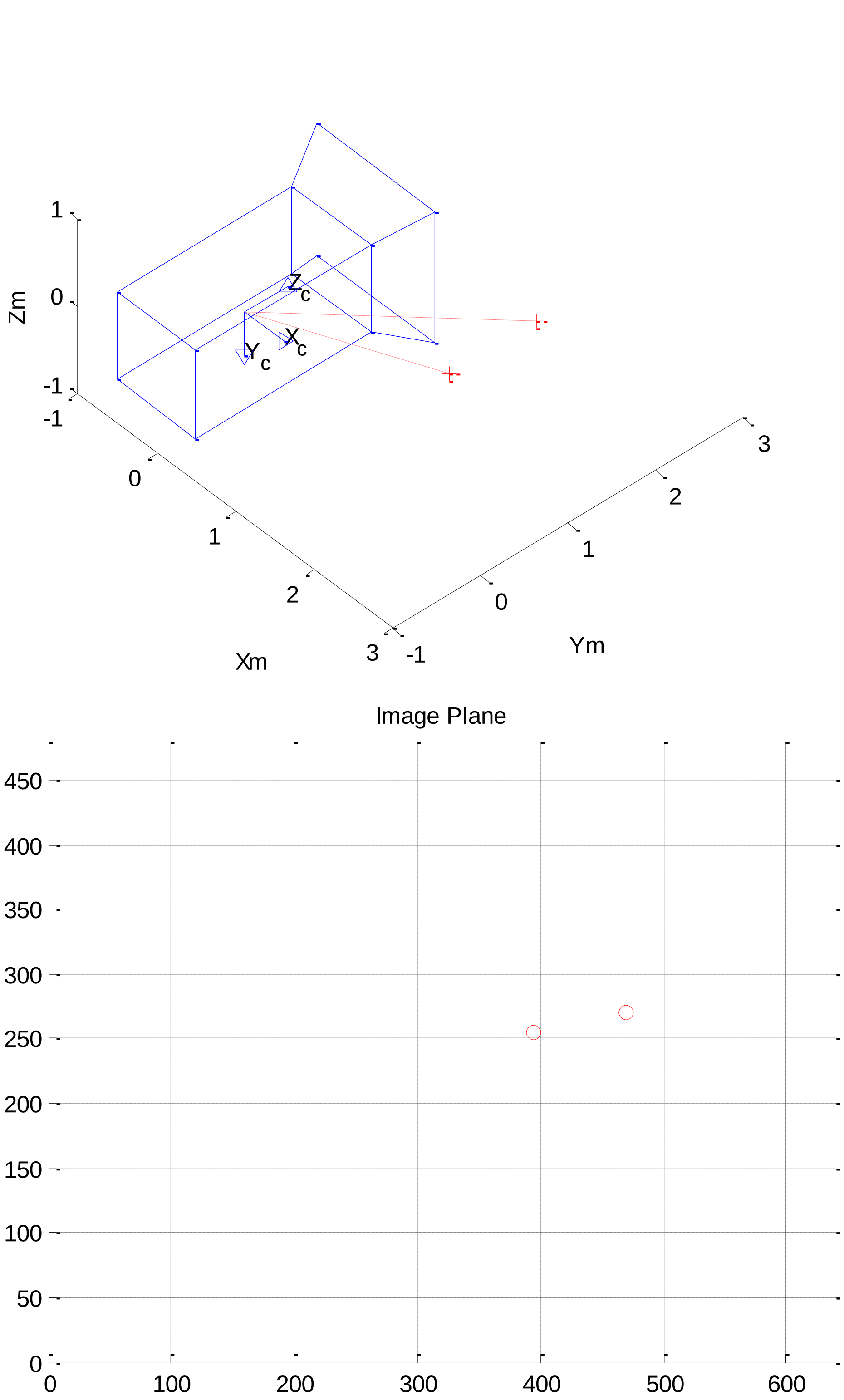
```
-------------------------------------------------------------------------------------
```

```
Solution :

D = 1.500000         Z1 = 1.000000

>>
```

```
Images :
```



Camera Placement & Perspective Projection

Image Plane