Homework 2
Project Report
3D Object geometry via triangulation

Fnu Karan
NetID: kx361

March 14, 2017

Date Performed:	February 27, 2017
Instructor:	Professor Gerig

1 Objective

The objective is to come with the 3D Geometry of a 3D object, with its images taken by a stereo pair of cameras.

2 Experimental Procedure and Data Capture

An object, similiar to a cuboid is put on a plane, and its two images are taken from a stereo pair of cameras, ensuring that at least 7 same corners are visible from both the images as shown in Figure 1 and 2, (with the corresponding points that were taken). The camera was shifted in X direction, and hence the disparity would also be in X direction in the images.

The origin is at the lens of the camera such that, Z is towards the object, and X is in the direction where the camera is being moved, and Y is pointing upwards.

The points to be measured are shown. Total seven points are chosen. The distance reorded from left camera origin to the right camera origin is 124.46 mm.

The camera used was of the phone iPhone 6S. It is 12MP and we can see it in the picture resolution as specified above, and it uses "Stacked back-illuminated CMOS image sensor", and its size is 4.80 x 3.60 mm (1/3"). The pixel size is 1.22 μm. The aperture is f/2.2. The focal length is 4.15 mm.

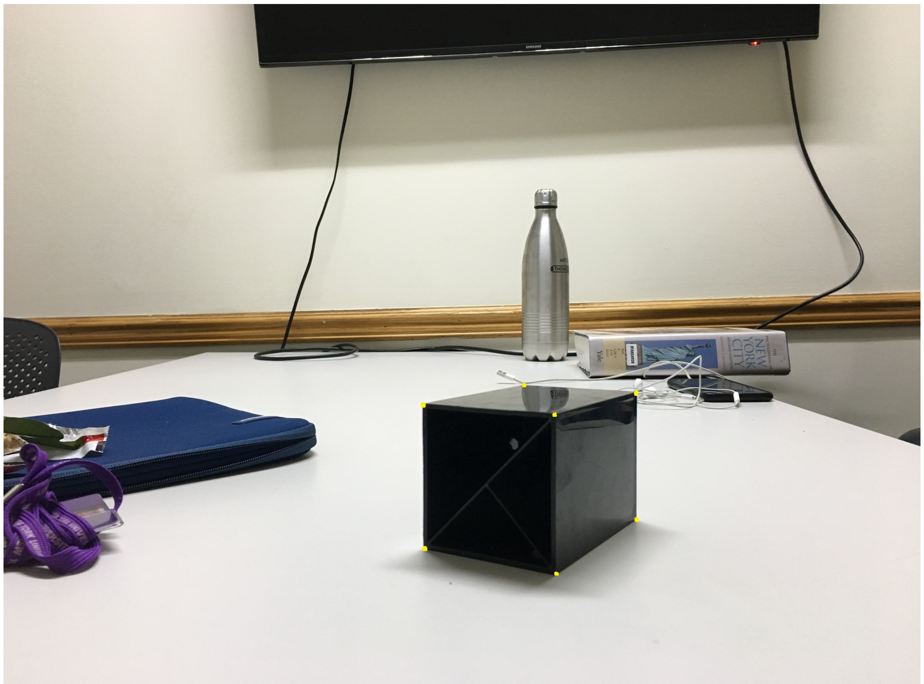

Figure 1: Left input image.

3 Calculation of Disparity

Once the 7 points are taken from both the images, then we could observe that the disparity lies only in the X direction, since the camera was moved in the X direction only. There is a little disparity in the Y direciton as well, but that is just because of human error and it can be ignored.

The points had these following coordinates in the images. For left they are:

$$left\_pixels = \begin{bmatrix} 1845.5 & 1755.5 \\ 2421.5 & 1803.5 \\ 2427.5 & 2511.5 \\ 1851.5 & 2391.5 \\ 2283.5 & 1683.5 \\ 2781.5 & 1707.5 \\ 2781.5 & 2265.5 \end{bmatrix} \tag{1}$$

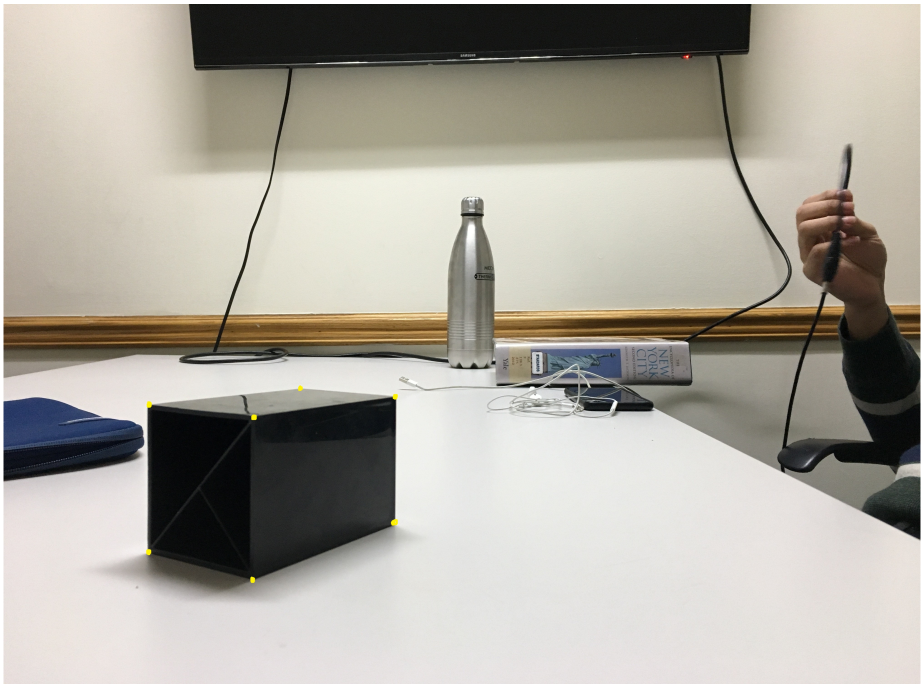

Figure 2: Right input image.

For the right, they are:

$$right\_pixels = \begin{bmatrix} 633.5 & 1761.5 \\ 1101.5 & 1815.5 \\ 1095.5 & 2529.5 \\ 639.5 & 2409.5 \\ 1311.5 & 1695.5 \\ 1725.5 & 1725.5 \\ 1719.5 & 2283.5 \end{bmatrix} \tag{2}$$

Here, the catch is that these are the pixel values for both the images, but we need it in mm. So here comes the camera calibration into account. Since, I used the same camera to take these images as the first homework assignment, so I know the respected parameters for these images. I used those pixel densities for getting these pixel values in mm.

Now, we could subtract the X values to get the disparity, for all the 7 points.

$$d = \begin{bmatrix} 1.44 \\ 1.57 \\ 1.58 \\ 1.44 \\ 1.15 \\ 1.25 \\ 1.26 \end{bmatrix} \tag{3}$$

4 Estimation of the 3D coordinates

Since we have the disparity then we can also calculate the coordinates of all the seven points. We use the principle of triangulation as shown in the figure.

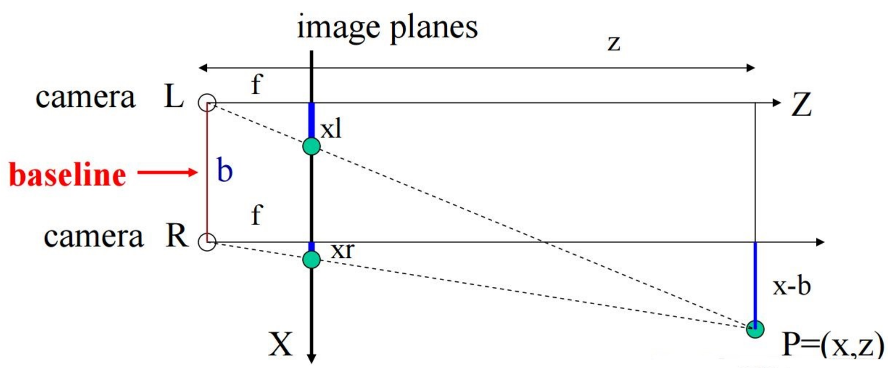

Figure 3: Principle of triangulation.

We can use the following equations to calculate the 3D coordinates of the seven points with respect to camera frame.

$$Z = \frac{fb}{x_l - x_r} = \frac{fb}{d} \tag{4}$$

$$X = \frac{x_l}{f} Z \tag{5}$$

$$Y = \frac{y_l}{f} Z \tag{6}$$

We can use the points we got from the above equations to plot the 3D model for the image. The figure obtained from MATLAB has been attached in the zip file attached.

5 Results

The estimated coordinates is as follows:

$$\mathcal{P} = \begin{bmatrix} 0.529 & 0.503 & 357.976 \\ 0.694 & 0.517 & 328.687 \\ 0.696 & 0.720 & 325.726 \\ 0.531 & 0.686 & 357.976 \\ 0.655 & 0.482 & 446.365 \\ 0.797 & 0.489 & 410.859 \\ 0.797 & 0.649 & 408.538 \end{bmatrix} \tag{7}$$

Each row corresponds to one point, and each column corresponds to X, Y and Z coordinates respectively.

Now, we can plot these points to get the 3D model for the object. This is shown in figure 4.

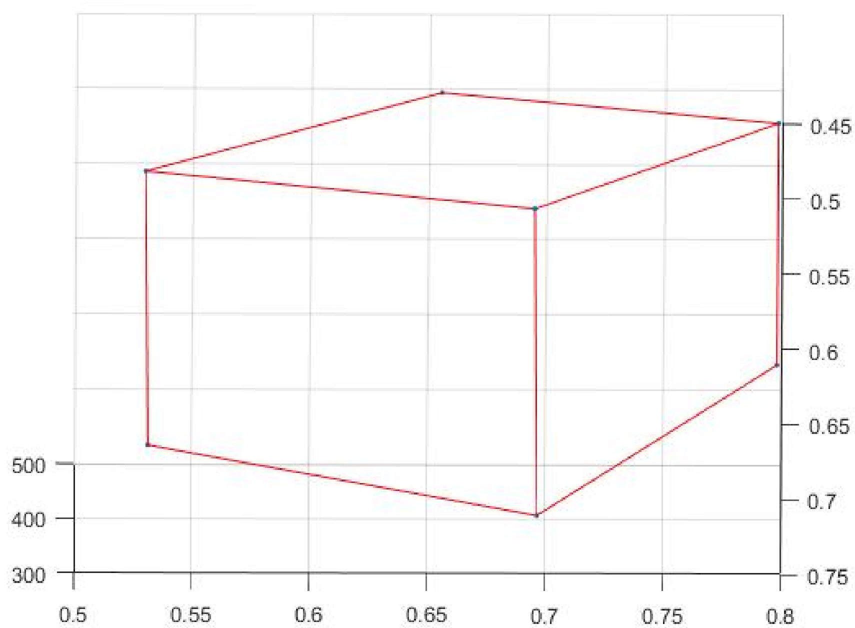

Figure 4: Principle of triangulation.

6 Discussion

6.1 Disparity

We get disparity very less, because when we place these images side by side, we can see the difference in mm for these corresponding points is very small which can be seen in the results.

6.2 3D Coordinates of the points

3D coordinates of these points are also pretty close to the real values. From my setup, the actual depth can be seen pretty close to the calculated values. Approximately the front points were 32-35 cms from the camera and the points at the back were around 42-44 cms from the camera. So we did get a pretty close depth from the stereo images. We can also see that in the figure obtained from MATLAB.

7 MATLAB Code

The code is self explanatory and can be understood with the comments written in the code.

```
%%
%Read the first image and take the 7 visible points as
    pixels.
il=imread('il.jpg');
imshow(il);
[xl,yl]=ginput(7);
left_pixels = [xl, yl];

%Read the second image and take the 7 visible points as
    pixels.
ir=imread('ir.jpg');
imshow(ir);
[xr,yr]=ginput(7);
right_pixels = [xr,yr];

%%
format long g;

%These are the pixel densities as observed in Camera
    Calibration experiment.
k = 840;
l = 840;

```

```
%Convert the pixels to mm using the intrinsic parameters Alpha and Beta for both the images, since, the same camera was used to take these images.
left_mm = left_pixels/k;
right_mm = right_pixels/l;

%Set the focal length, and the observed base line distance between the two cameras.
f=4.15;
b=124.46;

%Calculate the disparity for all the points in the same direction in which the camera was moved.
d=left_mm(:,1) - right_mm(:,1);

%Using the principle of triangulation, calculate the 3D coordinates of the points with respect to camera.
Z = f*b./d;
X = (left_mm(:,1) .* (Z/f))./Z;
Y = (left_mm(:,2) .* (Z/f))./Z;

%Putting these points together
P = [X Y Z];

%Now, we have the 3D points of the all the 7 points and we can plot it in 3D space.
plot3(X,Y,Z,'.');
grid on;
hold on;

%Joining the various points individually.
v1 = P(1,:);
v2 = P(2,:);
v = [v2;v1];
plot3(v(:,1),v(:,2),v(:,3),'r');

v1 = P(2,:);
v2 = P(3,:);
v = [v2;v1];
plot3(v(:,1),v(:,2),v(:,3),'r');

v1 = P(3,:);
v2 = P(4,:);
v = [v2;v1];
plot3(v(:,1),v(:,2),v(:,3),'r');

```

```
v1 = P(4,:);
v2 = P(1,:);
v = [v2;v1];
plot3(v(:,1),v(:,2),v(:,3),'r');

v1 = P(5,:);
v2 = P(6,:);
v = [v2;v1];
plot3(v(:,1),v(:,2),v(:,3),'r');

v1 = P(6,:);
v2 = P(7,:);
v = [v2;v1];
plot3(v(:,1),v(:,2),v(:,3),'r');

v1 = P(1,:);
v2 = P(5,:);
v = [v2;v1];
plot3(v(:,1),v(:,2),v(:,3),'r');

v1 = P(2,:);
v2 = P(6,:);
v = [v2;v1];
plot3(v(:,1),v(:,2),v(:,3),'r');

v1 = P(3,:);
v2 = P(7,:);
v = [v2;v1];
plot3(v(:,1),v(:,2),v(:,3),'r');
```