Visual Perception Lab 1

<< Projective Reconstruction >>

Master in Computer Vision

UNIVERSITE DE BOURGOGNE

Centre Universitaire Condorcet - UB, Le Creusot

20 May 2017

Submitted to: Prof. David Fofi

Submitted by: Mohit Kumar Ahuja

Functions Made/Used:

1. **VP_Project** - Complete Code file.
2. **calculate_2D_point** - Will calculate 2D Points form projection Matrix.
3. **compute_Fundamental_Matrix** – Compute FM using RANSAC.
4. **normalHartley** - Normalisation using normal_Hartley.
5. **funmat7p** – function used in RANSAC.
6. **Normalonetoone** - Normalisation using normal filter.
7. **funmatRANSAC** - Compute FM using RANSAC.
8. **Calculate_Proj** - Will compute projection matrix.
9. **Calculate_Proj_from_Fund** - Will compute projection matrix from FM.
10. **plot2D** - Used for plotting.
11. **plot3D** - Used for plotting.
12. **Plot_Image** - Used for plotting.

Task:

1. **Simulate a simple 3D scene and stereovision system, assuming that all the parameters are known (intrinsic and extrinsic parameters). Compute the resulting images (projections of the scene onto the two camera planes).**

Solution: I used an internal Point Cloud of matlab ('teapot.ply') for 3D scene. And as shown in figure there are two camera's shown, Red one is the first camera and the orange one is the second camera. In the starting of the code, We assigned {x,y,z} values to both the camera's. As shown below,

```
Camera_1_Coordinate = [0 0 0];
Camera_2_Coordinate = [7 0 0];
```

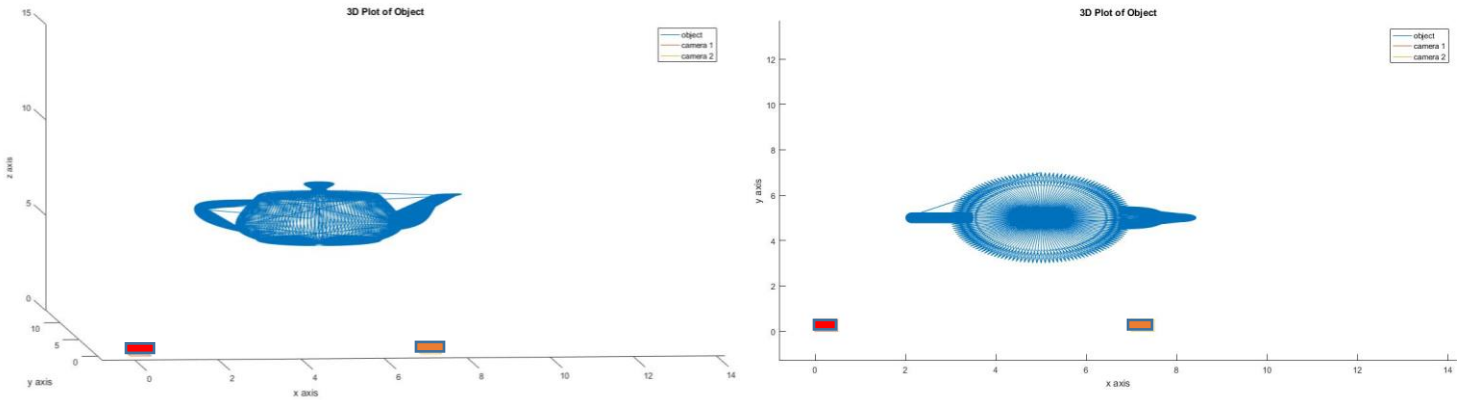

Figure 1: 3D Scene and stereovision system

```
RT_Matrix = [Rotation_Matrix Translation_Matrix];
Proj_Geo_1 = Intrinsic_Parameters*e1*RT_Matrix;
RT_Matrix2 = [Rotation_Matrix2 Translation_Matrix2];
Proj_Geo_2 = Intrinsic_Parameters2*e2*RT_Matrix2;
P1 = Proj_Geo_1*a;
P2 = Proj_Geo_2*a;
```

In code, RT Matrices represents the external parameter matrices for both camera 1 and camera 2. The Intrinsic_Parameters matrices contains the intrinsic parameters of both camera's which is then further used to calculate the projective matrices of both camera's. Then, P1 and P2 are calculated in "calculate_2D_point" function. In which we calculate the 2D {x,y} coordinates of 3D object. The resulting image is being plotted in figure 2 and figure 3.

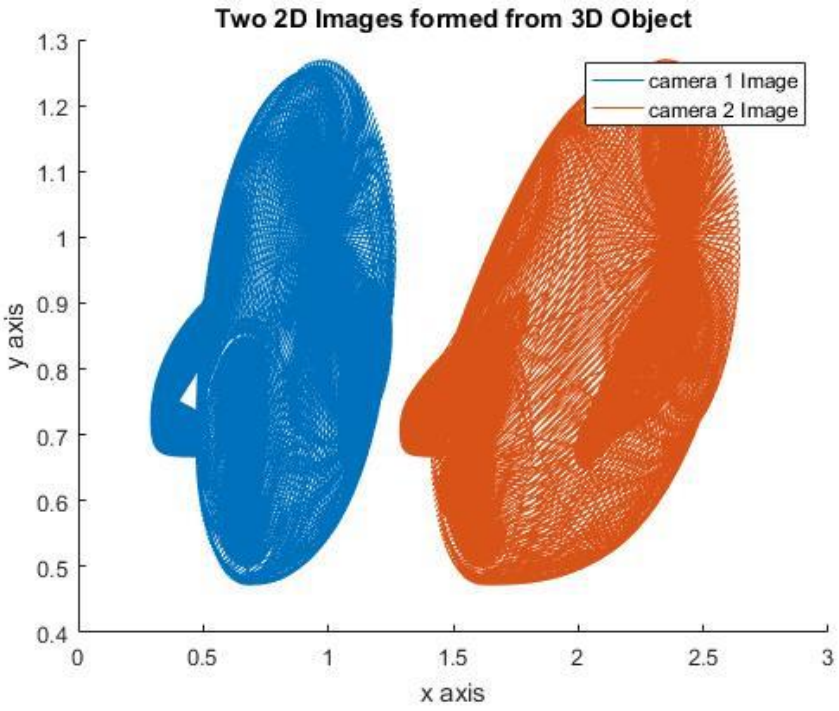

Figure 2: 2D projection of 3D Object

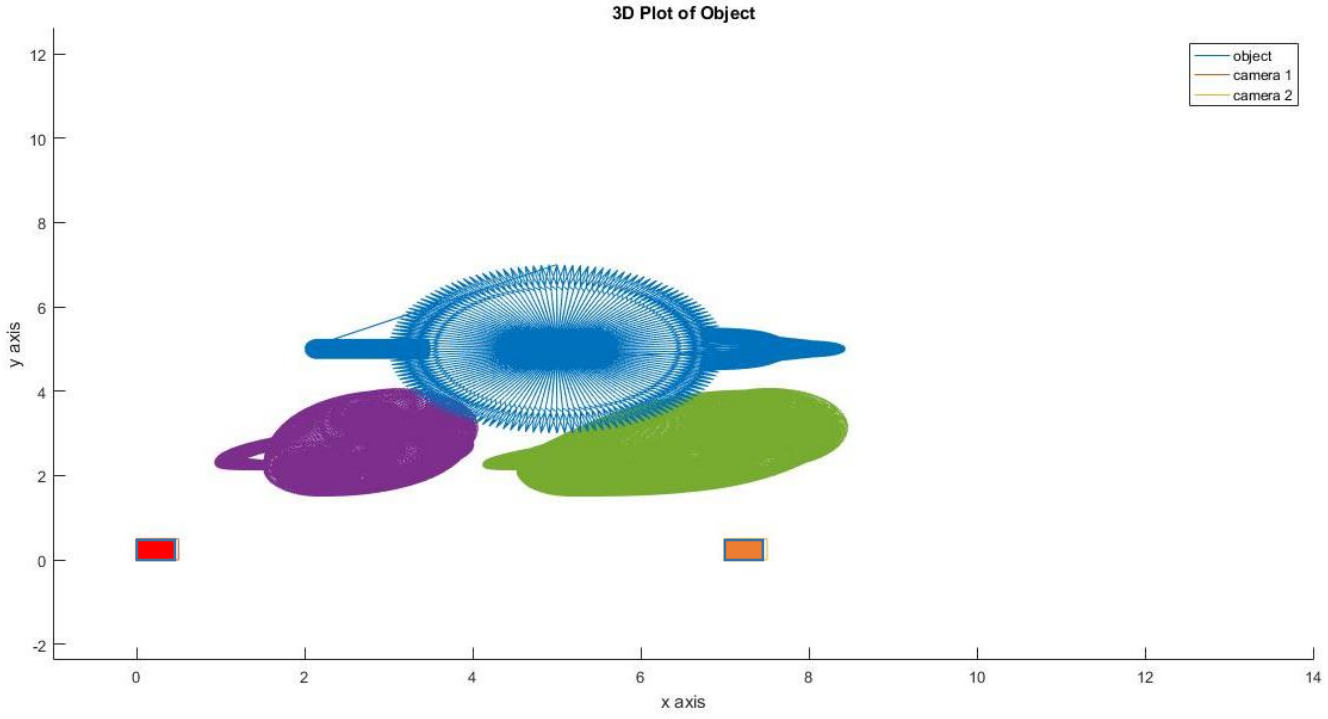

Figure 2: 2D projection of 3D Object in 3D plot

2. **Compute the fundamental matrix from the known parameters.**

Solution: As the Internal and External parameters are known to us, we can calculate the Essential matrix and using essential matrix we can calculate the Fundamental matrix. So, the formula used to calculate essential matrix is:

$$E = R[t]x$$

And using essential matrix, we can calculate the Fundamental Matrix;

$$E = K'^{T} * F * K$$

And the resulting Fundamental matrix was;

Fundamental_Matrix =

$$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 7 \\ 0 & -7 & 0 \end{matrix}$$

3. **Estimate the fundamental matrix from the two images.**

Solution: Now we have two 2D images and we will use that data to estimate our Fundamental matrix. I choose RANSAC method for estimation. In RANSAC, the M matrix I took is of 4*999 where first two rows are the x-y-coordinates of first image and third forth rows are collinear x-y-coordinates of second image.

So, using Salvi's toolbox, the Estimated Fundamental Matrix was;

Fundamental Matrix RANSAC=

$$\begin{matrix} 0.0000 & -0.0001 & -0.0000 \\ -0.0001 & 0.0001 & 1.0000 \\ 0.0000 & -0.9999 & -0.0000 \end{matrix}$$

Which will be then multiplied with a scale factor of 7 and after multiplication, we will round off the values and the final Estimated Fundamental Matrix we got is;

Fundamental Matrix RANSAC=

0	0	0
0	0	7
0	-7	0

4. Compute the 3D scene from the canonical representation.

Solution: Now for 3D scene, I calculated the Projection Matrix for Camera 1 and camera 2 using function "Calculate_Proj" & "Calculate_Proj_from Fund".

After which I calculated 3D points from 2D points using a prebuild function "triangulate". Syntax of which is;

```
Three_D_Points = triangulate(Image_1(1:2,:)',Image_2(1:2,:)',Proj_1,Proj_2);
```

The Result of plotting the new calculated 3D object is shown in figure 4 and figure 5.

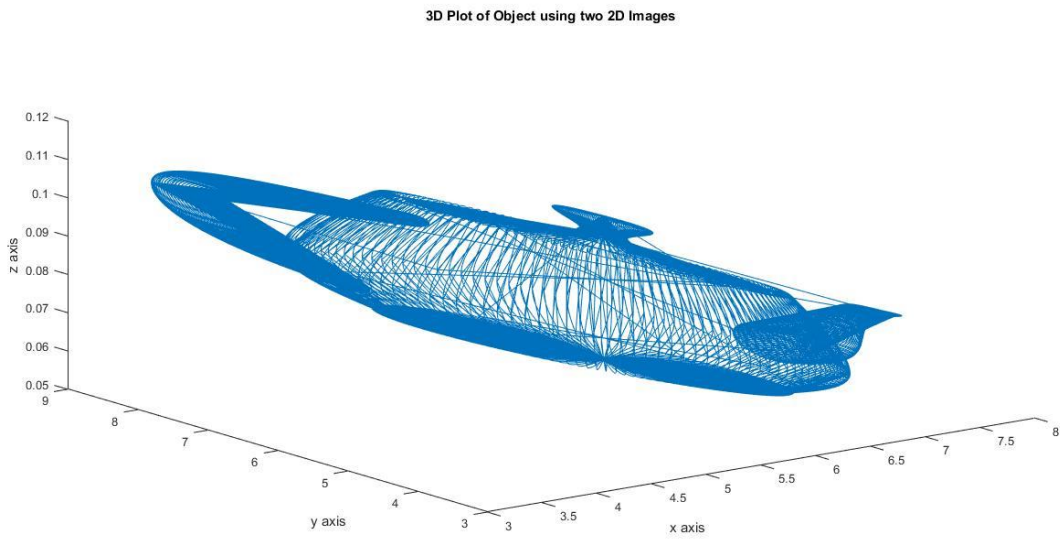

Figure 4: 3D Reconstruction of 3D object using two 2D Images

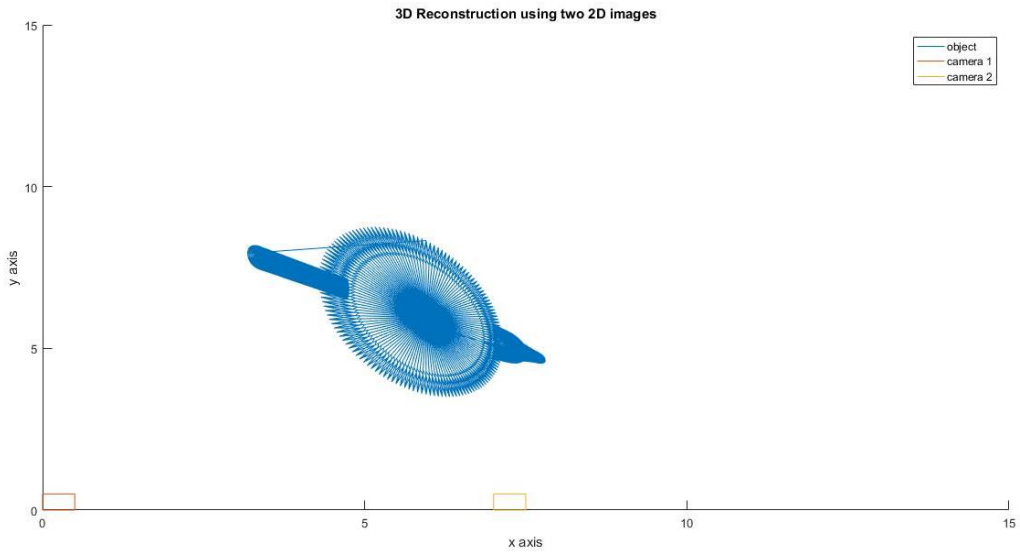

Figure 5: 3D Reconstruction of 3D object using two 2D Images

5. Compute the residual error (2D error).

Solution: For Residual Error, I first computed 2D images of my new obtained 3D structure using the same method as used in question 3 and from those points, the residual error was computed to be very less. Near to zero.

As we can see in figure 6, the images are similar to figure 2;

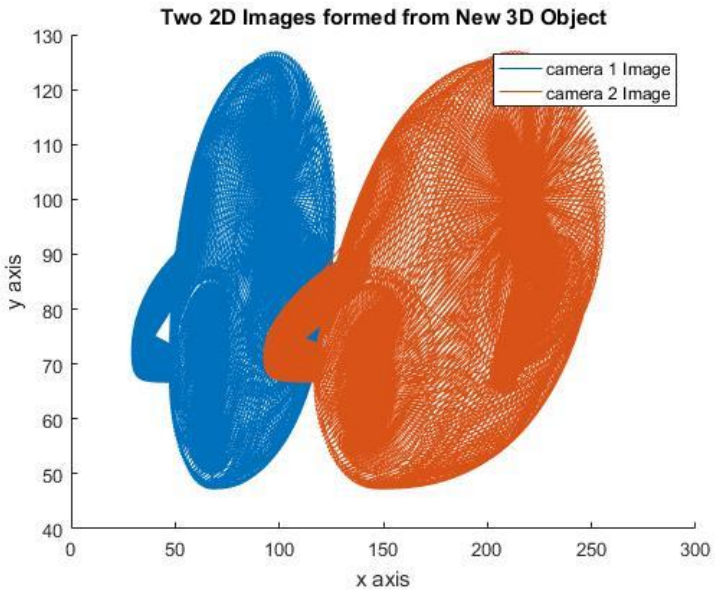

Figure 6: 2D Reconstruction of new 3D object

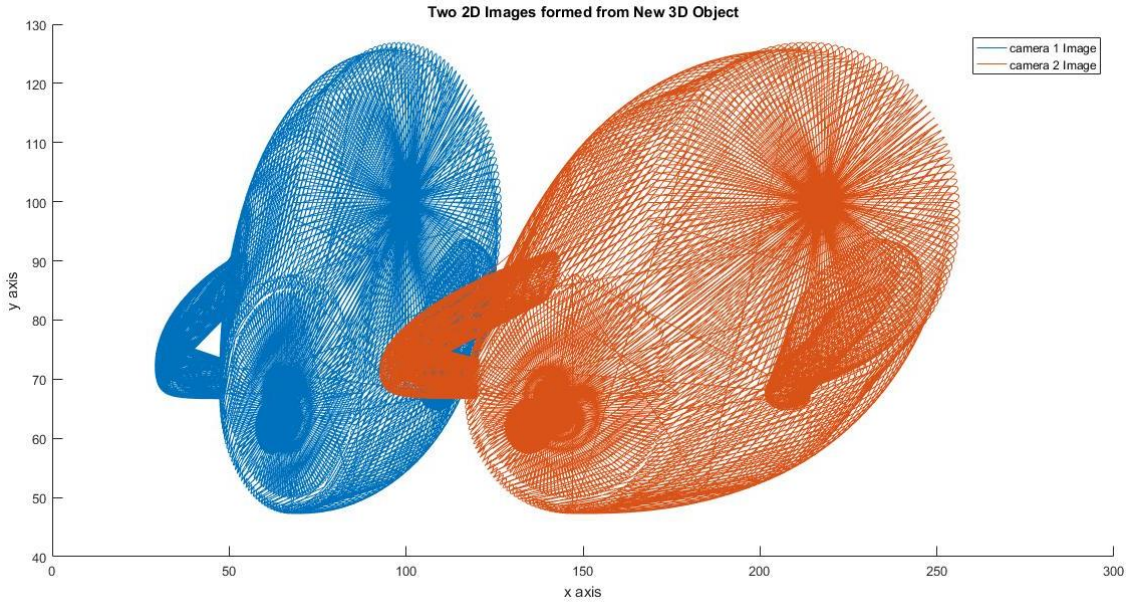

Figure 7: 2D Reconstruction of new 3D object

7. Compare the estimated 3D scene with the initial one.

Solution: There was a slight difference between the reconstructed one to the original 3D scene, which can be clearly seen in the figure 8. But it is not a major change.

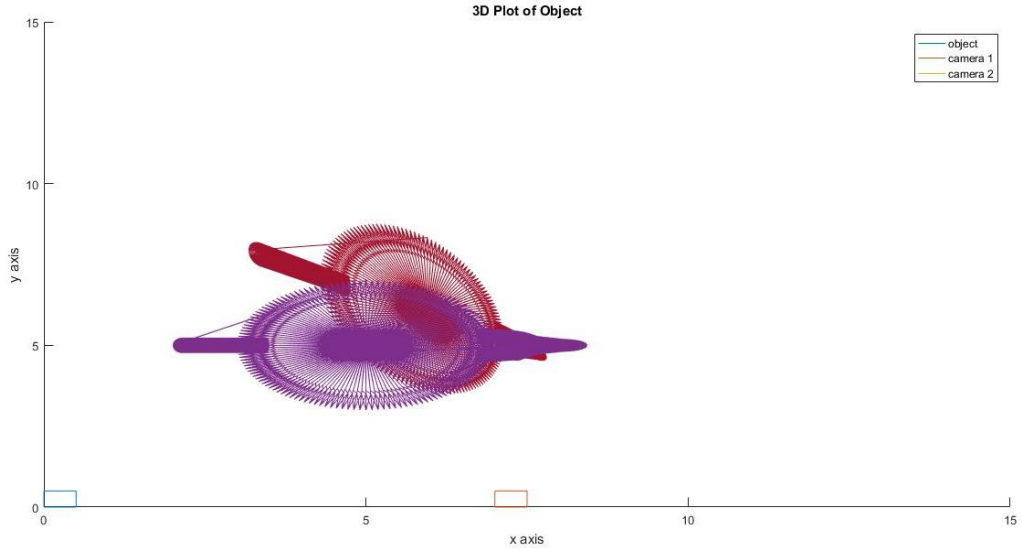

Figure 8: Comparison between original 3D object with new 3D object