Université Jean Monnet

Course: Computer Vision

Final Project

Image reconstruction using Kinect sensors

Arunava Maulik, Jorge Chong

March 31, 2017

1 Objective

We want to reconstruct a 3D scene from this two images

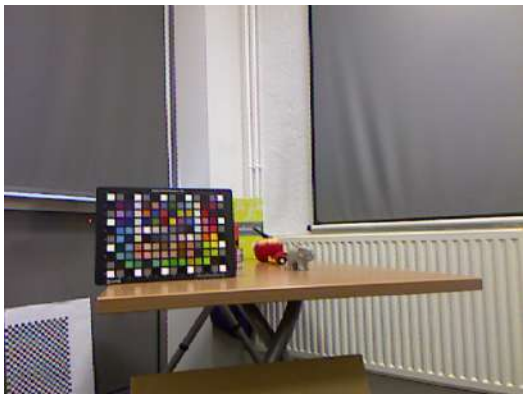


Figure 1: Left camera

Figure 2: Right camera

2 Process

1. Calibration of the RGB camera parameters in the left and right setup
2. Calibration of the Stereo setup
3. Transformations
   (a) From pixel frame to IR camera world coordinates reference
   (b) From IR camera reference to the RGB camera reference. We have a point cloud in 3D space.
   (c) To pick the color of that point in space we go from RGB 3D to pixel frame in RGB camera and use that to find the color in the RGB image
   (d) Do the same for the other camera
   (e) Finally, transform from one camera reference to the other and merge the points

3 Calibration

We used the pictures of the checkerboard that we took to calibrate the RGB cameras using the calibration tool for matlab developed by the Caltech University [1]. The final results are shown in the image. The final parameters are:

1. Focal length of right RGB camera
2. Principal Point of right RGB camera
3. Focal length of left RGB camera
4. Principal Point of left RGB camera
5. Rotation vector from left to right camera
6. Translation vector form left to right camera

```
Intrinsic parameters of right camera:

Focal Length:          fc_right = [ 594.82345   596.71940 ] ± [ 33.54462   42.40504 ]
Principal point:       cc_right = [ 313.32119   204.22972 ] ± [ 48.81210   40.46981 ]
Skew:             alpha_c_right = [ 0.00000 ] ± [ 0.00000  ]
                                  => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:            kc_right = [ 0.36731   -0.23886   -0.06112   -0.00334  0.00000 ]
                                  ± [ 0.34684   2.78035   0.04327   0.05616  0.00000 ]

Intrinsic parameters of left camera:

Focal Length:          fc_left = [ 515.98373   506.36716 ] ± [ 32.08629   29.01246 ]
Principal point:       cc_left = [ 356.82633   223.83975 ] ± [ 33.34426   21.78764 ]
Skew:             alpha_c_left = [ 0.00000 ] ± [ 0.00000  ]
                                  => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:            kc_left = [ 0.54143   -1.02578   -0.02251   0.07919  0.00000 ]
                                  ± [ 0.25389   0.90103   0.02223   0.05348  0.00000 ]

Rotation vector:              om = [ -0.08170   0.75781  -0.07675 ]
Translation vector:            T = [ -711.50008   61.41676  500.67368 ]
```

4 Transformations

We project each point in 2D to 3D for each IR camera using this transformation:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} \frac{(x-c_x)d}{f_x} \\ \frac{(y-c_y)d}{f_y} \\ d \end{pmatrix}$$

The next step is to do a transformation from the IR 3d camera reference to the 3d RGB camera reference. For this we use the kinect extrinsic parameters R and T, and transform each point as R * Point3D + T.

Then we reproject to the pixel frame of the RGB camera to find out the color of each point in 3D.

We end up with two point clouds for each RGB camera in left and right. The final step is to merge the two point clouds. We transform the points in 3D from the Left camera to the Right camera reference.

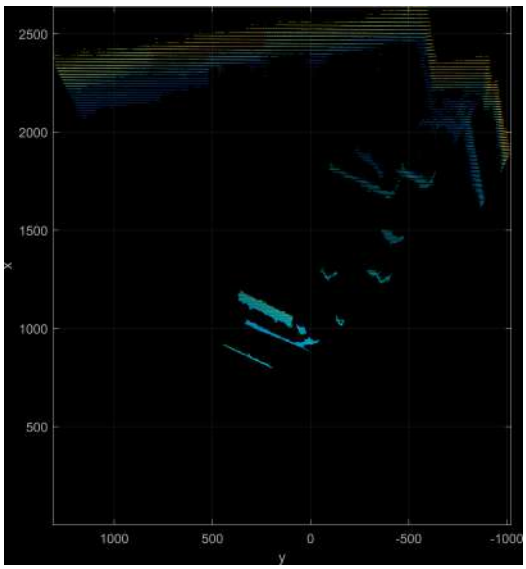


Figure 3: Right camera

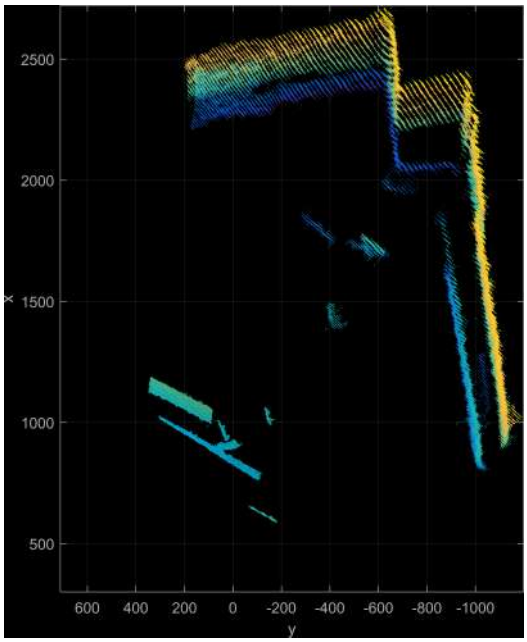


Figure 4: Left to Right camera

In order to merge the two point clouds we transformed the left point cloud using the rotation and translation vector from the stereo calibration. The problem we found is that the translation vector is not accurate and we had to try to align both point clouds manually.

After manual adjustment we found the translation vector [-711.5001 -23.5832 300.67370]

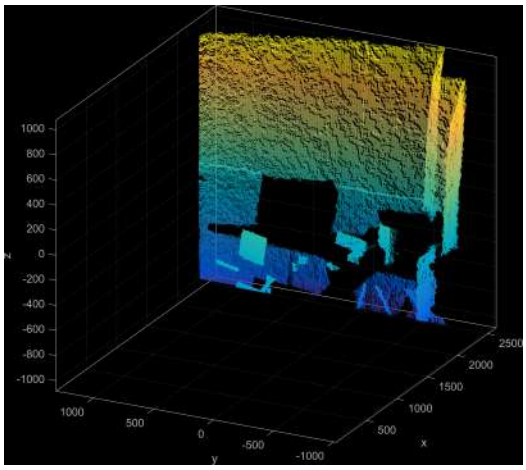


Figure 5: Right camera

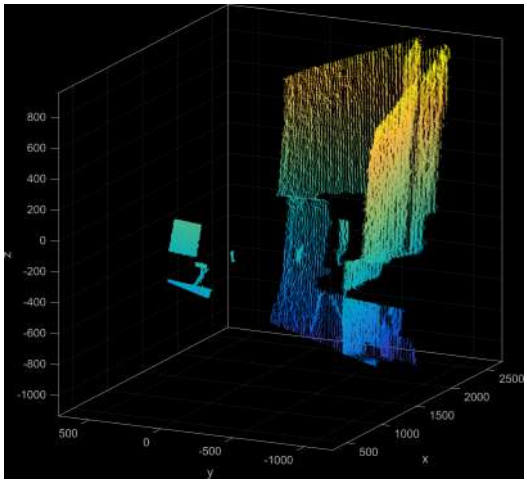
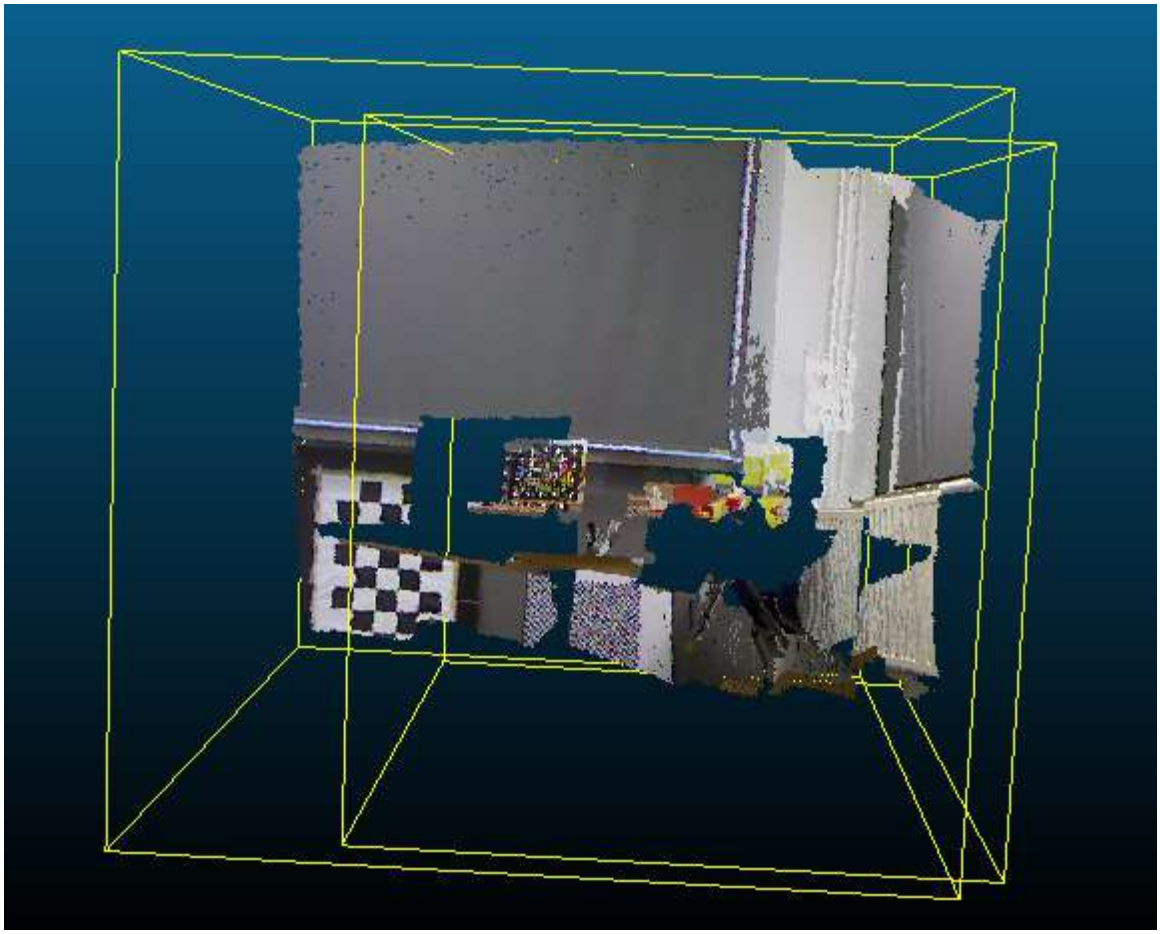
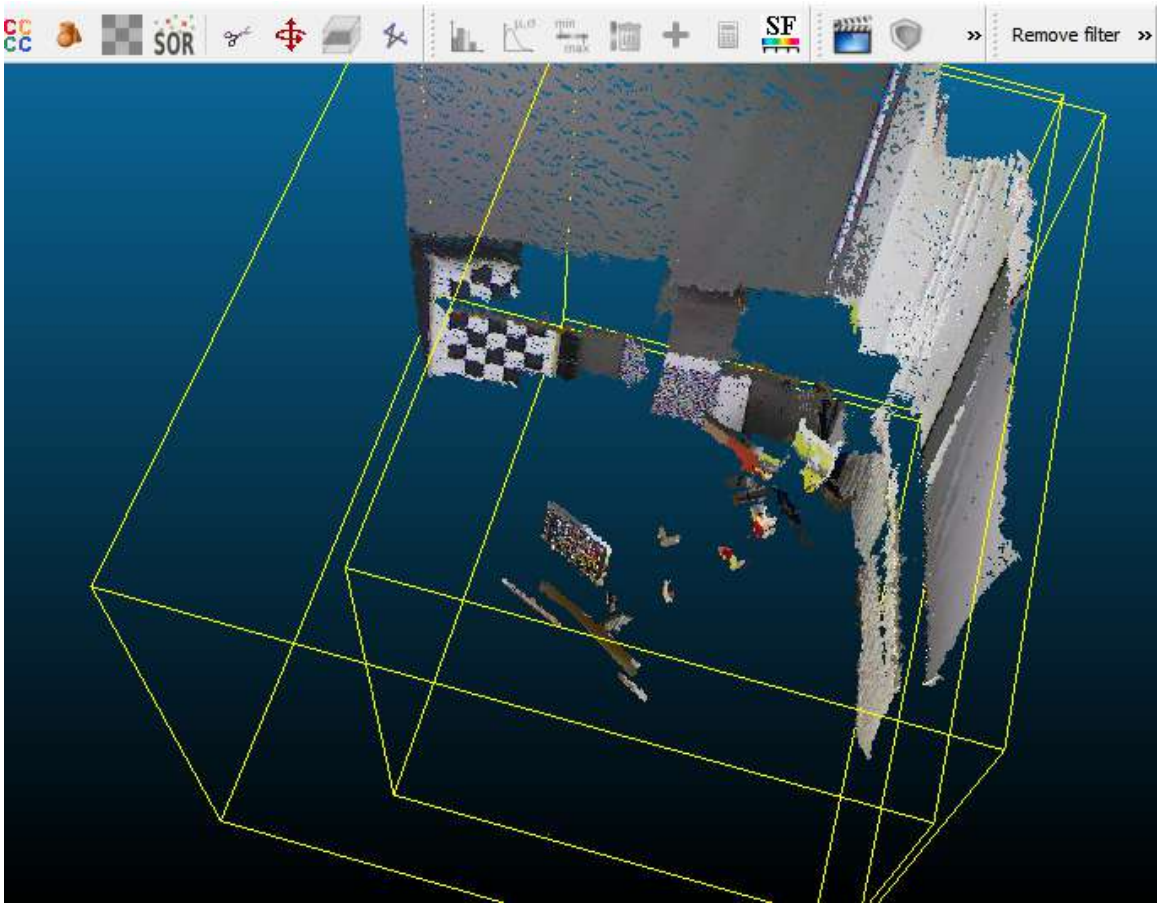
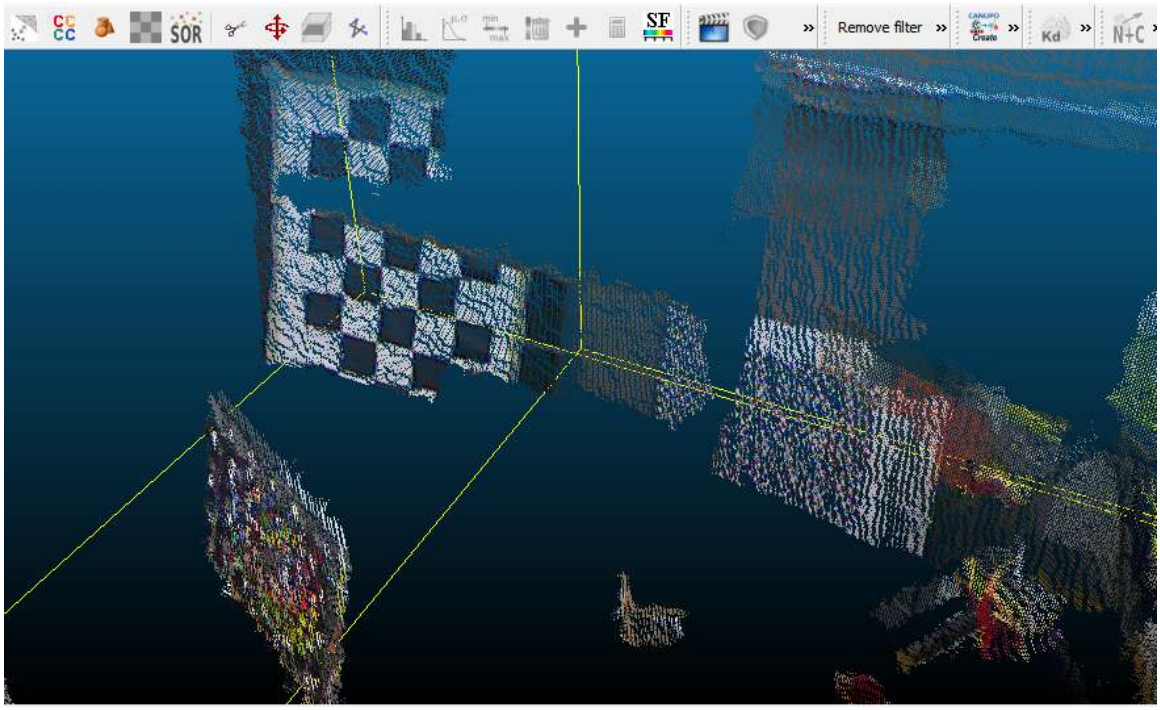
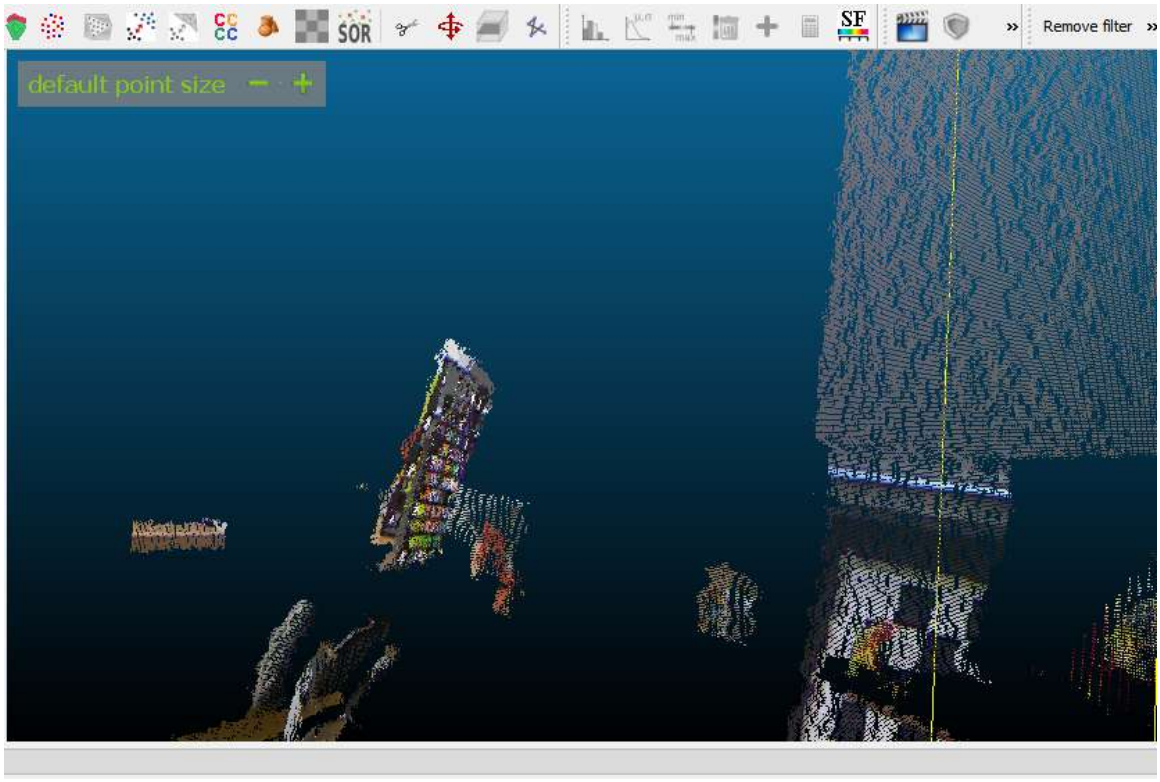


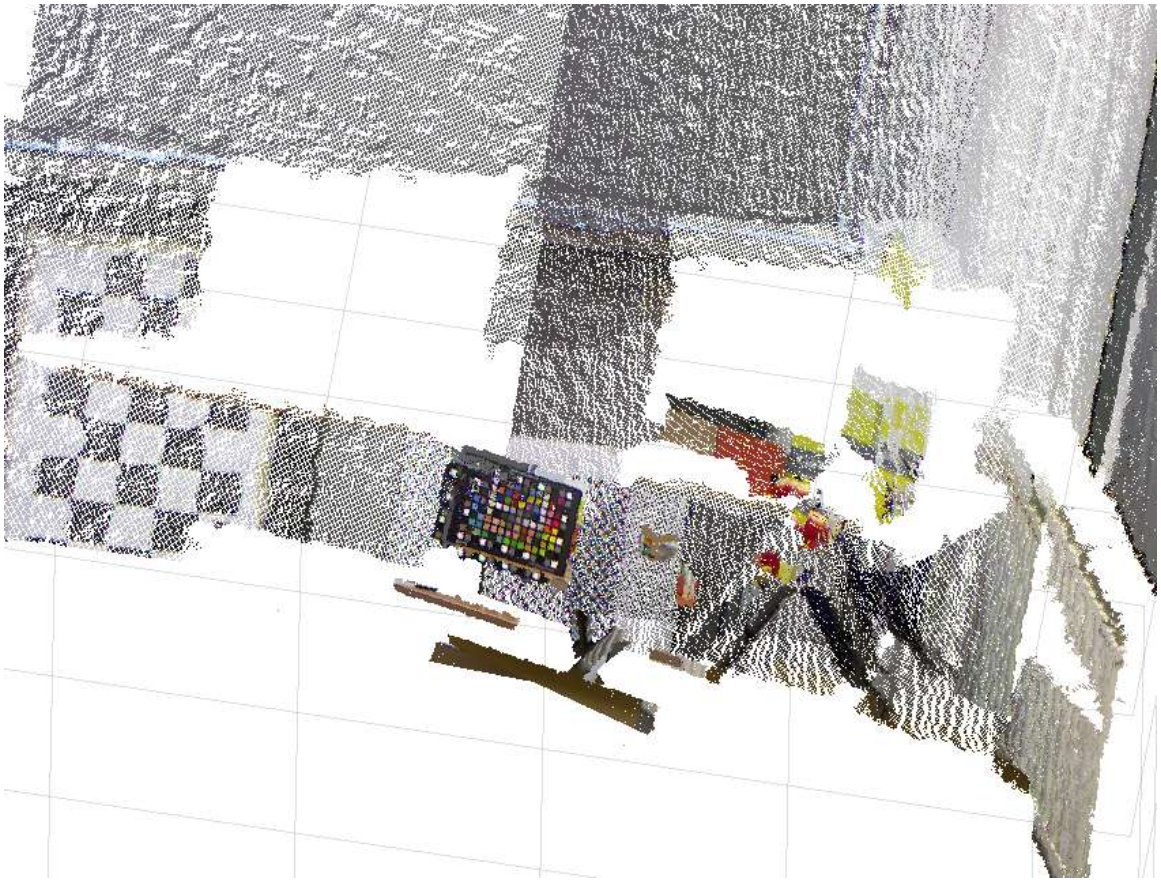
Figure 6: Left to Right camera

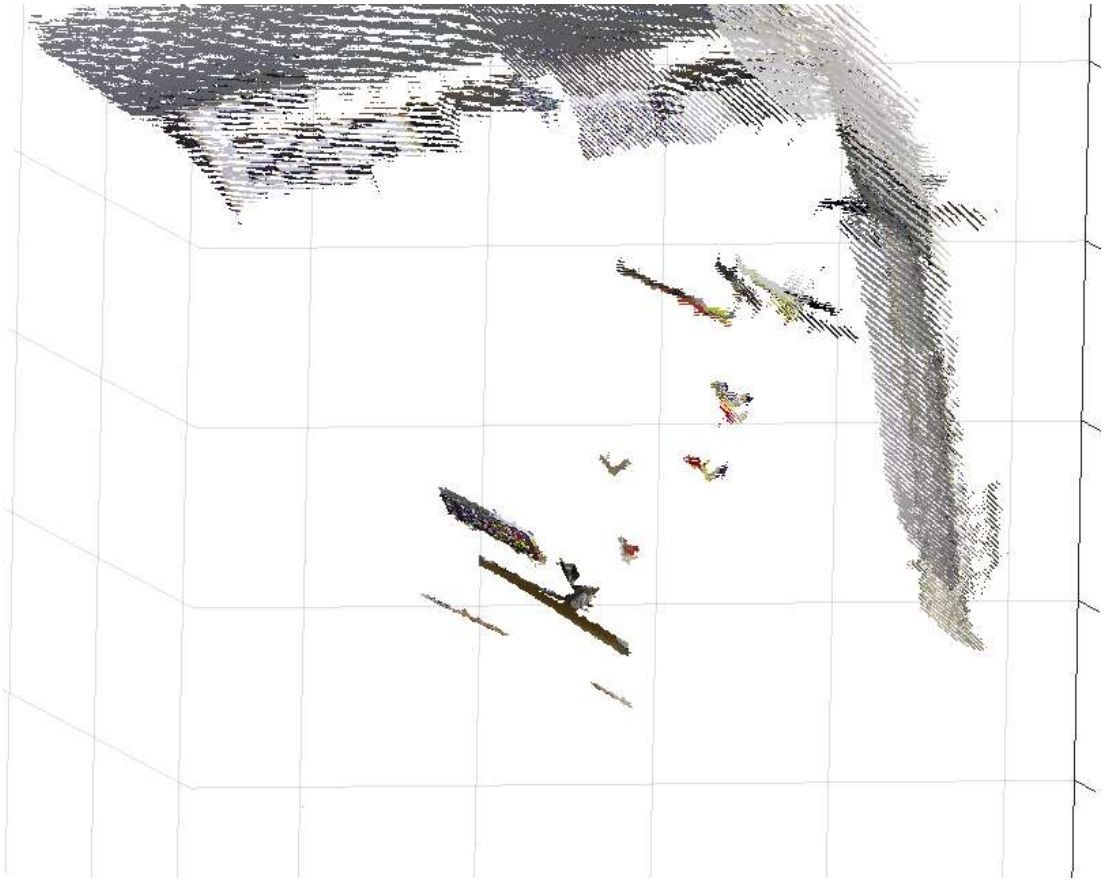


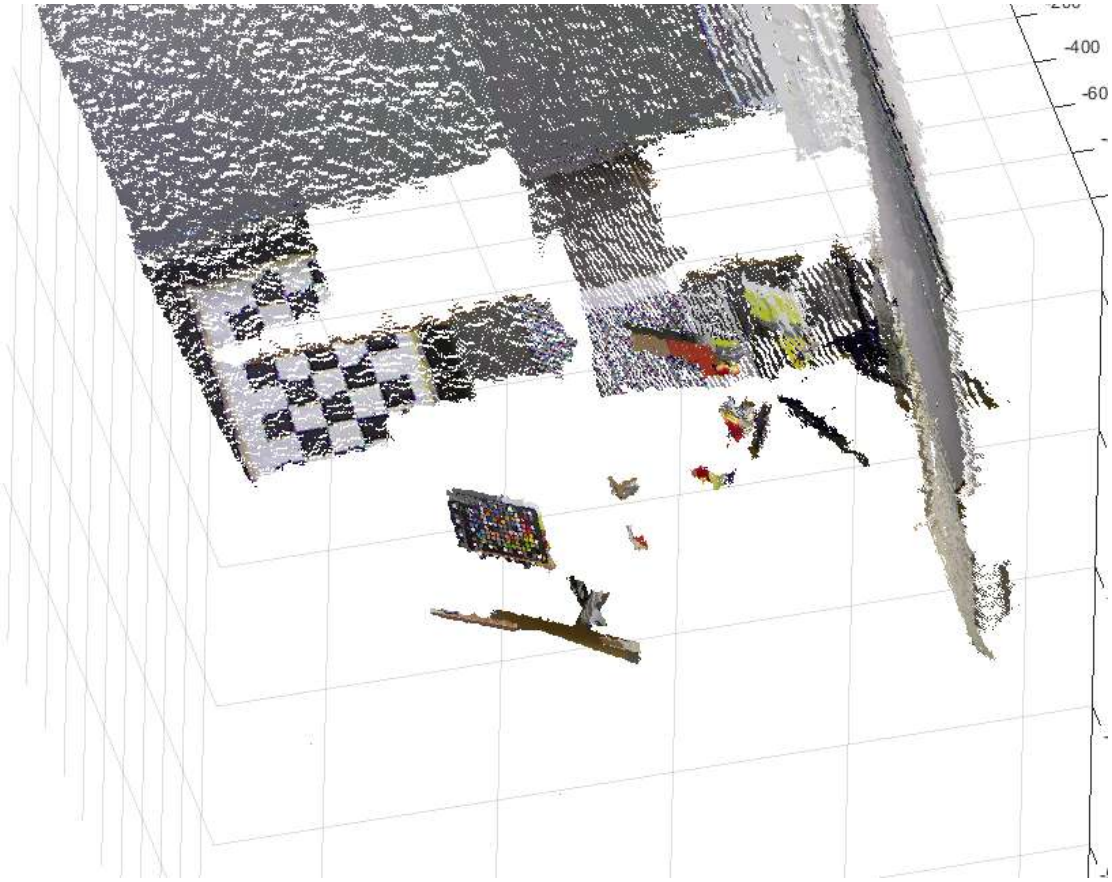

SOR
SF
Remove filter


SOR
SF
Remove filter
Kd
N+C


SOR
SF
Remove filter
default point size






-400
-60

5 Final Results

We were able to reconstruct the point clouds but some manual adjustment was necessary in the translation vector of the stereo calibration. The problem is we need more pictures to get a better calibration. The complete program is in D3_6.m

References

[1] Camera Calibration Toolbox for Matlab, http://www.vision.caltech.edu/bouguetj/calib_doc/