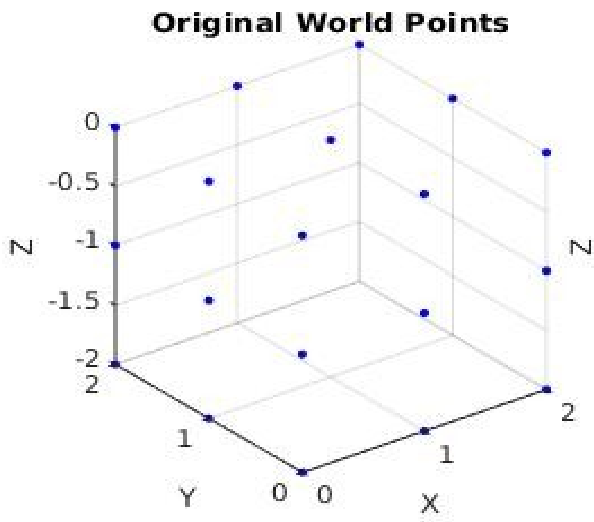
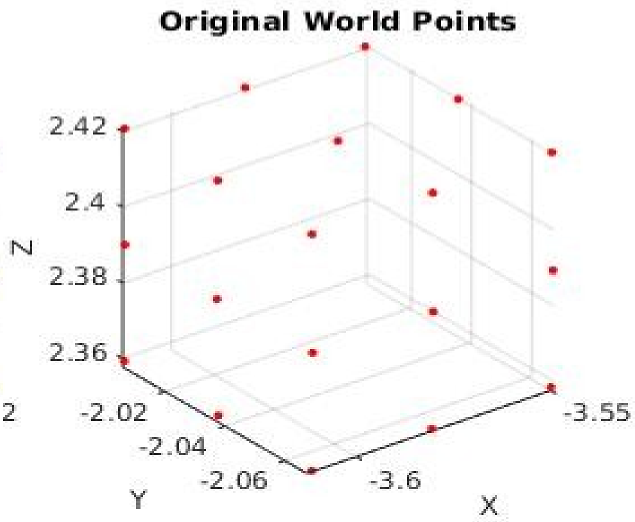
EE417 Computer Vision
Hakan Buğra Erentuğ
23647 Post Report #9





The resultant is placed below as asked in post lab description in lab document.

There is a hard theory behind the how we estimate. But basically we have 8 different equation that corresponds each point and with transform and rotation matrix. There are 4 rotation and transformation matrix totally. By the help of them we can write euclidean transform which help us to change point coordinates to one camera to another. The first thing we need to do is the finding scale factor A. Scale factor vector is an array with 20 element with 19 point an one Y. Due to homogeous coordinate system the last rows of matrix does not change the equations. Then we have to create M matrix by putting diagonals 3x3 x1Rx2 matrix. X1Rx2 basically the coeffcient of lambda 1 in the euclidian transform formula. After we form the MA matrix with the size of 57,20 (57 comes from 19*3 because for each point we have 3x3 matrix) and 20 is coming from scale factor array. Then, we need to solve the equaiton by use of eigen vectors. M transpose * M is the way of the solution. If we take the singular decomposition of the MTM matrix basically we can conclude the eigen vectors.

```
[Uest, Sest ,Vest] = svd(MA'*MA);
lambda = Vest(:,end);
```

and last colomn of the V vector is associated with smallest eigenvalue of MTM.

The last thing we need to do form a reconstructed coordinates by created 19x4 matrix. 4 is coming from x, y z coordinates with 1 due to homogeneity. Of course we cannot find the scaled version of original object. The mean squared error is basically, calculated as 0.56789877, by centralized their means and re-scaling their optimum values.