EE 417 Computer Vision

Berkant Deniz Aktaş

19515

Post-Laboratory Report

October 25, 2018

Sabancı University

Faculty of Engineering and Natural Sciences

Computer Science and Engineering

In Lab

In this lab, we constructed 3D world from given two input images with knowing rotation and translation (from previous lab) and our camera was already calibrated. As we known from lectures, we can recover 3D World up to a scale because we do now know the theta parameters of translation vector.

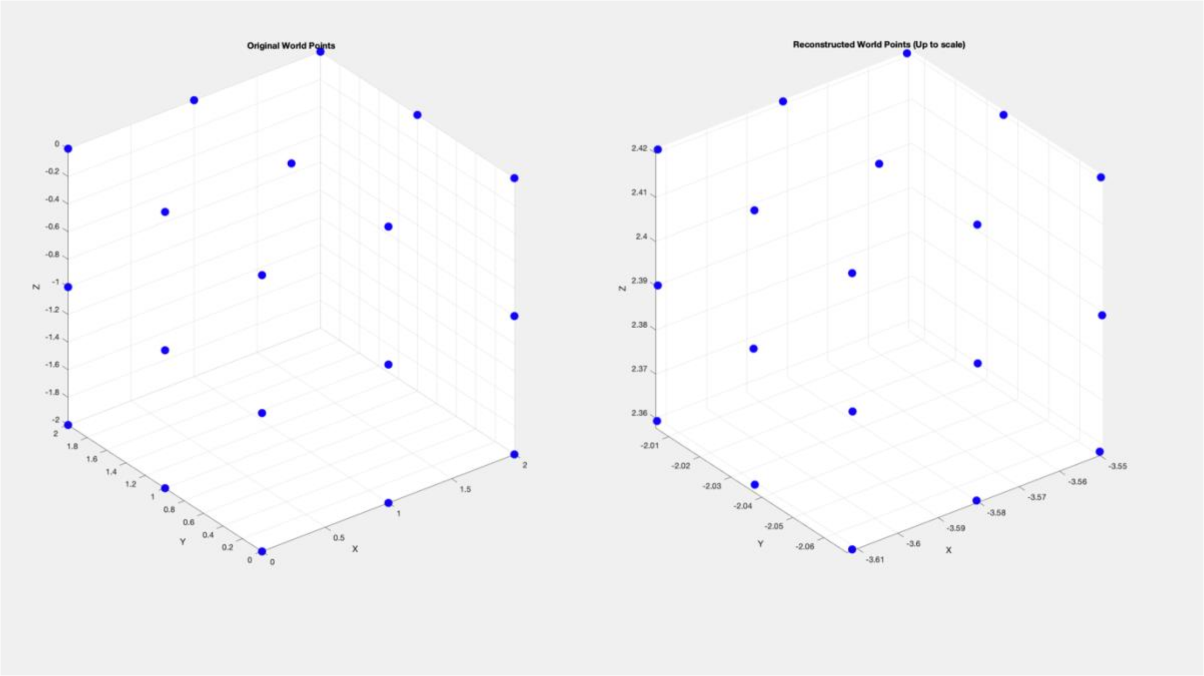

Figure 1. Reconstructed with wrong scale

As its shown in the Figure 1 the scale is not matching the original Word points are much much bigger than reconstructed World.

Post lab

There are different ways to achieve scale and reconstruct 3D World. Unfortunately, I could not write my own code. Rather than that I did a research and found related papers.

My first idea was using SIFT to compute the matching points and then reconstruct the world using some tool. Unfortunately, I could not also implemented it.

https://pdfs.semanticscholar.org/398a/05ba68dfc145164fb932dd4c251896d71174.pdf

This paper written by Carlo Tomasi, emphasized the triangulation (reconstruction) of the world. First it used Longuet Higgins to compute matching points then it reconstructs based on 4 possiblities.

I also found a paper from MIT, written by Horn called Closed-form solution of absolute orientation using unit quaternions

http://people.csail.mit.edu/bkph/papers/Absolute_Orientation.pdf

And I found paper’s implementation by wise guys from ETH Zurich CV lab.

https://www.mathworks.com/matlabcentral/fileexchange/22422-absolute-orientation

I tried to modify it and I could not achieve robust solutions.

There are more solutions exist but I due to time conflicts I could not spend lots of time in this post lab

In lab code

```
%% Lab#9 Assignment starts here.
matrix = zeros(NPTS*3,NPTS+1);

for i =1:1:NPTS
tempmatrix = makeskew(p2(:,i)) * R * p1(:,i) ;
matrix(i*3-2,i) = tempmatrix(1,1);
matrix(i*3-1,i) =tempmatrix(2,1);
matrix(i*3,i) =tempmatrix(3,1);
end

for i=1:1:NPTS
tempmatrix = makeskew(p2(:,i)) * T ;
```

```
matrix(i*3-2,NPTS+1) = tempmatrix(1,1);
matrix(i*3-1,NPTS+1) = tempmatrix(2,1);
matrix(i*3,NPTS+1) = tempmatrix(3,1);
end
[U S V] = svd(transpose(matrix)*matrix);
lambdas = V(:,20);


matrix2 = zeros(4,19);
for i=1:1:NPTS
matrix2(1,i) = p1(1,i)*lambdas(i);
matrix2(2,i) = p1(2,i)*lambdas(i);
matrix2(3,i) = p1(3,i)*lambdas(i);
matrix2(4,i) = 1;
end

matrix2 = inv(Hc1)*matrix2;
```