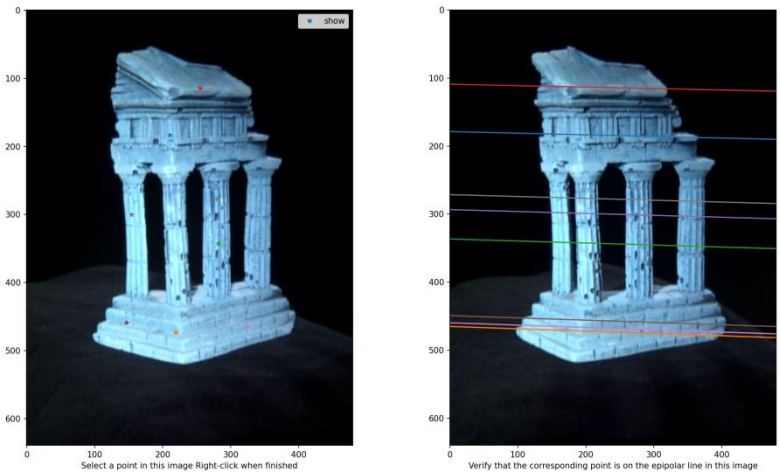
CMPT 762 X100, Fall 2024, Computer Vision

Project 3: 3D Reconstruction

Name: Wenhe Wang
Student Number: 301586596
Email: wwa118@sfu.ca

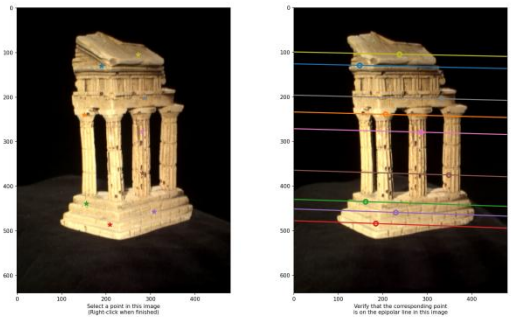
3.1.1 Implement the eight point algorithm

```
(venv) PS C:\Users\wenhe\Downloads\project3_package> python .\python\eightpoint_test.py
Normalization factor M: 1440.0
Computed Fundamental Matrix F:
[[ 1.77324730e-09 -1.70876560e-08 -8.77582235e-06]
 [-6.63558810e-08 -4.05296638e-10  4.95571221e-04]
 [ 1.69307908e-05 -4.75954392e-04 -2.06190050e-03]]
```



3.1.2 Find epipolar correspondences

The similarity metric here is using Euclidean distance on the Grey scaled pixel values of different windows. Because we are trying to do the square to the error here, so I converted my images to Grey color to prevent the loss to become too huge for RGB images. Obviously, if we pick any points such as in the poles or arbitrary background “black points”, the mismatching will happen more often, as all background point values are 0, so our algorithm can’t tell the difference and will just pick a arbitrary background black point.



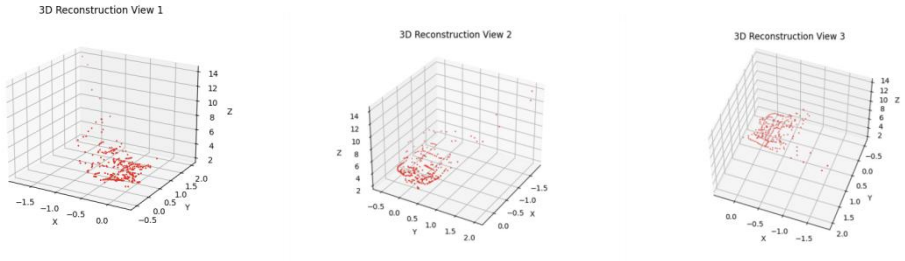
3.1.3 Write a function to compute the essential matrix

```
(venv) PS C:\Users\wenhe\Downloads\project3_package> python .\python\essentialMatrix_test.py
Computed Essential Matrix E:
[[ 0.00409907 -0.03964299 -0.01894139]
 [-0.15394421 -0.00094368  0.72542881]
 [ 0.00165055 -0.73429419 -0.00084402]]
Scaled Essential Matrix E:
[[ 1.97678776e-09 -1.91179553e-08 -1.31537447e-05]
 [-7.42400694e-08 -4.55093008e-10  5.03770010e-04]
 [ 1.14621534e-06 -5.09926521e-04 -8.44017591e-04]]
```

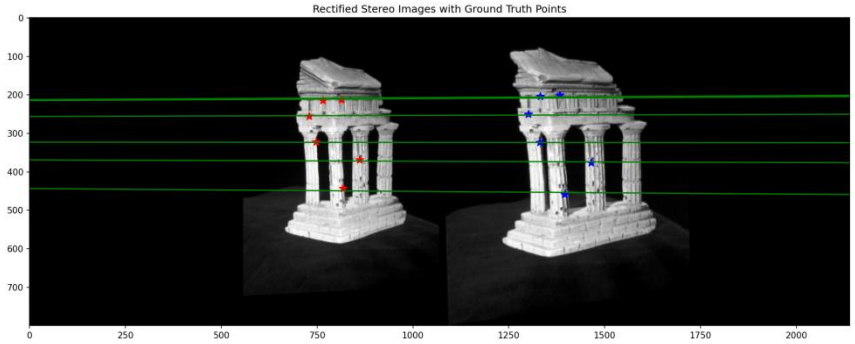
3.1.4 Implement triangulation and 3.1.5 Write a test script that uses templeCoords

As can be seen, there are 4 extrinsic matrix to choose after we pass the essential matrix to the camera2 function. The correct one is the fourth matrix as it has 288 correspondences. As for the projection errors, I got 0.5593 for pts1 and 0.5650 for pts2.

```
(venv) PS C:\Users\wenhe\Downloads\project3_package> python .\python\testTempCoords.py
Computing correspondences using epipolarCorrespondence...
Computed Essential Matrix E:
[[ 0.00409907 -0.03964299 -0.01894139]
 [-0.15394421 -0.00094368  0.72542881]
 [ 0.00165055 -0.73429419 -0.00084402]]
M2 candidate 0: Number of points with positive depth: 0
M2 candidate 1: Number of points with positive depth: 0
M2 candidate 2: Number of points with positive depth: 0
M2 candidate 3: Number of points with positive depth: 288
Computing Reprojection Errors...
Mean Reprojection Error in Image 1: 0.5593 pixels
Mean Reprojection Error in Image 2: 0.5650 pixels
Saved 3D reconstruction image: results\3D_Reconstruction_View_1.png
Saved 3D reconstruction image: results\3D_Reconstruction_View_2.png
Saved 3D reconstruction image: results\3D_Reconstruction_View_3.png
Saved extrinsic parameters to 'results\extrinsics.npy'.
```

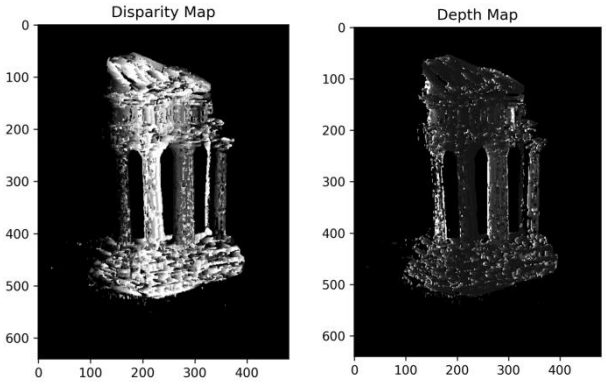


3.2.1 Image rectification

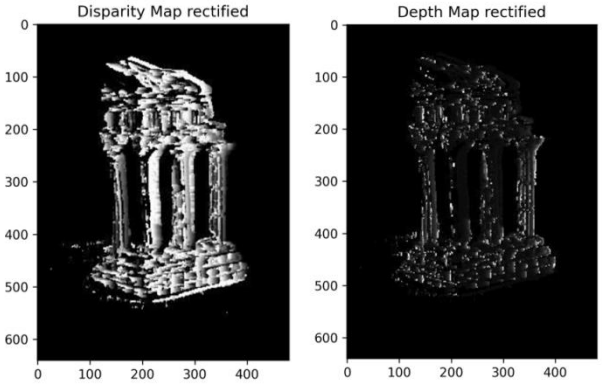


3.2.2 Dense window matching to find per pixel density and 3.2.3 Depth map

Disparity map(left) and depth map(right) before rectification:



Disparity map(left) and depth map(right) after rectification:



3.3.1 Estimate camera matrix P

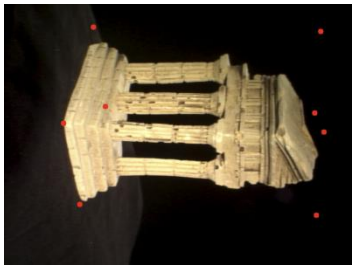
```
(venv) PS C:\Users\wenhe\Downloads\project3_package\python> python .\testPose.py
Reprojected Error with clean 2D points is 0.0000
Pose Error with clean 2D points is 0.0000
------------------------------
Reprojected Error with noisy 2D points is 2.9601
Pose Error with noisy 2D points is 0.5907
```

3.3.2 Estimate intrinsic/extrinsic parameters

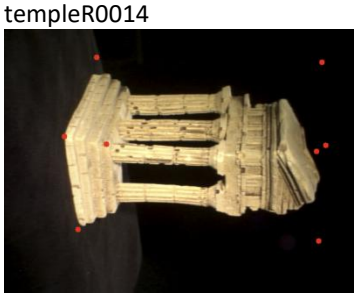
```
PS C:\Users\wenhe\Downloads\CMPT762-ComputerVision\project3_package\python> python testkRt.py
Intrinsic Error with clean 2D points is 0.0000
Rotation Error with clean 2D points is 0.0000
Translation Error with clean 2D points is 0.0000
------------------------------
Intrinsic Error with noisy 2D points is 0.7965
Rotation Error with noisy 2D points is 0.2024
Translation Error with noisy 2D points is 1.9136
```

3.4.1

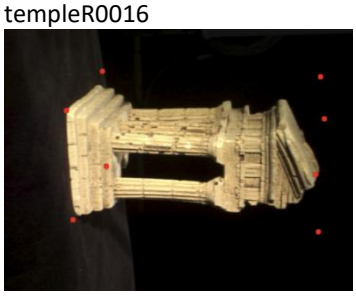
templeR0013



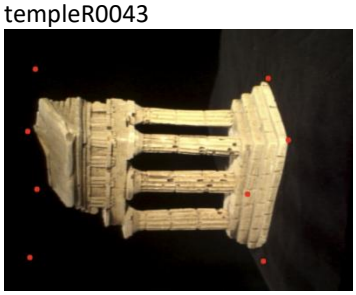
templeR0014



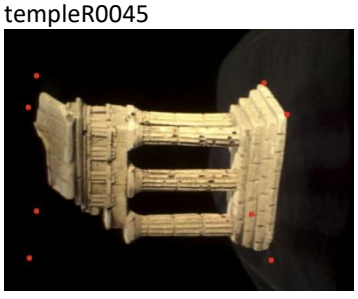
templeR0016



templeR0043



templeR0045



3.4.2

I have some issues with this visualization plot, but the depth information is preserved well

3.4.3