

2023 Fall 3DCV - hw2 Report

生醫電資所 碩一 r12945040 郭思言

Problem 1: 2D-3D Matching

Q1-1 For each validation image, compute its camera pose with respect to world coordinate. Find the 2D-3D correspondence by descriptor matching, and solve the camera pose. Implement at least one kind of algorithm that solves a PnP problem. Briefly explain your implementation and write down the pseudo code.

Implemented algorithms:

- P3P with RANSAC: Performs not really well
- DLT with RANSAC: Performed better

Pseudo code:

- P3P with RANSAC

```
get points2D, points3D matches with OpenCV Brute-Force Matching (applied ratio test)
for i in iteration_number:
    random sample 3 matching points and perform P3P algorithm
    this will get a list of R, T
    for each R, T: calculate number of inliers within the threshold
        keep track of the R, T that has the most inliers
return best_R, best_T
```

- DLT with RANSAC

```
get points2D, points3D matches with OpenCV Brute-Force Matching (applied ratio test)
for i in iteration_number:
    random sample 6 matching points and perform DLT algorithm to get the R, T
    with the R, T: calculate number of inliers within the threshold of
        keep track of the R, T that has the most inliers
return best_R, best_T
```

Note:

- 在實作P3P, DLT時基本上都照著講義上的步驟、算式將每一個參數算出來。
- 在做camera projection時都有使用OpenCV **undistortImagePoints**來處理Distortion (DistCoeffs)

Q1-2 For each camera pose you calculated, compute the median pose error (translation, rotation) with respect to ground truth camera pose. Provide some discussion.

| | Pose Error | Rotation Error |
|-----------------|------------|----------------|
| P3P with RANSAC | 3.6486 | 57.1705 |
| DLT with RANSAC | 0.0397 | 0.5137 |

- P3P with RANSAC

```
processing ...
100%|██████████| 129/129 [29:27<00:00, 13.70s/it]
pose error:3.6486178331289, rotation error:57.1704489208524
```

- DLT with RANSAC

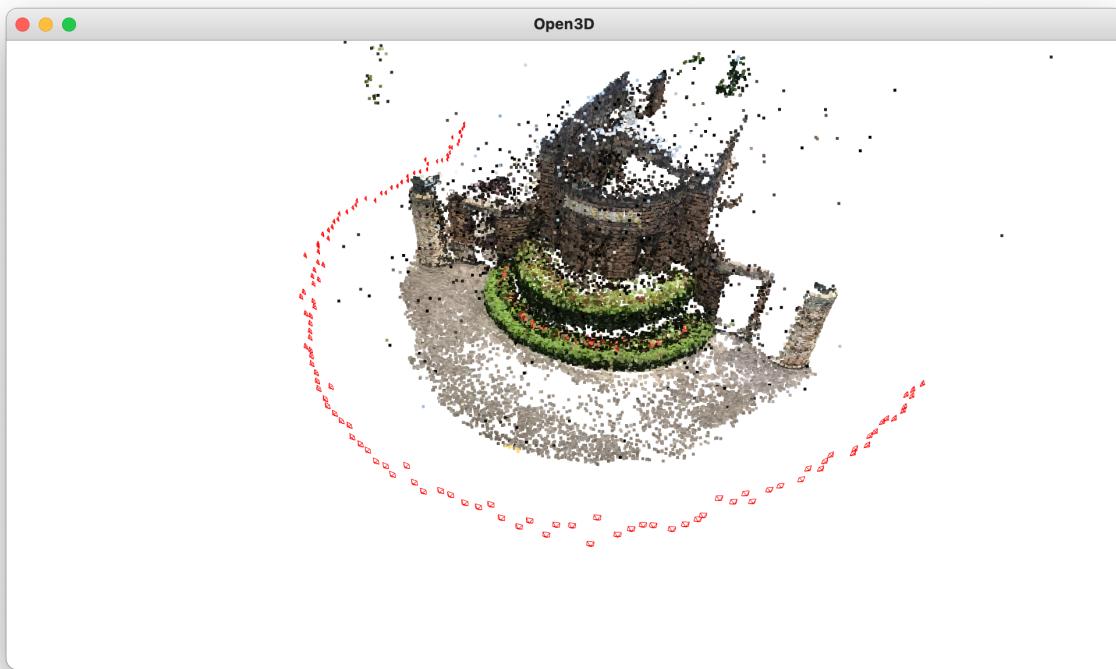
```
[  ┌─~ /De/c/3/homework2-kszuyen ── on ┌─ main !2 ?12
└─ python 2d3dmatching.py dlt
    loading data...
    processing...
    100%|██████████| 129/129 [30:25<00:00, 14.15s/it]
    pose error:0.03972756893383861, rotation error:0.5136728643749076
```

Discussion

- 結果上，不管是以Error來看，或是以後面的Q1-3 3D model, Q2-1 AR video，我實作的DLT演算法看起來才真的有將camera position給算出來。而P3P演算法與ground truth相差甚遠。
- 理論上P3P以及DLT應該都能夠求解，會造成誤差的應該是random sample非常容易sample到outlier。不過利用RANSAC在iteration次數夠多的情況下應該都能夠找到誤差夠小的組合。
- 以理論來看的話DLT因為要random sample 6個point matches, P3P只要random sample 3個point matches，因此DLT較容易選到outlier，而影響到表現。

- 我認為會造成我的P3P演算法無法準確預測camera position應該是我在implement時其中的code中有bug（雖然每一步驟都跟著課堂中講義做，但實在是找不到問題出在哪裡），以致於怎麼算都得不到良好的R, T。

Q1-3 For each camera pose you calculated, plot the trajectory and camera poses along with 3d point cloud model using Open3D. Explain how you draw and provide some discussion.



DLT with RANSAC

How I draw the camera poses:

- code: `visualization.py`
- Tools: `open3d.visualization.draw_geometries()`
 - 利用`draw_geometries` function可以在3D空間中繪製多個點、線，也可以先組合出一個圖案組合再繪製在3D空間中。
- Get the NTU school gate 3D point cloud and its colors from `points3D_df(points3D.pkl)`, and make it into a `open3d.geometry.PointCloud()`

- Use **open3d.geometry.LineSet()** to draw a quadrangular pyramid, get each images' camera position (rotation, position) from **camera_position_df(camera_position.pkl)**, then plot each quadrangular pyramid into its corresponding positions with the functions below:
 - **open3d.geometry.LineSet().rotate(rotation)**
 - **open3d.geometry.LineSet().translate(position)**
- with **open3d.visualization.draw_geometries()**, plot the 3D point cloud and the quadrangular pyramids into the 3D space.

Problem 2: Augmented Reality

Q2-1 With camera intrinsic and extrinsic parameters, place a virtual cube in the validation image sequences to create an Augmented Reality video. Draw the virtual cube as a point set with different colors on its surface. Implement a simply but efficient painter's algorithm to determine the order of drawing.



P3P with RANSAC



DLT with RANSAC

Environment

- python: version 3.9
 - numpy==1.26.1
 - open3d==0.16.1
 - opencv_python==4.8.1.78
 - pandas==2.1.1
 - scipy==1.11.3
 - tqdm==4.66.1