

Q1-1. Pseudo code:

- **main**

Create P3P solver

For all valid image

Find 3d-2d match point

Do 2d point undistortion

Do P3P and Ransac

Load groundtruth of valid RT

Save result of P3P and groundtruth as files for Q1-2

- **Ransac**

Calculate maximum of number of sample (99% correct), 1% outliers number

For maximum of number of sample

Random select 4 point for p3p (1 for validation)

Do P3P (return R, T)

Use R, T to calculate projection 2D point

Calculate points which errors inside count (means inlier)

Select R, T which get the most much inliers

Check if outliers number already < 1%, stop for loop immediately

- **P3P**

Compute angles and distances of 3D points

Compute x and select the real roots

Compute y

Compute radius (a,b,c)

Calculate camera center T by three sphere centers and radius

For all camera center

Calculate lambda and determinate

If lambda > 1 and determinate = 1

Save R, T as solution

For all solution

Compute the 2D point by R, T

Select the solution which minimum the distance between groundtruth

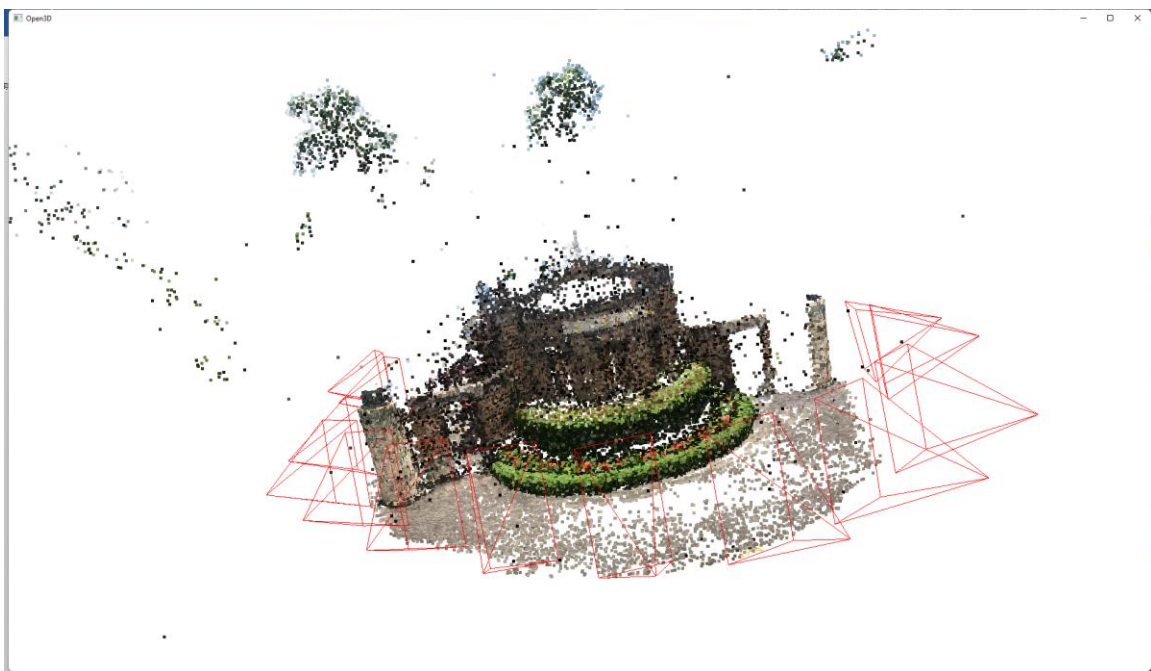
Q1-2:

Rotation Error: -0.000789290088732049

Translation Error: 6.340735147765215

Q1-3:

附圖為隨機選取多張 valid img 繪製結果



使用 open3D

先將 3D 點雲都標上

使用 Q1-1 計算出來的 RT 計算長方形四點在 3D 空間座標(1080*1920)

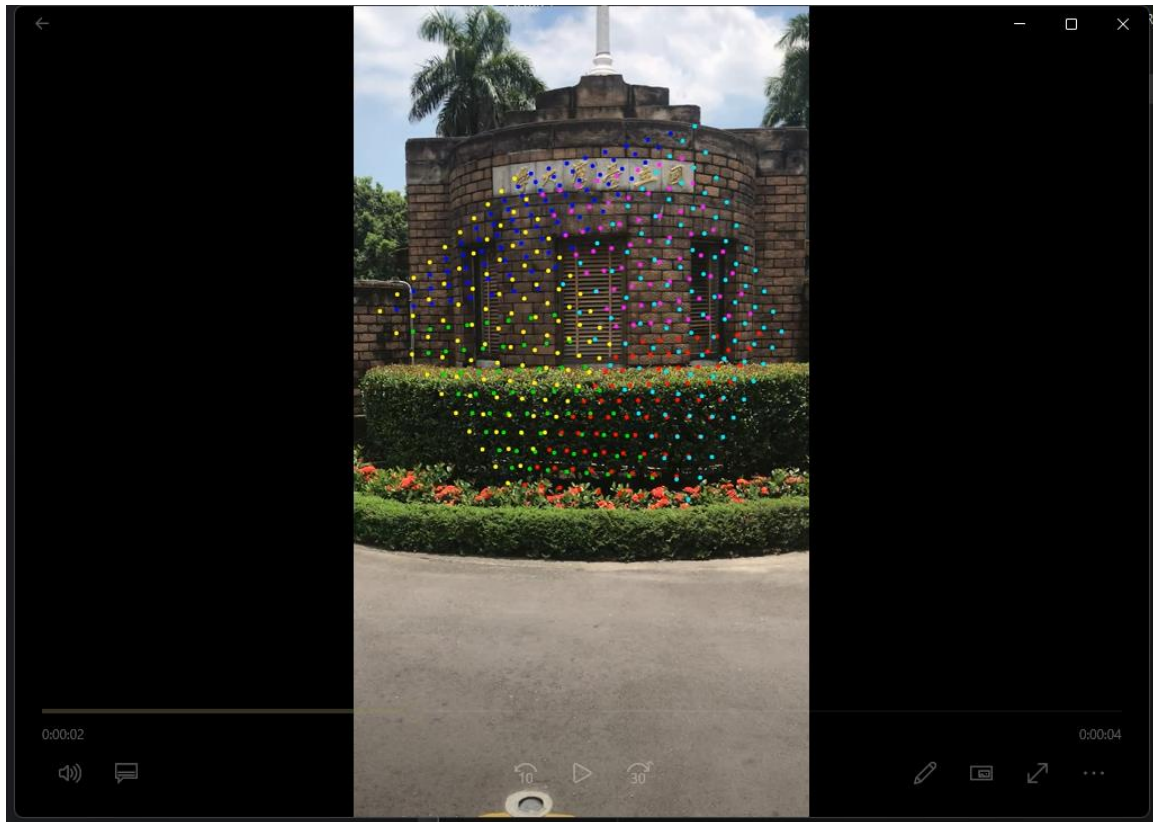
並將 T center 也串接起來成為第五個點

使用 lineset object 記錄點, 以及需要相連的點, 設定線條顏色

使用 sample code 提供的 `get transform mat` 設定 open3D 初始相機位置
繪製成功~~

Q2-1:

附圖為輸出影片截圖



Misc.

- **Python Environment: 3.8.13**
- **Package: OpenCV, numpy, pandas, open3d, scipy**
- **Q1:**
 - `python 2d3dmathcing.py`
 - `python draw_camera_pose.py`
- **Q2:**
 - `python AR.py`