

## 三維電腦視覺 hw2

姓名:葉冠宏 學號:R11943113

1.

執行環境:Python

安裝套件:

scipy.spatial.transform

pandas

numpy

cv2

math

open3d

### Problem1-1:

作法:

首先，在 matching 之前我們是有用 train image 去做各個 point id 的特徵向量平均，即類似 k means 的方法。接著針對 valid image 中的 2d,3d 點去用其各個 point id 的特徵向量去做 matching 的動作。接著，我們去把每個 2d 的 pixel 點先去做 undistort 的動作，用這些 undistorted points 和對應的 3D 點，我們去做 p3p 的演算法。在過程當中，我們是隨機取 4 個點的組合去做解。我們執行 100 次迴圈和 inlier width=8 去做 RANSAC 的動作，去用最多的 inlier 數目選取最佳的 rotation matrix 和 translation vector。

P3P 的 pseudo code 如下:

##

1.Compute cosines ( $\cos(\alpha), \cos(\beta), \cos(\gamma)$ ), distances ( $s_1, s_2, s_3$ )

2.Construct the equation from the slides and find the possible solutions

3.For each possible solution, use companion matrix method to find possible Rotation and translation. Identify correct solution by the 4th point.

4.Calculate the reprojection error and return the best rotation and translation given the input world and pixel coordinate.

##

### Problem1-2:

執行完實驗後的結果如下:

```
median of translation differences from ground truth is 0.060304524373485865.  
median of angle differences from ground truth is 1.2850149368989354.
```

我們得出的 translation 和 ground truth 的 vector 之間的 L2 norm 中位數只差 0.06。然後 rotation 如果用 rotation vector 表示的話，角度的差距和 ground truth 相比中位數只差 1.2，可以說預測的結果和實際上的差距不大，表現的不錯。還是會有一些差距可能由於在做 p3p 的時候我們是取樣其中四個點去做求解，當中還是會有一些隨機的成份在，所求得的可能是 local 最佳解，不見得是 global 最佳解。

### Problem1-3:

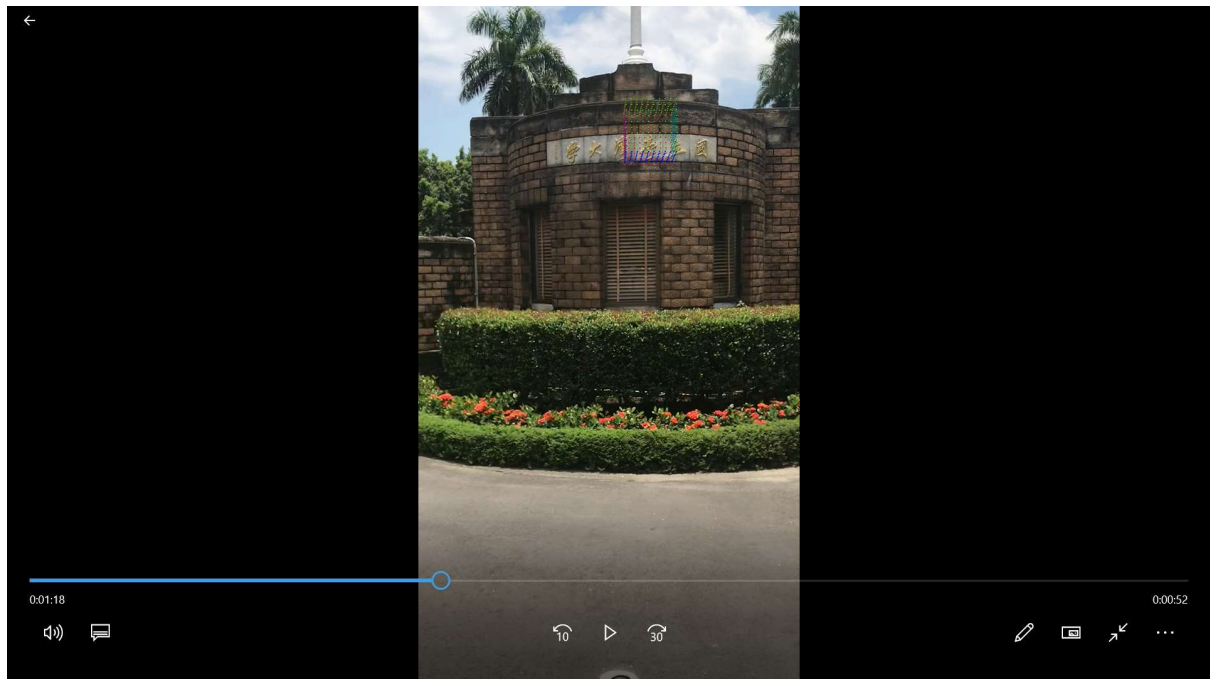
我們所畫出的圖如下:



其中我們的畫法是先做出一個 pyramid 是 center 在(0,0,0)的地方，然後底的四個頂點在 (0.5,0.3,0.3), (0.5,-0.3,0.3), (0.5,-0.3,-0.3), (0.5,0.3,-0.3)。接著，針對每張 image，我們去求其 camera pose，求出 rotation matrix 和 translation vector 後，我們針對那個 pyramid 做一個 transformation，去求其對於原來 pyramid 轉換後各個頂點的(x,y,z)的座標。然後用 lineset 的方法去用 open3d 畫上去。

其中我們可以看到所畫出的 pyramid 整體的感覺是很 smooth 的，除此之外我們也看到 pyramid 所面向的方位和實際上 image 所拍攝的角度也差異不大。顯示出我們的 camera pose 求出的解很好。

### Problem2-1:



我把 **cube** 的位置是放在國立台灣大學的招牌上面。得出結果的影片之後，我們可以看到影片中 **cube** 的絕對位置是一直沒有在動的，顯示出所得出的 **camera pose** 結果很好。

**Painter's algorithm** 我的作法是先針對 **cube** 中每一個點去做 **reprojection**，然後針對每個 **pixel** 點我們去用 **dictionary** 去做存取 **buffer** 的動作。除了存取每個 **pixel** 是被第幾個點投影外，我們也去計算存取那個點的 **3d** 座標和 **camera position** 的距離，以讓最後決定是誰去被投影。最近的點會被看到。