三維電腦視覺 hw3 姓名:葉冠宏 學號:R11943113

1.

作業環境:python

安裝套件:

open3d

numpy

cv2

sys

os

argparse

glob

執行方式:

python vo.py --input ./frames/ --camera_parameters camera_parameters.npy

其中./frames/ 為 image 檔所放置的位置。camera_parameters.npy 為 camera 的內在參數的檔案，即 camera_calibration.py 執行後所產生的檔案。

2 說明:.

Step1:Camera calibration

在這個步驟當中，我們藉由抓角點的方式來找尋相機的內在參數和 distortion 參數。

Step2:Feature matching

在這個步驟當中，我們藉由 ORB 的演算法和採用 HAMMING 的 norm 來去計算距離來去抓前後兩張 image 當中兩兩配對的點。採用 orb 演算法為 feature extractor 是因為我們是 online 的計算，所以需要較快、即時的運算速率。

Step3:Pose from Epipolar Geometry

找到前後兩張 image 配對的 2D 點之後，我們用 opencv 的套件 cv.findEssentialMat 去抓兩者之間的 essential matrix，然後我們藉由 cv.recoverPose 去取出 rotation matrix 和 translation matrix。

因為 open cv 的 cv2.recoverPose 求出的結果是 unit 的，所以我們需要求出前面的 frame 和後面的 frame 之間 scaling 的差異。假設現在有連續三張的 frame，我們藉由中間的那張 frame 當作橋梁去看說哪些點是在上一回合有出現在中間那張 frame 和這回合也出現在中間那張 frame

的。我們藉此去只取第一張 frame 中和中間 frame 共有點的對應點。我們藉由 frame1,2,3 的這些點和到目前為止的 rotation 和 translation 資訊就可以去用 opencv 的 cv.triangulatePoints 去求出前一回合和這一回合的 triangulation points。最後我們去算說前一個回合 triangulation points 兩兩的 norm 和這一回合的做比較，就可以算出前一回合和這一回合 scaling 的 factor。因為照理說同樣的點相對距離是只差一個比例的。

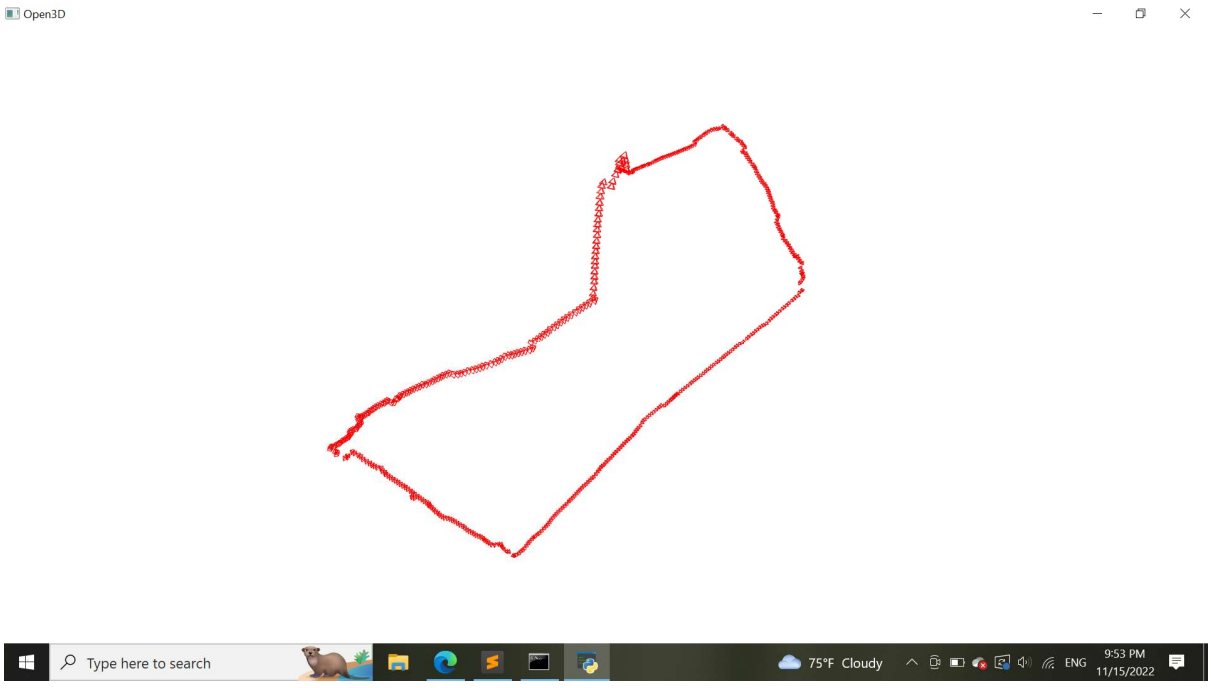
回合結束後我們有去存這一回合的 match, rotation, translation, image 等資訊以供下一回合使用。

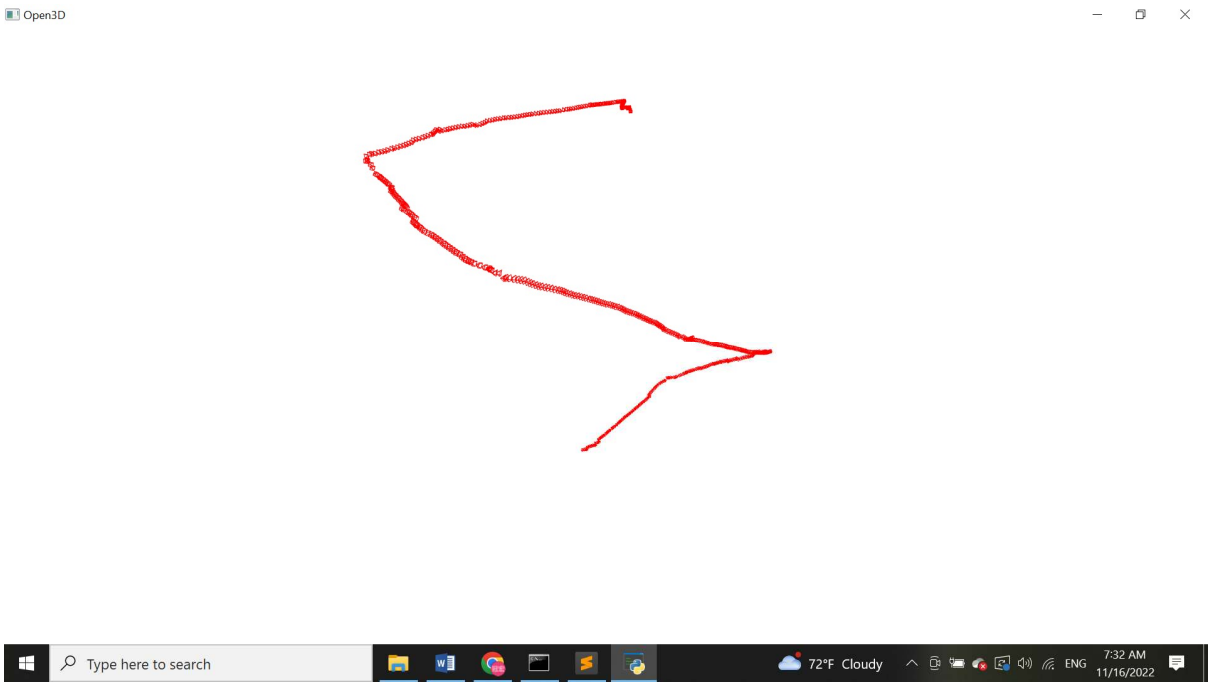
Step4:Results visualization

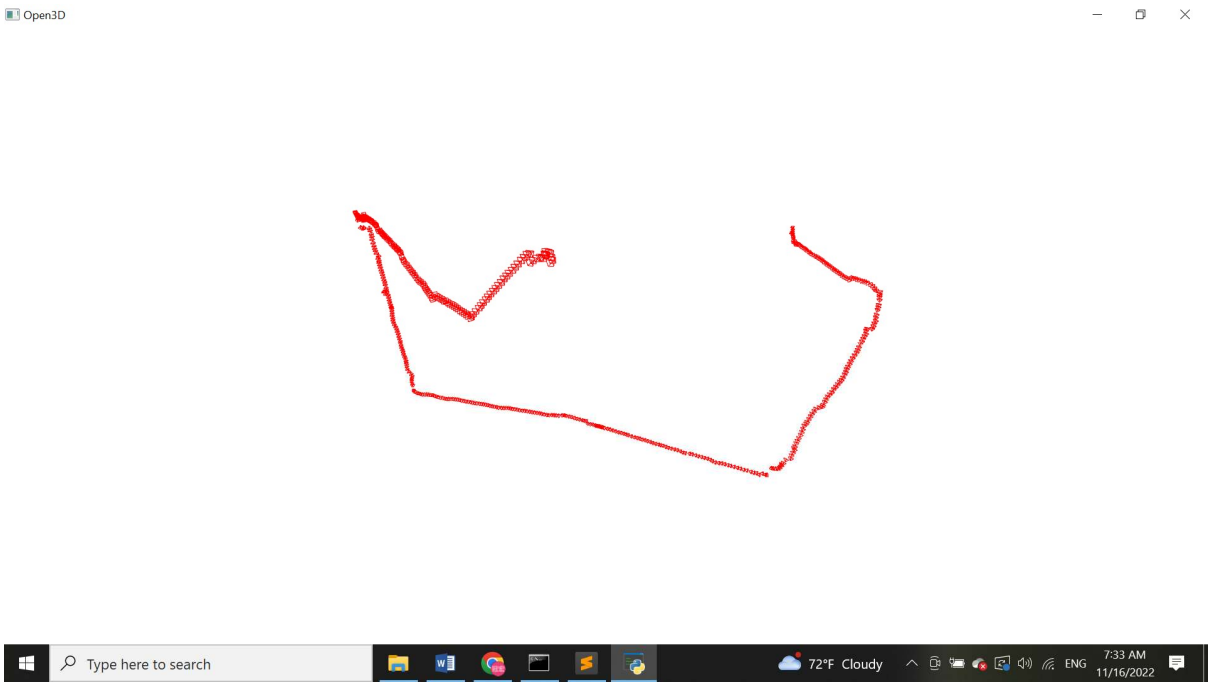
最後在每一個 iteration 的時候我們去累計從最初始那張 frame 到目前為止總共累積旋轉了多少，還有就是相對於最初始的 frame，你的位置在哪裡。得出這些資訊之後，你就可以用預先設定好的相機金字塔的初始座標算出其現在的中心點位置以及 corner 的位置在哪裡。最後我們用 open3d 的套件去顯示目前的路徑，也用 opencv 去顯示出目前的 frame 以及其特徵點。

3.實驗結果

從以下的結果我們確實可以看出從某個角度看其 trajectory 有繞出一圈。但從另一個方向看，可以看出其 z 軸的方向是沒有對上的，原因是因為我們的資訊量有限，僅僅可以從 2D 的點去推測可能的 3D 點雲位置。除此之外我們可以看到其中間有些地方沒有走的那麼的順，可能是因為拍攝的時候馬路的上下震動，還有就是如果有行人、腳踏車的穿越等都會影響判斷的品質。







Youtube 連結:

https://youtu.be/WJ-Dt7BL2QM