Final Project Report

第4組 吳原博 李懿麒 陳品戎

Test Video: `test_video.MOV`　　實驗過程影片: `orb-slam_demo.mp4`

影片拍攝場地: 交大十三舍走廊

影片長度: 1分18秒

幀率: 30FPS

攝影設備: iPhone XR

Video Preprocessing

code: `video2pic.py`

我們利用OpenCV讀取影片檔案，得到影片中每一個frame。因為之後讀取大量相片的運算(COLMAP)之時間複雜度很高，而且我們發現到網路上範例圖檔庫的每個frame之間有一段距離，這樣的狀況下仍然能夠成功繪製，所以為了節省運算的時間，我們每四幀只取一個frame作為後續處理的圖檔(30 → 7.5FPS)。

總共取出541個frame。

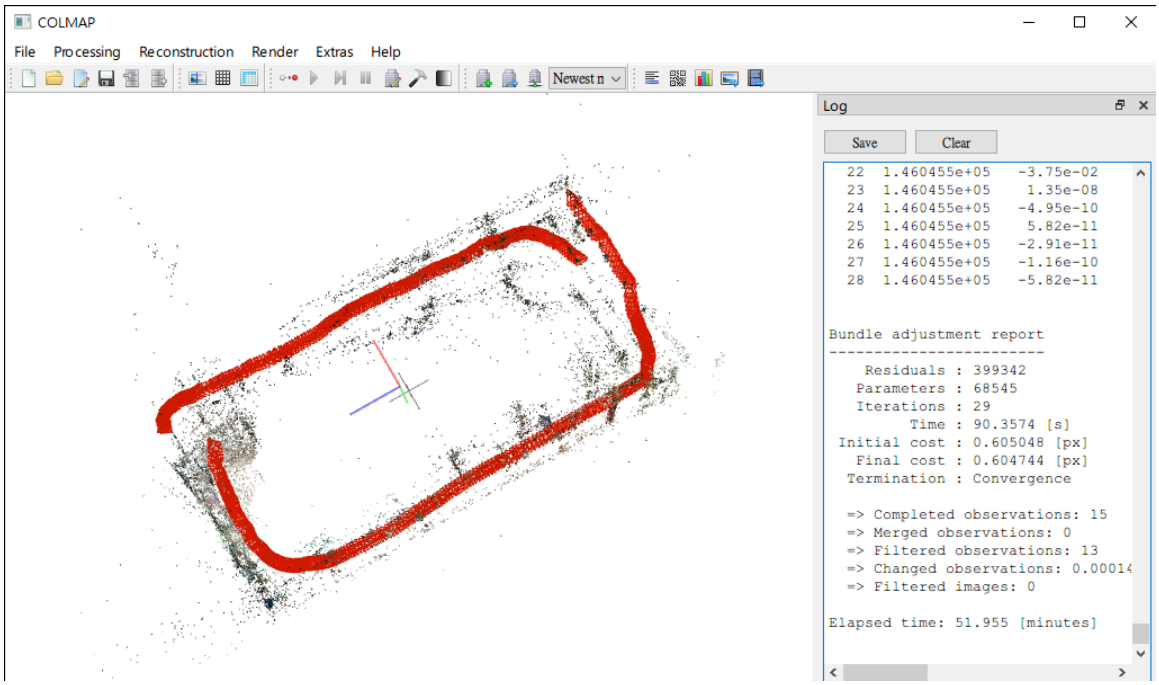
COLMAP

COLMAP 是一種多檢視立體 (MVS) 的三維重建技術，這樣的技術能夠從多個視角由外向內觀察和獲取景物的影像以完成匹配等任務。MVS 已經被廣泛應用於 3D 列印、離線地圖重建和文物修復等行業中。

使用COLMAP，需要先將欲處理的image存在資料夾內，並建立一個`.db`檔。之後依序執行Feature Extraction, Feature Matching與reconstruction，若繪製的結果並不滿意或一直畫不出來，也可以直接reset reconstruction重新來過。

在前幾次測試colmap時，生成的座標點經常無法成環。我們發現到座標點形成的路徑經常就斷在轉角處，原因應是在於拍攝時轉得太快導致前一個frame的feature已經不在下一個frame的範圍內，而無法match。因此我們在重錄時，會在轉彎處特別放慢視角轉移的幅度。

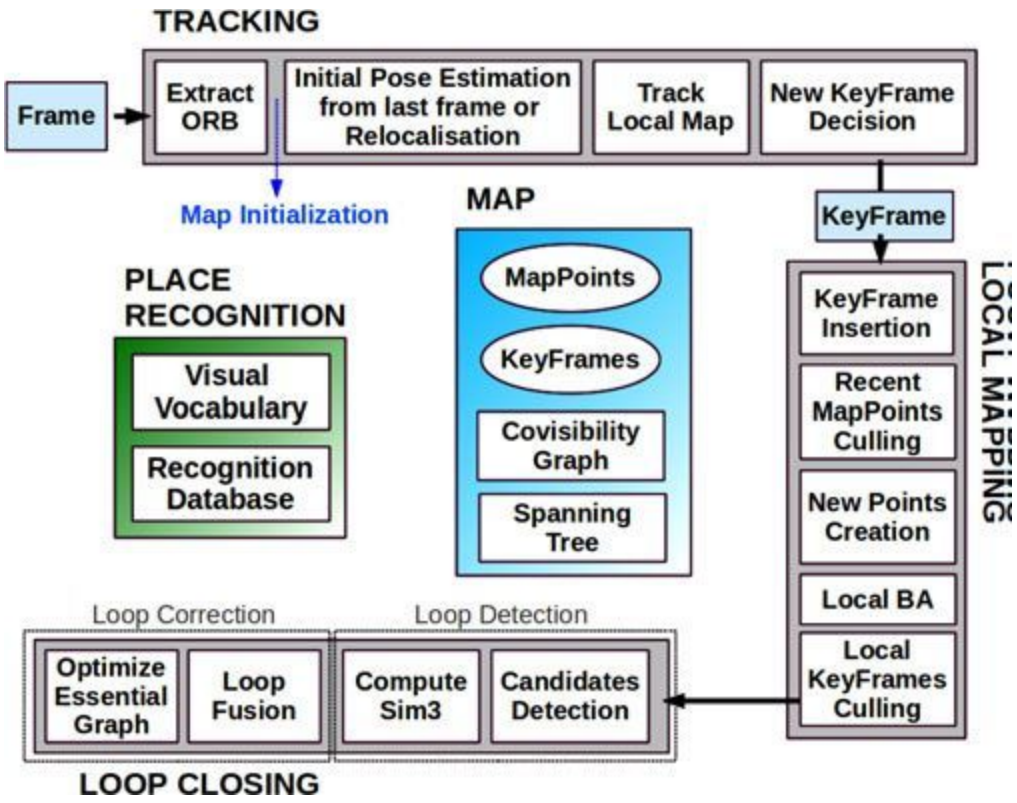
而另外一個問題是路徑偶爾會從錯誤的地方延伸，而延伸的方向是在走廊的直線上。這可能是因為走廊有著重複性高的features，容易誤判成同一個位置。



ORB-SLAM

ORB-SLAM 是用於單目、立體和 RGB-D 相機的多功能且準確的 SLAM 解決方案，並且能夠在各種環境中實時計算相機軌跡和場景的 sparse 3D 重建。ORB-SLAM 由 Tracking, LocalMapping, LoopClosing 三大流程同時運行 :

1. Tracking : 負責在每一幀中定位相機並決定何時插入新的關鍵幀
2. LocalMapping : 處理新的關鍵幀並執行 local BA 以在相機位置周圍實現最佳的重建方式
3. LoopClosing : 搜索每個新關鍵幀的 loop



reference : ORB-SLAM: a Versatile and Accurate Monocular SLAM System

使用ORB-SLAM，也需要先將欲處理的image存在資料夾內，並且還得在資料夾外留一個與資料夾同名的`.txt`檔案紀錄每一個Timestamp所對應的圖檔路徑。

我們是用助教提供的虛擬機來執行，剛開始執行時會報出找不到`.so`檔的錯誤。解決方式是從ORB_SLAM3中找出那些`.so`檔並放入`/usr/lib`。

在前幾次執行orb-slam時，生成的路徑在轉彎時很容易比真實的角度還要大得多，使得無法成環。後來我們發現是因為當初在求相機校正值的方法有誤，校正當時我們是以camera當作網路攝影機的模式連接電腦，跟錄影時的尺寸並不同。(後來我們發現內建的`TUM2.yaml`的校正值其實就還不錯了)

而我們發現到orb-slam在拍攝到比較單調的區域(例如牆壁)時，有可能會match不到任何features，而無法繼續做之後的繪製。因此我們在重錄時，會刻意往雜物較多的方向拍攝，使得features的數量會更多。

在路徑繪製完後，我們發現幾乎每次在儲存`KeyFrameTrajectory.txt`的時候，都會直接報出Segmentation Fault並跳出程式，但這似乎並不影響到output的內容，因此我們沒有多加留意。

Evaluation

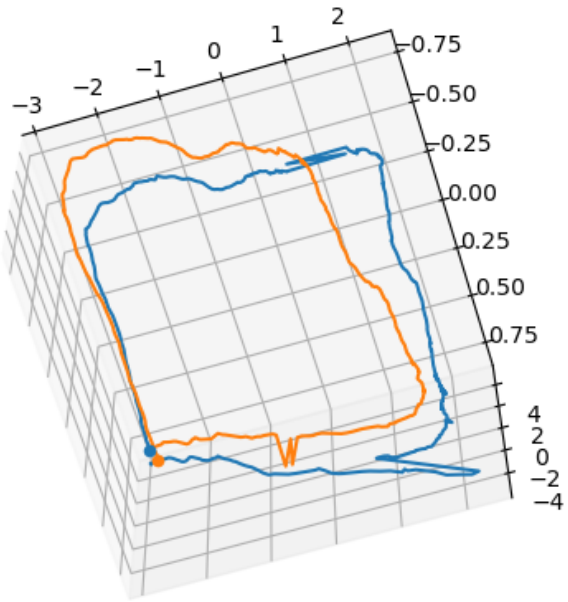
code: `traj_plot.py`

座標轉換

將ORB-SLAM的軌跡轉到COLMAP的座標系時，我們使用了ORB-SLAM提供用來做estimate與ground truth之間轉換的python function (`evaluate_ate_scale.py`的`align()`，是使用一個叫做Closed-form of Horn的演算法去實現，會用到svd等概念)。只需將要轉移的座標pairs帶入，就能得到需要的變換矩陣、平移矩陣及縮放倍率。

Transposed Point = (縮放倍率)*((變換矩陣)*Point + (平移矩陣))

https://github.com/UZ-SLAMLab/ORB_SLAM3/blob/master/evaluation/evaluate_ate_scale.py

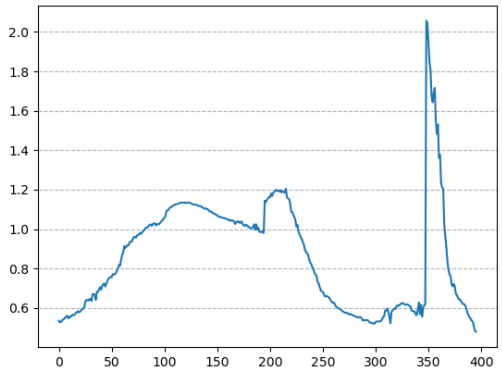


COLMAP: 藍色 Transposed ORB-SLAM: 橘色 (圓點為起點，皆為順時鐘方向路徑)

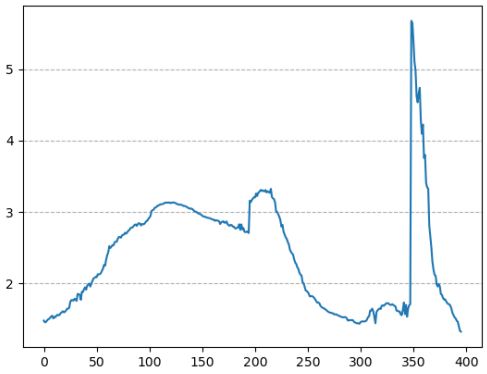
誤差

誤差為同一frame中兩種定位方法得到的座標之歐幾里得距離

$$d = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}.$$



誤差折線圖(由左往右是起點到終點)



誤差折線圖，Y軸單位為公尺

可以發現到在COLMAP的路徑斷點發生時(200,350附近)誤差值急遽增加。

實際換算

若以起點到第一個轉角距離約20公尺(在座標空間中距離約7.25單位)來估算，座標空間的1單位約等於20/7.25=2.76公尺

- 平均誤差: 0.862572 → 約2.4公尺
- 起點和終點誤差:
  - COLMAP: 0.071076 → 約20公分
  - ORB-SLAM: 0.022457 → 約6公分
  - ORB-SLAM表現較好，可能是因為它在執行的過程中會嘗試找出路徑的環狀接合。