

# 無人機自動飛航與電腦視覺概論期末報告

## 第十四組

組員:0611078 張祐誠、0611017 施君瀚、0616035 林均翰

### 一、兩種方法之間的差異

ORB-SLAM:

ORB-SLAM 主要進行三件事情：第一個是 Tracking，在提取每一幀的 ORB 特徵點，根據前一幀進行姿態估計，或著經過閉環檢測後，根據已經建置的地圖，重新優化定位。第二件事是 LocalMapping，從前面 Tracking 獲得 keyFrame，生成新地圖，並且在其中進行篩選跟更新。最後就是 LoopClosing，因為誤差會不斷累積，所以在經過一圈後，會先用 WOB 進行檢測，然後透過 Sim3 計算，最後進行校正與優化。

Colmap:

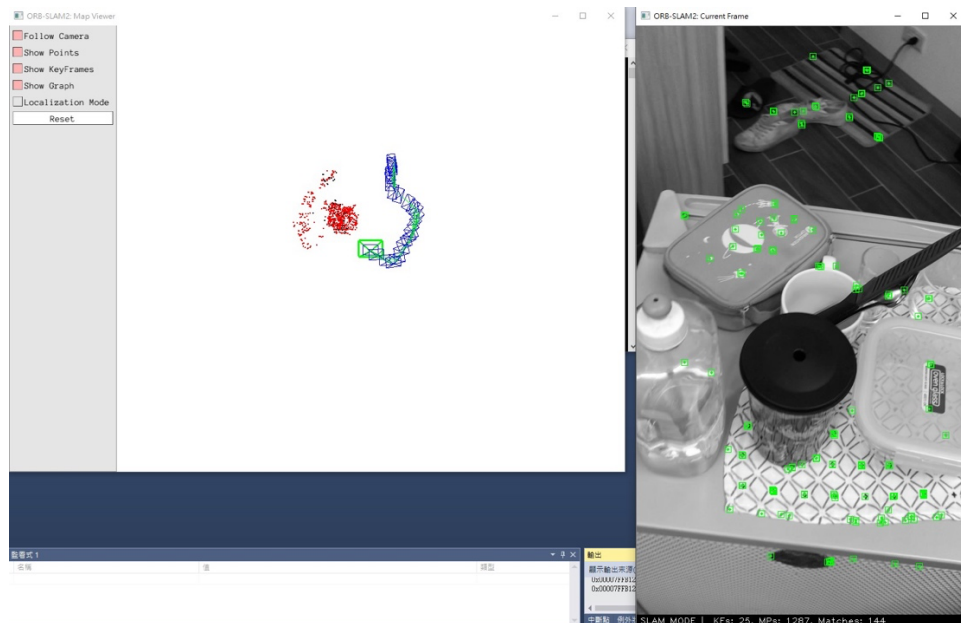
與 ORB-SLAM 最大的不同是，Colmap 需要事先收集好的資料集，然後再進行特徵點的提取，對特徵點進行匹配後，透過計算將其優化，最後輸出一個稀疏的三維點雲。

差異：

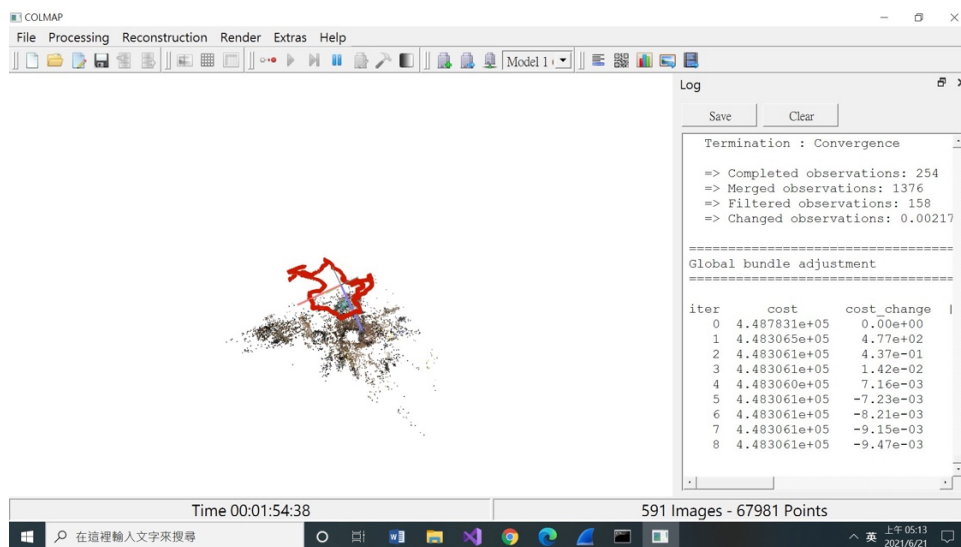
除了上述之外，一個很大的不同的是，ORB-SLAM 是在每一幀進行一次新的建模，代表著有即時建模的能力，相對的在 Colmap 則是需要先搜集資料再進行建模，才能進行比對，速度相當的緩慢。不過在過程中我們也發現，ORB-SLAM 在一些情況下，會無法擷取到畫面的特徵點，相對的在 Colmap 則是能很好的估計現場實際的深度，所以有很好的建模效果。

## 二、比較建出來的模型跟各自的相機定位

在模型上可以看出很大的不同，ORB-SLAM 只會建立出特徵點跟比對出來的特徵點，也就是圖片中的黑點與紅點；而在 Colmap 上則是有包含顏色資訊的點雲。不過正如前面所述 ORB-SLAM 的模型會再經過 Loop 之後重新校正，而 Colmap 則是先經過數個優化的計算過程後，將點雲加入地圖之中，但也會有 Bundle adjustment 的過程。不過直得提的是，Colmap 在儘管只有單顆鏡頭的狀況下，仍然有相當好的深度估計能力。



ORB-SLAM 建模過程

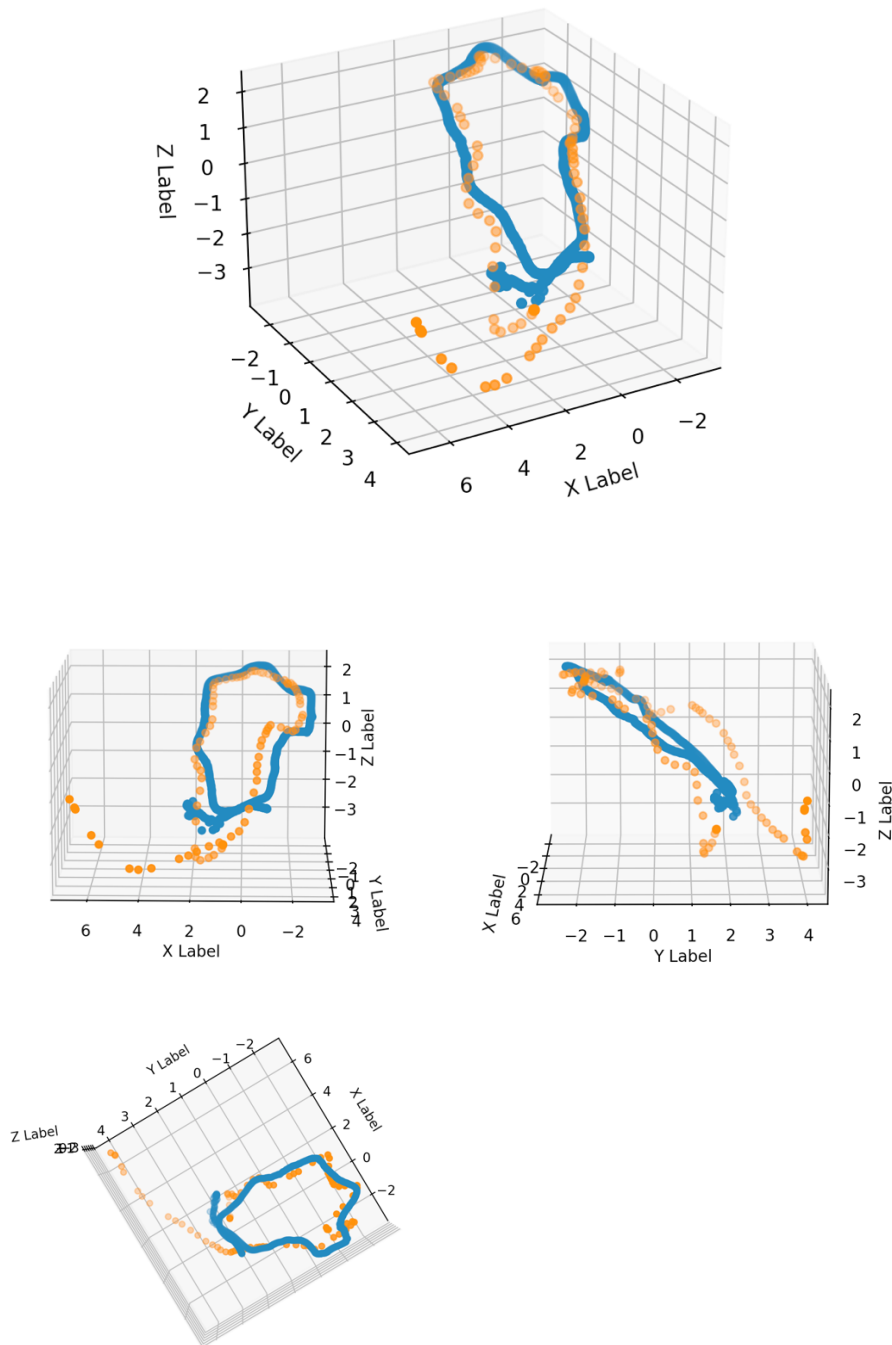


Colmap 建模過程

在相機的定位上，當然從相片是絕對無法獲取真實世界的座標的，而 ORB-SLAM 使用的是第一個辨識出來相機姿態的影格作為起始座標，事實上在初始化時，他會找出兩個模型（vbMatchesInliersH 跟 vbMatchesInliersF）然後透過評分機制找出適當的模型，最後透過 GlobalBundleAdjustment 來調整。Colmap 使用的座標是第一張圖為基準，利用 quaternion 產生的旋轉矩陣的反矩陣去推算相機的位置。

### 三、兩個不同方法所得到的相機軌道

下面四張圖是我們這次 lab 得到的相機軌道圖，藍色的是 Colmap 所建出來的圖，而橘色是 ORB-SLAM 得到的相機軌道圖，在這邊 ORB-SLAM 的座標係已經由旋轉和平移，轉換到 Colmap 的座標係。



過程中在特徵點充足下（如下圖），可以看到兩者所建立出的軌道其實相去不多，但在特徵點相對少的時候，ORB-SLAM就會偏離實際位置，可以在上面的解過看到這個現象。



在計算兩者的誤差，我們所使用的方法是 Root-Mean-Square，

$$error = \sqrt{\frac{\sum_{i=1}^n (\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2)}{n}}$$

而所有資料點所計算出的 error 為 1.7892075364576163，如果不考慮前面 15 個誤差較大的資料點，error 為 0.8131214164989768，是前者的一半左右，可以看到某些狀況下 ORB-SLAM 依舊有很好的建模能力。