

這個題目可以分成兩個步驟：找出關鍵角點、利用角點建出點雲

1. 找出關鍵角點

這個步驟目標是從每個畫面中找出特徵點，與一般 3D 重建演算法不同之處在於這個題目要從地面路標中去找出特徵點而不是整個畫面。

最直覺的方法是從 dataset 中提供的方框中去尋找，工研院這邊有個簡單的演算法，第一步是利用方框的位置去擷取 raw_image 中的部分，接著利用一個 threshold 去切割深色跟淺色區塊，可以大致理解為柏油路跟標線，接著再利用 opencv 中的 function 去優化這些淺色區塊的輪廓並找到角點，詳細的 function 有在說明投影片中的 introduction 提到，像是 findContours 等等。

不過上述這個基礎演算法在使用 threshold 時，可能會因為路面陰影的關係造成區塊切割不理想，同學們可以針對這個部分優化或是找尋其它演算法。

這個步驟中同學們也可以不用糾結於要找出當下 frame 內所有的角點，畢竟有些角點可能距離相機很遠比較模糊，是有機會透過連續畫面的資訊去補齊的。

2. 利用角點建出點雲

這個步驟工研院也有提供一個基礎演算法，同學們可以透過 pinhole model 的原理去直接將第一步驟找到的角點利用外參及內參投到 base_link 上，這步驟會使用到架設相機高度為 1.63 米這個資訊。

另外一種方法同學們可以參考 Structure from Motion (SFM)或是 SLAM，這個演算法是透過單一相機連續畫面中的特徵點去建出 3D 環境。但 SFM 與本題目不同的點在於本題目的特徵點限於地面上的特徵點且相機的角度有四種，除此之外，如果需要的話，第一步驟找的特徵點需要另外使用 descriptor 去找出 feature 來進行前後連續畫面的配對，所以朝這個方向來發展的話會需要克服一些困難，但相對來說建出的點雲也會比單純 pinhole 的方法還好。

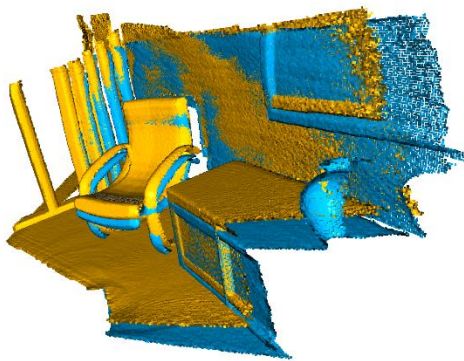
想要試試看的同學們可以參考 SFM 的演算法去將同一相機的前後畫面中的角點去建出點雲，但此時只會有單一面向的環境，接著再想辦法去合併不同角度建出的環境

● ICP 的計算

計算成績時第一步會利用 ICP 去對齊預測的點雲及 sub_map 中的點雲，同時會利用 initial_pose 當作起始點去 match，這個演算法會有 iterative 跟 threshold 可以調整，這兩個參數會影響配對的終止條件進而影響對齊的效果，同學們可以調整看看。

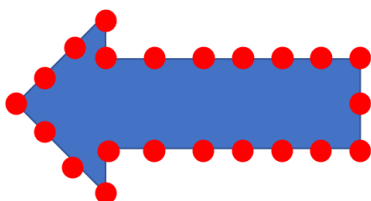
下圖是 open3d 中，使用 ICP 對齊的例圖。ICP 在對齊時兩個點雲的點數量不需要一模一樣，主要是兩個點雲的結構要相似，所以少一些部分是不會影響對齊的，不過當然結構相似的前提下，越多點匹配到會更加對齊。想更加深入了解的同學可以參考 open3d 中的 point-to-point ICP:

http://www.open3d.org/docs/latest/tutorial/Basic/icp_registration.html



除此之外，工研院那邊補充說在 sub_map 中道路中線跟雙黃線也是有標記點的，同學們可以將 sub_map 畫出來看看，不過在 detect_road_marker.csv 中沒有偵測到中線及雙黃線的資訊，但單純靠斑馬線、箭頭及停止線等建出的點雲就會有一定效果，而是否要去找出並標記中線和雙黃線的特徵點是同學們可以優化的方向之一。

關於 sub_map 中的資訊是工研院他們利用 lidar 去蒐集並人工標註的環境資料，而他們標註的點都只有地上的標線(中線/雙黃線/箭頭/斑馬線/停止線/...)，最後生成特定 frame 的 sub_map 時在以汽車的定位去擷取周邊 25 米內所標註的點。工研院也補充說 sub_map 中的角點與角點之間是有補點的，間隔是 0.8 米，下圖是個大概的示意圖，詳細狀況可以畫出 sub_map 來看。也就是說從 sub_map 中的點是可以看出標線的輪廓的，但是計算 ICP 時只要結構相似仍可以對齊，同學們不一定也要在角點間補點。



- 建立點雲時使用的資料範圍

做 ICP 時會將你們的點雲與 sub_map 去比較，而 sub_map 代表該時間點 25 公尺內的範圍，所以同學們如果只使用當下資料夾的資訊去建點雲會比較片面且範圍會較小，可能會造成對齊效果較差。

相反的在建點雲時可以考慮不用一次建完 sequence 底下所有資料夾中的點雲，可以只參考需要計算 ICP 的資料夾的前後某個範圍內的資料夾的資料就好