

SDC Localization Competition Report

310605005 王映勻

Pipeline:

1. Load the map

利用 PCL 函式庫讀取 pcd 檔的點雲資料作為地圖。

2. Downsample the map

利用 voxel grid 及 passthrough filter 將資料進行降維並去除欲觀測範圍之外的點，達到降低運算時間的效果。

3. Get initial guess

使用 GPS 的資料作為 x, y, z 的初值，而 yaw 的部分則是將 IMU 量測到的四元數轉換取其 yaw 的值。

4. Transform the frame of lidar scans from the lidar to the car

要利用 ICP 匹配點雲必須在同個座標系之下，因此這邊必須先將原先位於 lidar 座標系下的點雲轉換到車子的座標系。

5. Use passthrough filter to remove useless lidar points

利用 passthrough filter 將對 ICP matching 沒有幫助的點去掉，以提升點雲匹配的準確度。

6. Match the points onto point cloud map using ICP

使用最近點迭代演算法 ICP 將 lidar scan 的點雲與地圖匹配，有了合適的 initial guess 之後，針對不同的情況反覆測試，設定最合適的 ICP 參數。

7. Transform the frame of lidar scans to the world frame

得到 ICP 的轉換矩陣後，必須再將點雲轉換至地圖的世界座標系以完成定位。

8. Save the result as a csv file

最後將每一個 lidar scans 時間點車子的位置 x, y, z 及轉向 $yaw, pitch, roll$ 結果存至 csv 檔即完成。

Contribution:

這次主要是利用 ICP 這個演算法作為定位的方法，但若直接使用 ICP，未對資料進行任何處理的話，效果會很不好，於是這次我加入了以下幾個方法來提升定位的準確率：

1. Voxel grid filter

Voxel grid filter 運作的方式是在輸入的點雲資料上建立一個三維的網格，將每個網格內所有的點都用以該網格點的重心來近似，以達到資料降維的效果，利用這個方法將龐大的地圖資料點進行取樣的前處理即可大幅降低運算的複雜度及時間。

2. Passthrough filter

Passthrough filter 主要的功能是将沿著指定的軸向，將設定範圍之外的點過濾掉，僅保留設定區域範圍內的點。若點雲太過複雜，ICP 有可能很難收斂，因此就能利用這個方法將不合適的 lidar points 移除，以提升 ICP matching 的效果。

Problems and Solutions:

1. Initial guess 設定

對 ICP 這個演算法來說，好的 initial guess 能大幅提升它的表現。原本是想使用 IMU 測量到的數據作為 yaw 的初值，但第一及第三題算出來的值與後面 ICP 計算出的值相差太大，於是最後我是大略取 ICP 一開始算出來的 yaw 值作為初值。

2. 運算時間大且會有 message frame 遺漏

由於輸入的地圖檔案通常很大，為了降低運算的時間，我利用 voxel grid filter 將地圖資料進行降維。而由於第二及第三題的地圖範圍太大，我也使用了 passthrough filter 分別將 x, y, z 三軸上欲觀測範圍外的點移除，縮短讀取地圖資料的時間。

3. 定位結果誤差大

ICP 本身就有許多參數要設定，透過反覆測試並仔細觀察 rviz 的畫面，且利用 ICP 函式庫提供的 getFinalTransformation 所得到的分數作為評估的依據去調整出表現最佳的參數組合。

另外，為避免資料太複雜導致 ICP 難以收斂，可以利用 voxel grid 和 passthrough filter，不過這邊我考慮到 lidar points 沒有到很多，因此我只使用 passthrough filter 將對 ICP matching 沒有幫助的點去掉，特別是第二及第三題，y 方向的調整就對結果影響很大。且像是在第二題我一開始一直遇到無法前進的問題，後來發現移除 y 方向左半邊的點會大幅提升 ICP 的表現，調整 setEuclideanFitnessEpsilon 這個參數也能有效改善無法前進的問題。