SDC Localization Competition

311512064鄧書桓

1. **Pipeline**

一張含有 文字 的圖片

自動產生的描述

**基本code介紹:**

1. **Get the map**

在競賽一使用pub\_map.cpp讀取地圖(將pcd轉成ROS的資料形式再傳至topic供localization.cpp使用)，競賽二、三使用map\_publisher.cpp，先藉由GPS資訊將車子位置附近的幾個子地圖發布至topic(由於地圖太大，一次讀取完會導致效能降低)。

1. **Wait and receive /lidar\_points**

藉由pc\_callback這個回調函數對/lidar\_points這個topic做一些取值或些許調整，並藉由aling\_map將吃到的點雲做一些調整、疊圖，如voxel down 或是濾除部分點雲，詳細的解釋會在附上的code中做說明，以下只放部分照片供參考:

Voxel down(將點雲數量縮小，可以順便去除雜訊):

一張含有 文字 的圖片

自動產生的描述

尋找最佳的initial yaw(initial pose):

一張含有 文字 的圖片

自動產生的描述

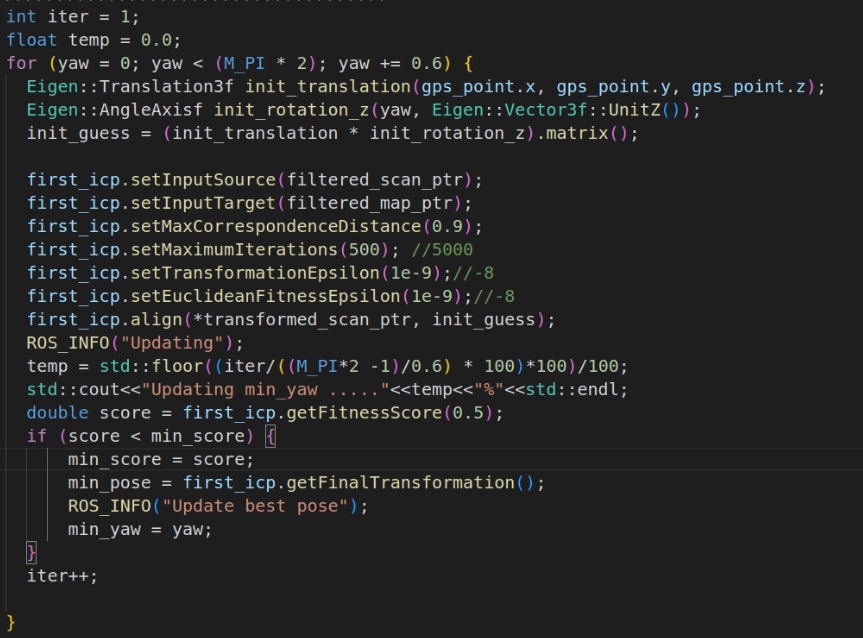
ICP部分參數調整與流程:

一張含有 文字 的圖片

自動產生的描述

1. **Contribution**

由於三個bag檔中的場景每個都差異很大，因此我這邊對於每個場景都有做case by case的調整。但是三個都有一個共同的部分，就是一開始找尋最佳yaw的方式(用於initial pose)，這邊使用助教提示的方式來完成，入下圖所示，透過嘗試不同大小的yaw，並以更低的getFitnessScore來更新最佳yaw。順帶一提，在這邊有使用過更低的間隔去做尋找(如:0.1、0.2與0.005)，但是它們既耗時且表現的也不會差到很多，因此後來就被我捨棄掉了。



接著是各場景case by case調整的部分。首先，先從第一個場景說起:

**場景一:**

這邊的ICP並沒有做多餘的變化，僅使用open source教的去做應用，本來考慮到新竹的風較大，會導致樹葉造成點雲疊圖的失誤率，因此想將地圖中上面包含到樹葉部分的點雲濾除，但是經過多次嘗試，儘管只有將樹葉的點雲部分濾除(也就是根據Z軸上去做限制)，效果也比全部點雲都吃進去做比較來的差，因此這邊就沒多做濾除。

另外，還有加入上網找到他人使用imu的方法，來根據慣性導航系統去更新transformation matrix(更新initial pose)。以下是使用到的imu code:

一張含有 文字 的圖片

自動產生的描述

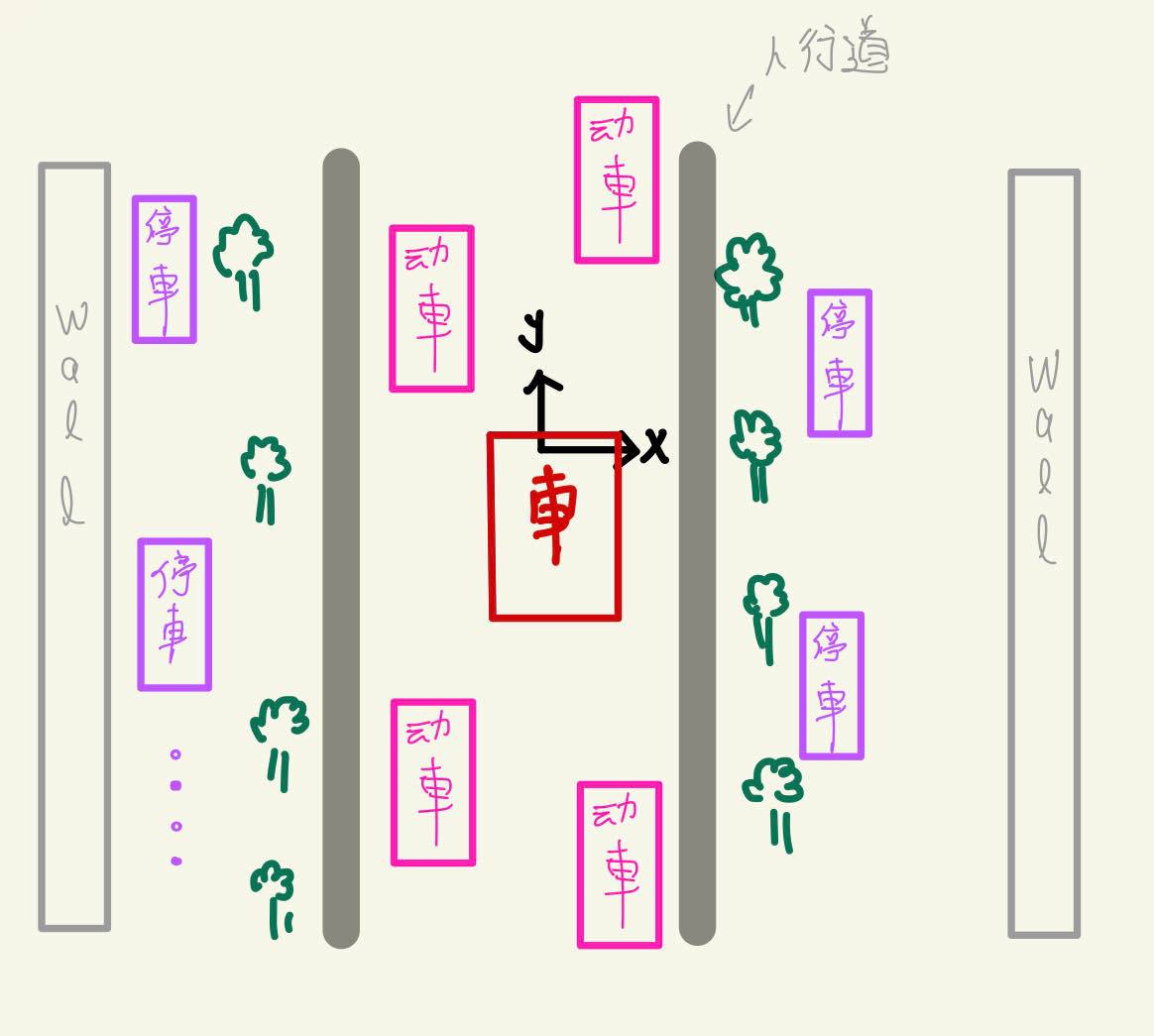
這邊的B是根據左下方這個式子去做計算，其中的各軸角速度是由imu 根據body frame 獲得的，而eigen\_C\_next是根據右下方的式子去做計算，以此來更新下一次做icp的初始值。這邊值得注意的是，/imu/data topic 發布的頻率比lidar\_points發布的頻率低許多，因此主要還是根據lidar\_point 中的點雲資料去做點雲疊合定位，imu的功能偏向輔助修正偏差。

一張含有 文字, 時鐘, 量表 的圖片

自動產生的描述 

**場景二:**

由於本場地在車子附近多了許多高精地圖上沒有的會移動的障礙物(汽車與巴士)，還有新的障礙物，如停止的汽車，因此這邊有將點雲資料許多做切割，示意圖如下:



因此我的作法為先做出左右兩邊僅有牆壁資訊的點雲、所有高於車子高度的點雲、左右由人行道至樹木的點雲(不包含車道，會有移動的車子)，最後將他們全部相加得到分割完的點雲後對y軸進行裁減，以避免吃到離車子太遠的資訊。而上述這樣分很多部分去做雲的原因為這樣相較於直接使用切割的方式，能夠較好的去觀察各部分點雲切割的狀態。以下為這部分的code:

一張含有 文字 的圖片

自動產生的描述 一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述 一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述 一張含有 文字 的圖片

自動產生的描述

而這個場景中的/imu/data topic傳遞的頻率遠低於lidar\_points，導致有加跟沒加一樣，因此這邊就不使用imu資訊。

**場景三:**

在這個場景一樣會有許多移動的障礙物車子，但是經過多次測試，有沒有濾除出來的效果很不明顯，因此最後僅對z軸做限制，將部分地板濾除而已。

而imu部分雖然其發布頻率比lidar\_points的低一些，但是仍能做到修正的效果，因此有加入使用，這部分與場景一使用相同的code，在此就不再做過多贅述。

**3. Problem and Solution**

在做本次期中競賽時主要遇到三個很麻煩的問題，第一個是場景二與場景三的地圖無法顯示的問題，由於我對於Rviz沒有很熟，所於在debug時花了很多時間，最後跟朋友一起找出來ppt上面指令給錯的問題(localization少一個a)，與Rviz的config檔中的topic給錯名子(/map給成/world)。另外，第二個問題為Rviz有時候在docker無法使用的問題，單是這個一直找不到答案，因此後來都改成在本地端開啟Rviz。第三個問題為ICP超參數的選擇方式，由於第二章圖的行進方式為後退一段距離後往前暴衝，這段因為速度太快會導致疊圖疊得很差，後來是將icp.setMaxiimumCorrespondenceDistance(x)調高才能成功疊到圖，這個函式是用來將大於x cm的點雲當成outliner濾掉，調高可以將較遠的點雲圖疊在一起。