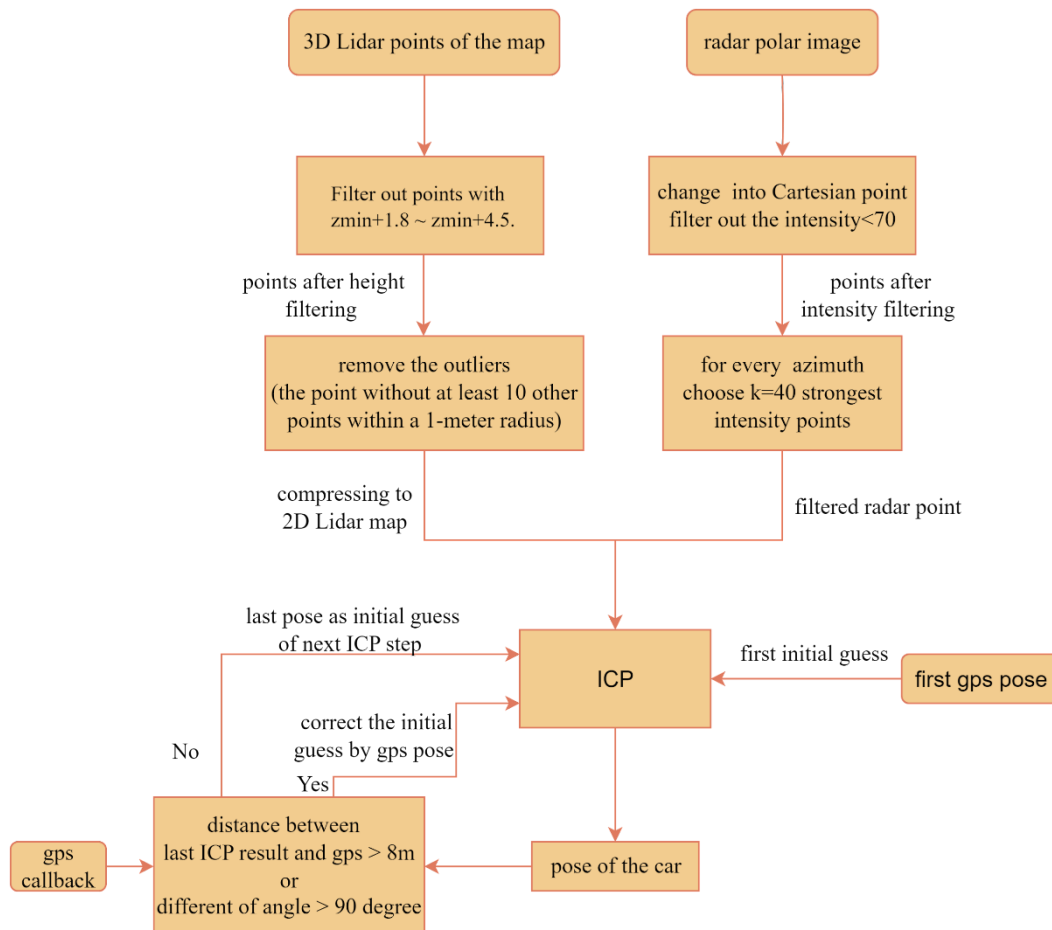


SDC Midterm Competition : Radar Localization Report

312512005 黃名諱

1. Pipeline of my localization program:



I. 3D Lidar points map 處理:

原先的想法是想辦法找到 radar 確切掃的範圍並取出範圍中的點，後來有找到 radar 規格及其上下俯仰夾角 1.8° ，但最後找不到在 off-line 情況下取出此範圍點的好方法，詳細緣由會在後面問題討論中提及。總而言之最後還是使用如助教的方法，取某高度範圍內的點，我 try 出來 $z_{min}+1.8 \sim z_{min}+4.5$ 是會有較好定位效果的區間，另外會發現有些許噪點，所以我用了 PCL 庫的 RadiusOutlierRemoval 來濾掉零星雜點，最後轉為 2D point cloud map 在 localization 使用。

II. Radar Polar image 處理:

此部分主要採用助教之方法，經過三角函數計算 radar image 能從極座標轉到卡氏座標上的點，接著用強度做過濾，濾掉強度小於 70 的點，70 是我 try 出來保有足夠特徵也不會有太多噪點而能使定位較佳

的強度閾值，接著用 k-strongest filter 每個方向取強度前 40 強的點，40 是助教提供的論文中使用的，我試出來也有較好定位表現，處理完之 radar point cloud 再接著再 localization 使用。

III. Localization:

將處理好之 radar point cloud 和 lidar point cloud map 做 scan matching 來定位，我主要定位還是基於 ICP 來定位，首先最一開始使用 gps 位置當第一個 initial guess，接著每次 ICP 的結果當作下次 ICP 的 initial guess，但我有加入一個檢查保險的步驟，當 gps 收到 gps pose data 時，會去檢查 gps 位置和上次 ICP 定位後的結果距離差距是否過大(>8m，大約車道寬度一半)及 yaw 角度是否差 90° 以上(差 90° 以上已經是反向了)，若差距過大會讓兩者做一個加權平均來當下次 initial guess 來去確保若我 ICP 定位結果有誤，能被 gps 修正回較正確的位置，詳細會於 contribution 處再做討論，以上為我的定位算法的大致架構及流程解釋。

2. Contribution:

我的定位算法中，主要與他人不同的貢獻在於加入了一個 gps 檢查修正步驟，以及提出一個對於 ICP 的 setMaxCorrespondenceDistance() 參數調整的估測方法。

1. Gps 檢查修正:

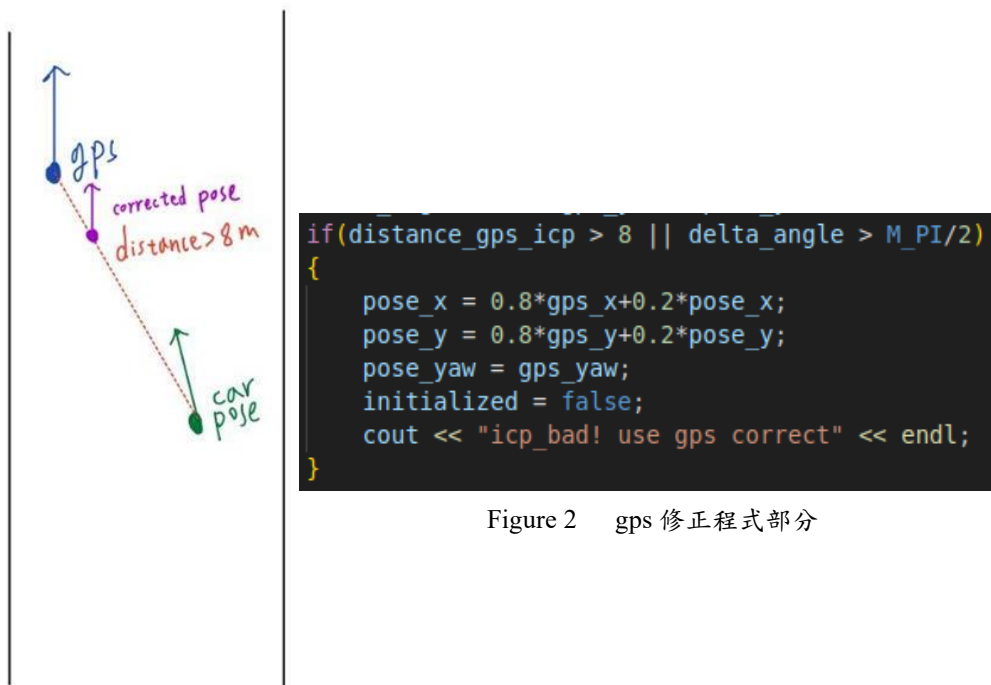


Figure 2 gps 修正程式部分

Figure 1 gps 修正示意圖

我在前期做 ICP 時，常有 ICP 沒有定位到準確位置的情況，而 ICP 又是以上次 ICP 結果當 initial guess，若定位一旦出錯而偏離正確位置太遠，之後的 ICP 就也很難再定位回正確位置去，導致陷入定

位失敗而卡在局部無法導正的情況。

為了解決此問題，我想到有 gps 的資訊可使用，雖然 gps 有 noise 存在，但至少是在正確位置附近，因此我的做法是在每次收到 gps pose data 時，就做一次檢查，檢查 gps pose 和先前的 ICP 位姿會不會有太大差距，若差距過大時修正，判斷差距過大的標準是距離 $>8\text{m}$ ，大約車道寬度一半，及 yaw 角度是否差 90° 以上，設定的距離閾值標準是若設太小有時候 ICP 只是差了一點就被 gps 拉走，因 gps noise 存在反而使左右定位變糟糕，因此抓了一個相對車道適中的距離，而 yaw 角方面，考慮到車朝向至少要和 gps 在相同前進方向差 90° 以上已經是反向了，因此這樣設定。

而我也考慮到 gps 有 noise 存在的問題，所以我在做修正時不完全相信 gps 之位置，而是如上 figure1 中所示，以 gps 位姿及上次 ICP 定位位姿做加權平均， $0.8 * \text{gps pose} + 0.2 * \text{last pose}$ ，較相信 gps 位姿但又不完全相信它，這樣做修正能將位姿拉回比較合理的地方，同時也有考慮到 gps noise 的問題從而完成整個合理有效的檢查保險機制。

2. 對 ICP 參數 setMaxCorrespondenceDistance() 的估計方式:

在 ICP 參數調整過程中，我發現 MaxCorrespondenceDistance 對整個定位效能是最有影響的，但是漫無目標亂調整不是一個好方法，所以我想說先估測出大概可能的參數區域去調整會較有效率，此部分雖然在算法中沒有體現出來，但此參數確實很大程度影響了我的定位結果。

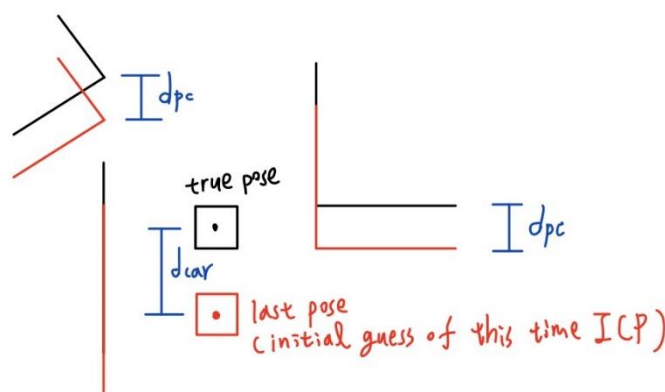


Figure 3 CorrespondenceDistance 示意圖

MaxCorrespondenceDistance 指的是能容忍的對應點之間的最大距離，所以此值太大可能導致一些離很遠的噪點被 ICP 考慮進去，太小則可能導致我真的需要的對應點距離都比此值大而使 ICP 有效點太少而失效，為此我的想法很單純，由於我的 initial guess 是上一次 radar data 做 ICP 得到的位置，而此次 radar 點雲要對到地圖對應的點雲，這個距離 d_{pc} 應該會跟我車子從上次收到 radar data 時到這次收到 radar

data 這段時間的移動的距離 d_{car} 大致相同，如上 figure 3 所示(圖畫的有點誤差請見諒)，為此我只要能估算出車子在這段時間內大概移動多少距離，就能知道此參數我應該取多少較為恰當。

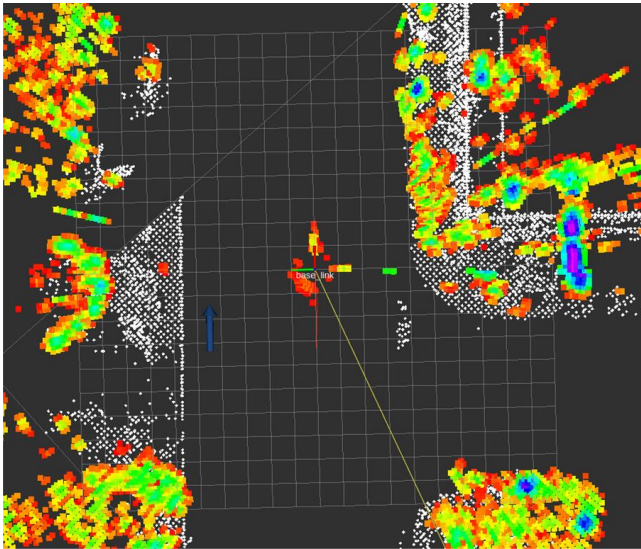


Figure 4 at time t-1

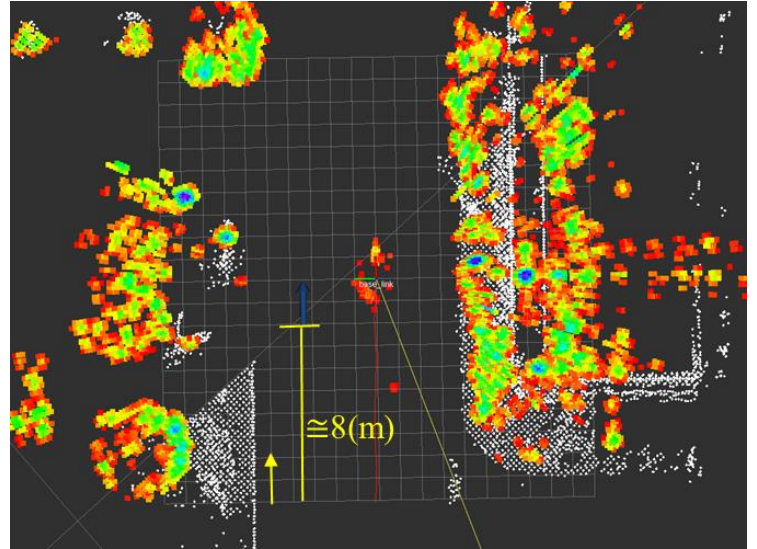


Figure 5 at time t

我的估算方法如上圖所示，是以 gps 來做推論，已知 gps 和 radar 的頻率及週期，藉由兩次收到 gps data 間及地圖網格的大小約 1m 得出車子平均移動速度，再由此平均速度推出兩次收到 radar data 之間車子移動的距離，推論如下：

$$\text{gps: } f_{gps} = 0.8\text{HZ} \Rightarrow T_{gps} = 1.25\text{s}$$

$$\text{radar: } f_{radar} = 4\text{HZ} \Rightarrow T_{radar} = 0.25\text{s}$$

兩次 gps 間車子移動距離約 8m

$$V_{car} = d_{gps} \div T_{gps} = 8 \div 1.25 = 6.4(\text{m/s})$$

$$d_{radar} = V_{car} \times T_{radar} = 6.4 \times 0.25 = 1.6$$

如此能估出大概 radar 兩幀間移動距離約 1.6m，但此只是很粗略的估算，但至少給了我調整 MaxCorrespondenceDistance 的一個初值，我在 1.6 附近調參，最後試出設 2.5 會是能同時使兩個 track 達到較好定位效果的值，而其他參數相對來說沒有太大的影響，設定則如下所示：

```
icp.setMaxCorrespondenceDistance (2.5);
icp.setMaximumIterations (100);
icp.setTransformationEpsilon (1e-5);
icp.setEuclideanFitnessEpsilon(1e-5);
```

Figure 6 ICP 參數設定值

3. Problem and solution:

接著此部分包含我在定位過程中曾發現的問題及想出的解決方法，但可惜最後這些方法並沒有得到更好的定位效果而最後沒有採納。

1. Radar 過濾:

前面提到強度閾值我設 70，原因是其保有足夠特徵也不會有太多噪點，可見下 figure 7 和 8，可見設閾值 50 時點有些太多且多了一些噪點，計算負擔可能提升且因噪點而定位效果降低

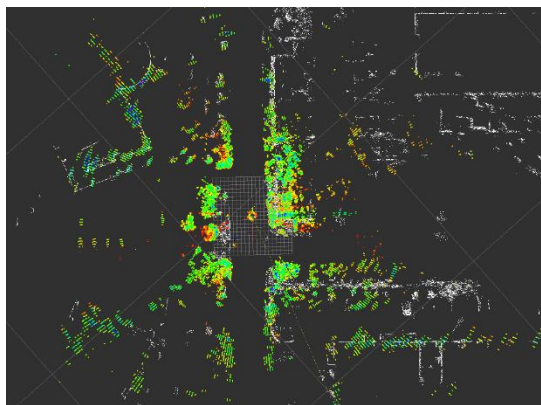


Figure 7 強度閾值 50

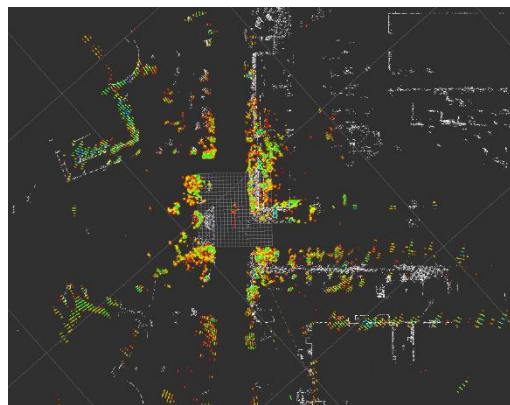


Figure 8 強度閾值 70

其實原先我在 code 中還有使用 RadiusOutlierRemoval 或 StatisticalOutlierRemoval 想去除噪點及用 voxel 做 down sampling，但定位效果上最後還是原本只用強度及 k-strogest 過濾會有最好的效果，因此最後就沒採用這些方法。

2. Lidar point map 點的選取:

我先前也有提到，我一開始對建構 2D lidar point map 的想法就是能取出 radar 掃的範圍的那些點應該會是最符合 radar 點雲的，因此我查了 radar 廠商的規格書，發現其是以一個上下俯仰夾角 1.8° 發射波的，而如果像助教的方法取某高度範圍內的點的話，會導致近車身處會有些不在 radar 掃描範圍的點被算進來，在遠處則是有在 radar 範圍內的點沒有被算進來，因此我就想說取出相對車身在此夾角範圍的點會更好，但後來發現是沒辦法計算的，因為我的原始 Lidar point 是相對世界座標的，因此在 off-line 建置地圖的狀態下我無法計算出這些點相對車身 radar 的夾角，造成此想法在這次定位上最後是不可行的，但我後來報告時教授也有提到，如果改為 online 建置地圖，就能知道車身位置，或許就能做到取出此夾角範圍的點當地圖，使地圖更符合 radar 的範圍，達成更好的 scan matching 定位效果。

Specifications	CTS350-X
Operating Range	500m
Operating Update Rate (Rotation Speed)	4 Hz
Operating Frequency	76 - 77 GHz
Range Resolution	0.175m
Azimuth Beamwidth	1.8°
Elevation Beamwidth	1.8°

Figure 9 radar 規格

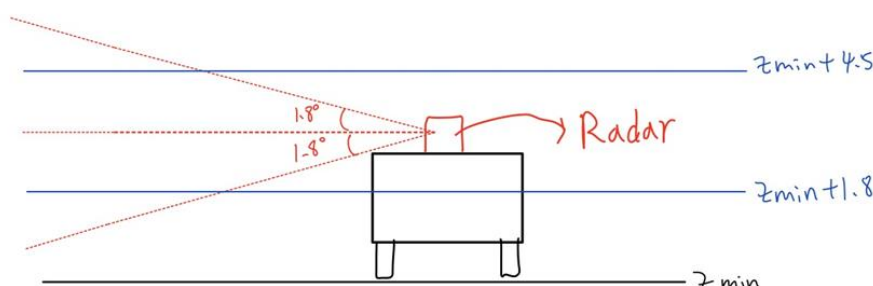


Figure 10 夾角示意圖

3. ICP 定位效果提升方法:

為了提升 ICP 定位表現，以下是我嘗試的兩個想法，但最後沒有較好的表現而放棄使用。

I. ICP with big and small MaxCorrespondDistance :

此前有提到影響 ICP 最大的是 MaxCorrespondDistance，而我想說我先前提出的估算只是利用某時刻的速度估算，但車子在各時刻移動速度不一定相同，每次點雲需要的 MaxCorrespondDistance 大小可能不同，我的方法是在每次定位時跑多種不同 MaxCorrespondDistance 大小的 ICP(礙於算力限制我最多每次取 2 個值跑)，再看哪個的 ICP score 比較低，就取其當較好的最後定位結果輸出，讓每次 ICP 定位時都能自適應的調整需要的 MaxCorrespondDistance 參數，我測試時使用 5 當較大值及 2.5 當較小值來做，最後定位效果卻沒有比較好，我認為此說明了 ICP score 只代表點雲間誤差大小，但不是完全代表你的定位好壞，畢竟還有噪點存在，才使此方法沒達到預期的效果，因此最後我沒有採納使用。

II. Go Forward:

另一個問題是如下 figure 11 所示，track2 在轉彎時有往回跳的現象，因此我想說解決此問題會不回提升定位表現，我採用的想法是說車子都是往前跑的，所以下一次定位結果

應該要在上一個位姿車子前方，所以我的方法是檢查定位結果有沒有在我上一刻車身坐標系 heading 方向的前方，若在後方(在上一刻車身坐標系下 $X < 0$)，或 yaw 轉變超過 90° (在這個頻率下不太可能和上一刻 yaw 差到 90° 以上)，我就認定此次定位有誤，而不採用，而是直接讓車子往原本的 heading 方向前進一小段距離當此次位姿輸出。

但此方法定位效果也不好，我後來想想應該是因為我的方法有個隱藏假設是上一刻位姿是正確的，但或許實際上上一刻位姿是多往前估的，則這次定位 ICP 當然會往回定位，但你卻判斷是 ICP 出錯了，導致後續錯誤，可能就是此方法沒有好定位結果的原因，因此最後我也沒有採用。

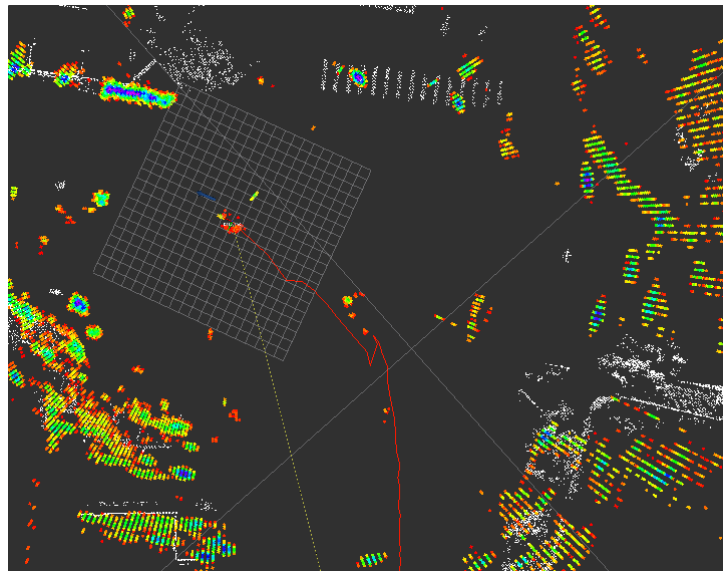


Figure 11 定位往回跳問題

4. Others:

其實在整個寫定位算法的過程中，中途 ICP 碰到很大的瓶頸，我調整參數調了 4~5 天都還是定位定得很差，挫折感非常大，直到後來才知道是我理解 ICP 給出的 FinalTransformation 出了錯誤，ICP 官方文件對得到 Transform matrix 的部分沒有很完整的定義說明，導致我原先是先將 radar 點雲轉到地圖坐標系上再帶入 ICP 中，以為得到的轉換是如下 figure 12 所示，但這是錯的，導致我 ICP 結果當然被我轉成錯的東西，最後才知道在 ICP 函式可以帶入 initial guess 直接得到新位姿相對世界座標的 Transform matrix，總而言之，透過這次期中比賽我熟悉了 Ros 及 ICP 的使用，也修正了我原先錯誤的理解，最後調出不錯的定位結果還是有蠻高

的成就感的。

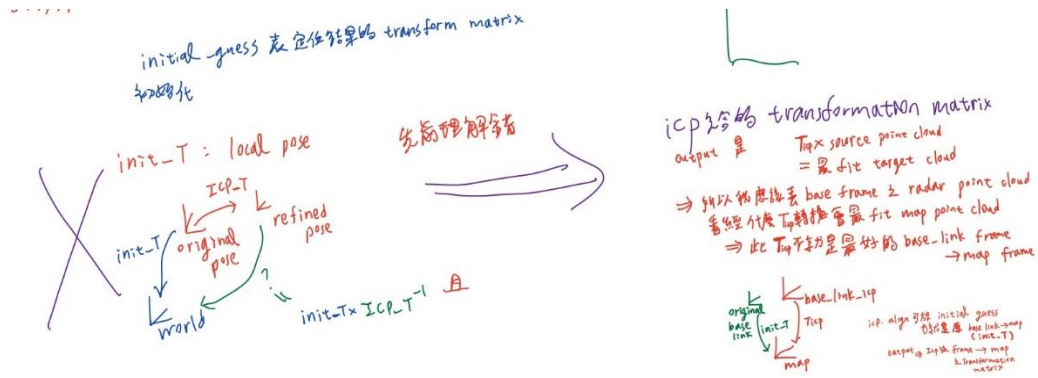


Figure 12 ICP 座標轉換矩陣錯誤理解