

Self-Driving Cars(Mid)

309611087 洪得瑜

2022 11/17

1 ITRI.bag

ITRI bag 實際流程如圖 1 所示，在 ITRI bag 中啟動 ROS 後建立共三個節點高精地圖、GPS 及光達頂雲圖，在近入光達節點後需確定第一次 GPS 及高精地圖是否 Ready, 若為 false 則等待 0.05 秒後進入雲點圖姿態及位置估算，當進入點雲估算時先對點雲圖進行降採樣有助於降低雲點數量及提高匹配皆設定為 1.0，完成降採樣後進行雲點初值估測，而第一步的位置採用 GPS 作為 trasn 設定，而初始姿態無法從感測得知因此在此採旋轉 360 度，尋找當中由 ICP 估測後的最小值，在此我以 10 度為單位收斂設為 10 次加快尋找，找到大概姿態後進行第一次的 NDT 定位，而在 ITRI bag 中我使用兩種點雲演算法 ICP 及 NDT 進行比較，最終將點雲圖估測後姿態位置儲存下一次初始設定，並將結果發布及儲存，在進行下一步估測值到 bag 資料結束。

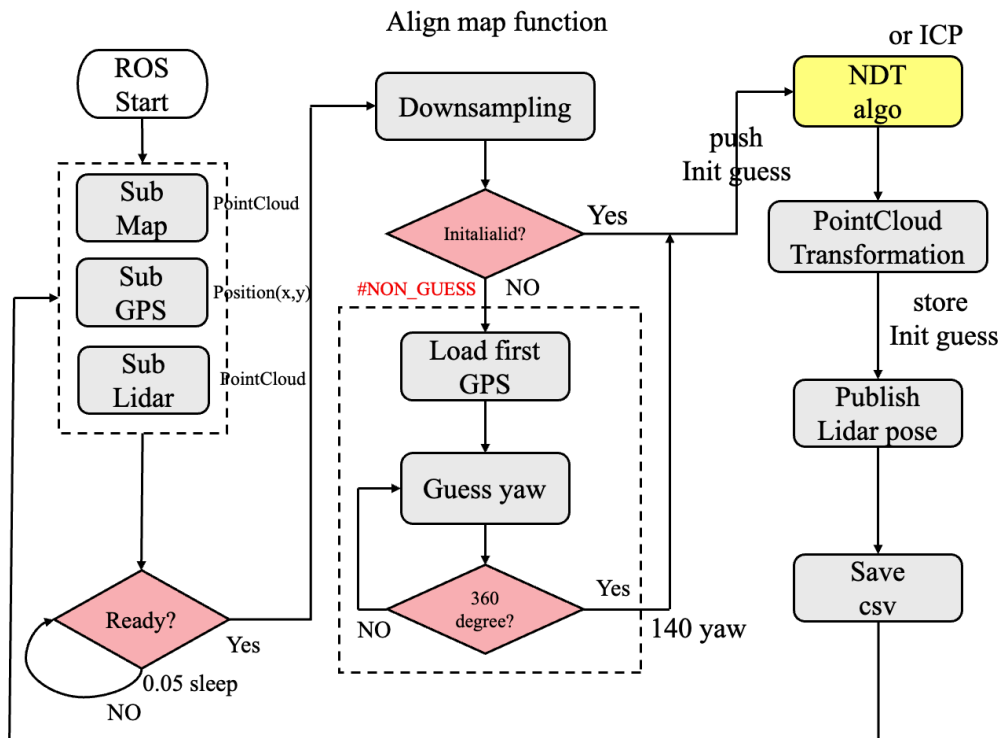


Figure 1: ITRI 流程圖

2 nuScenes2/3.bag

nuScenes2/3 可由 ITRI 基礎流程做改變，在此增加新節點 Wheel odom 及 IMU 感測資料進行融合使用 Strapdown 估測姿態矩陣及位置估測較佳的初始值提高精準度。

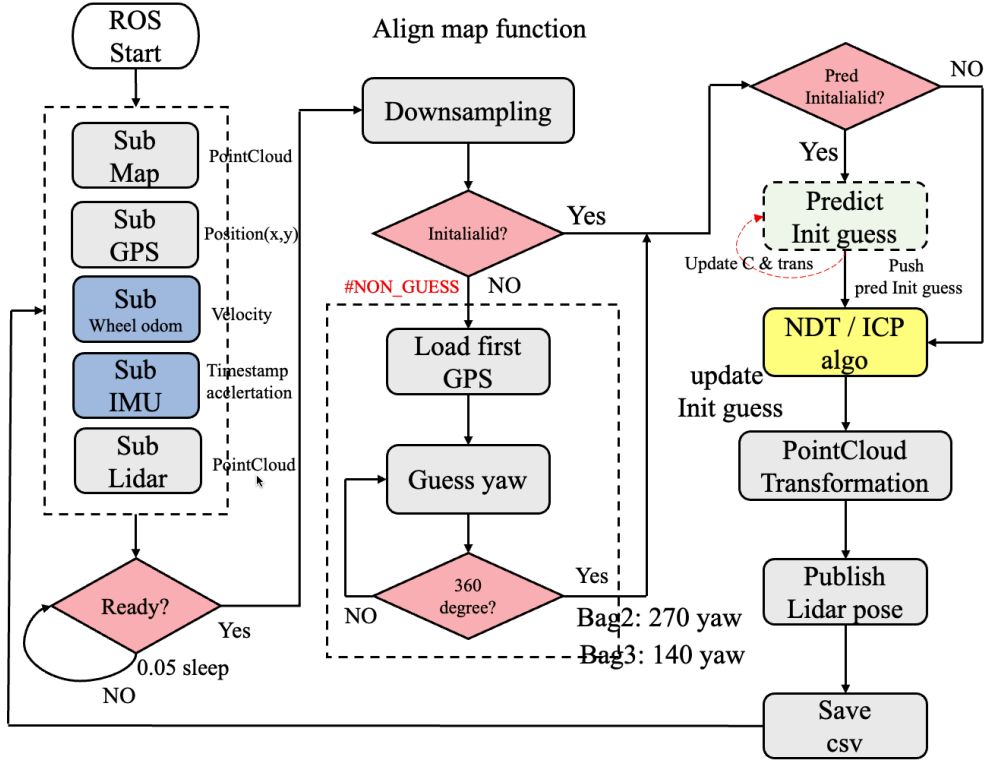


Figure 2: nuScenes 流程圖

2.1 Pred Init guess

估測方法參考文獻 [1] 中以角速度方法估測更新下一次姿態矩陣，及經由 IMU 加速度預測下一時刻位置，並融合 wheel odom 定時更新速度，降低因由 IMU 積分所造成的誤差，由估測方法可使 Init guess 更接近 ground truth 在 ICP 上有相當大的精準度提升，實際對應程式如下介紹。

2.1.1 Tracking Orientation Implementaion

由 [1] 中提及 $C(t)$ 定義為為姿態旋轉矩陣，而 IMU 提供非連續信號 ω 因此以積分採樣信號，短時距下變化 δt 可提供足夠應用而 [1] 使用 rectangular rule 計算出 B 矩陣，而根據當前角速度推測 δt 變化下，推測下一極小時刻姿態，而當中的 $\sigma = |\omega_b \delta t|$ 為角速度 norm 乘上短時間絕對值，預測姿態結果為 $C(t + \delta t)$ 計算過程如下所示。

$$C(t + \delta t) = C(t) \cdot \exp\left(\int_t^{t+\delta t} \Omega(t) dt\right) \quad (1)$$

$$\int_t^{t+\delta t} \Omega(t) dt = B \quad (2)$$

$$B = \begin{pmatrix} 0 & -\omega_{bz}\delta t & \omega_{by}\delta t \\ \omega_{bz}\delta t & 0 & -\omega_{bx}\delta t \\ -\omega_{by}\delta t & \omega_{bx}\delta t & 0 \end{pmatrix} \quad (3)$$

$$C(t + \delta t) = C(t) \left(I + B + \frac{B^2}{2!} + \frac{B^3}{3!} + \frac{B^4}{4!} + \dots \right) \quad (4)$$

$$= C(t) \left(I + B + \frac{B^2}{2!} - \frac{\sigma^2 B}{3!} - \frac{\sigma^2 B^2}{4!} + \dots \right) \quad (5)$$

$$= C(t) \left(I + \left(1 - \frac{\sigma^2}{3!} + \frac{\sigma^4}{5!} \dots \right) B + \left(\frac{1}{2!} - \frac{\sigma^2}{4!} + \frac{\sigma^4}{6!} \dots \right) B^2 \right) \quad (6)$$

$$= C(t) \left(I + \frac{\sin \sigma}{\sigma} B + \frac{1 - \cos \sigma}{\sigma^2} B^2 \right) \quad (7)$$

2.1.2 Tracking Position Implementaion

文獻 [1] 追蹤位置方法需先將 IMU 座標下加速度投影到以世界座標系統下 (8)，而可由姿態矩陣乘上加速度投影到世界座標下減去重力所造成的影響，由於積分的結果忽略重力所造成影響誤差將會累積過大，而在世界座標下後利用運動學公式推測下短時間位置 $s_g(t + \delta t)$ 。

$$a_g(t) = C(t)a_b(t) \quad (8)$$

$$v_g(t) = v_g(0) + \int_0^t a_g(t) - g_g dt \quad (9)$$

$$s_g(t) = s_g(0) + \int_0^t v_g(t) dt \quad (10)$$

$$\begin{cases} v_g(t + \delta t) = v_g(t) + \delta t \cdot (a_g(t + \delta t) - g_g) \\ s_g(t + \delta t) = s_g(t) + \delta t \cdot v_g(t + \delta t) \end{cases} \quad (11)$$

2.1.3 Implement to Code

由 [1]strapdown 預測 δt 極小時間下姿態矩陣，而初始姿態矩陣 $C(t)$ 由點雲圖匹配後計算的姿態給定，經由雲點圖計算後 Init guess 更新至 $C(0)$ 旋轉矩陣，及更新新的位置狀態降低因 IMU 積分導致的誤差累積，而更新後的 Init guess 透過 IMU 的感測資料 ω 及加速度資訊更新 $C(t + \delta t)$ ，而車輛初速度透過 wheel odom 獲得，在為更新前由 IMU 加速度持續更新速度，當新的 wheel odom velocity 更新至 v_0 ，而預測持續更新若需要做雲點圖時將當前估測作為 Init guess，由於本次忽略 z 軸上估測因此 $g = 0$ ，感測器融合關係如圖 3。

Wheel Odom Update Velocity

$C(t + \delta t)$

$a_g(t + \delta t) = C(t + \delta t) * a_b(t + \delta t)$

$v_g(t + \delta t) = v_g(t) + \delta t(a_g(t + \delta t) - g_g)$

$s_g(t + \delta t) = s_g(t) + \delta t v_g(t + \delta t)$

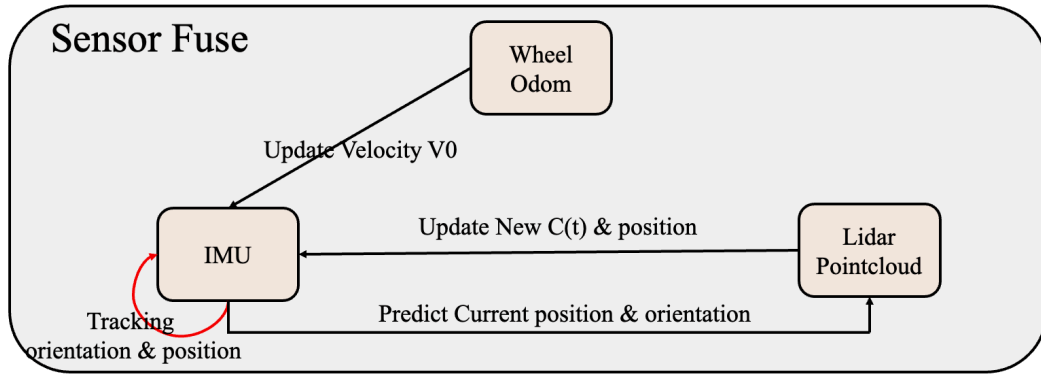


Figure 3: 感測器融合關係圖

3 Contribution

經由實際結果分析數據，本分析結果皆基於全程使用 NDT 結果視為 ground truth 比較分析 (由於無法取的的實際 ground truth 資料，而在 kaggle NDT 分數相較較低)，因此我視為最接近 ground truth 進行資料比對，根據不同 bag 分析結果如下。

3.1 ITRI result

ITRI 以全程使用 ICP 及全程 NDT 進行比較分析，分析兩者差值共分析 x,y,yaw 三軸數據，在全使用 ICP 下 kaggle score 為 0.0247 而使用 NDT 方法可降至 0.01396，結果顯示以不進行預測 Init guess 下使用 ICP 會有較大誤差，而在第一步以 ICP 下由於 Init guess 為概略位置導致起步位置誤差達 20cm，而在 120 幀時進入轉彎 ICP Init guess 若使用前一時刻進行估測位置誤差漸漸增加大約在 5cm，而在 yaw 也增加到 0.015，如圖 4 所示。

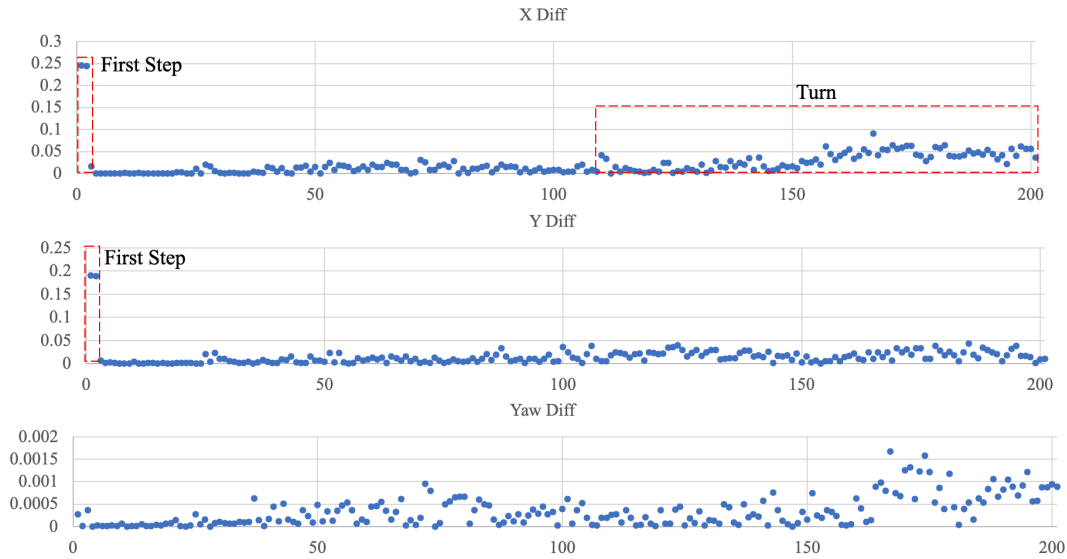


Figure 4: ITRT Bag NDT 減去 ICP 各別 x y yaw 數據分析結果

3.2 nuScenes2 result

由 ITRI 經驗可得到 ICP 方法需要較佳的 Init guess，若使用前一時刻作為 Init guess 計算出姿態及位置誤差較大，因此在此我增加 IMU 及 wheel odmo 感測資料做 tracking 使 init guess 更接近當前位置降低誤差，詳細方法如 2.1.3 所示，除此之外在 ITRI 中全程使用 NDT 方法計算時間過長，因此設計切換方法降低計算時間，第一步以 NDT 計算下一步使用 ICP 進行不斷切換，可降低一半時間且可有助於修正 ICP 結果提供更佳 Init guess, 分析結果如下所示。

3.2.1 x 軸結果分析

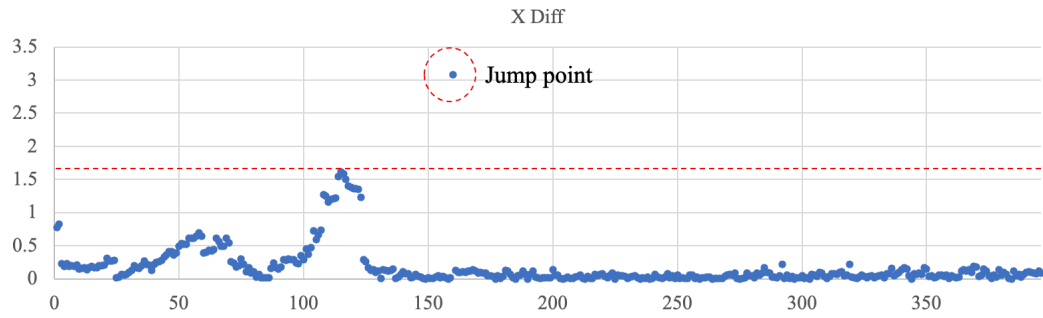


Figure 5: Bag2 NDT 減去 ICP 各別 x 軸數據分析結果

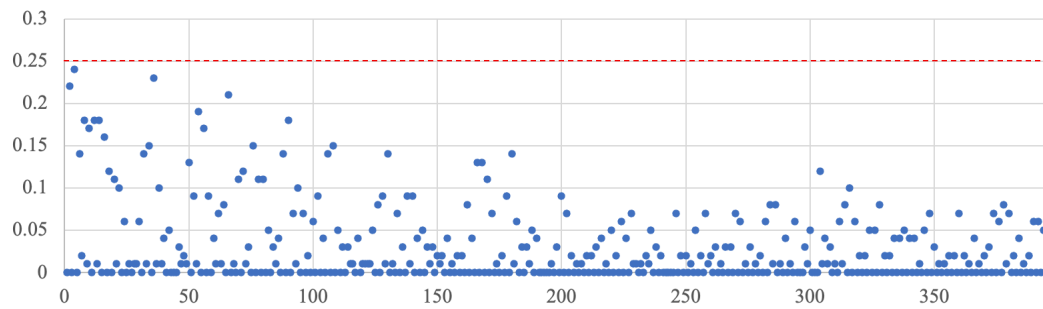


Figure 6: Bag2 NDT 減去 NDT 及 ICP 切換方法各別 x 軸數據分析結果 (驗證有助於降低誤差)

3.2.2 y 軸結果分析

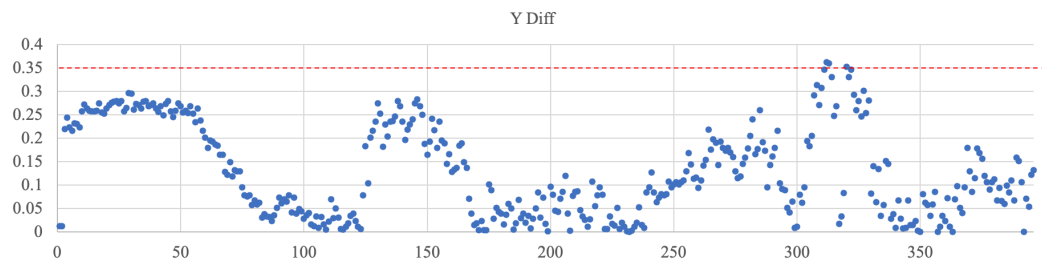


Figure 7: Bag2 NDT 減去 ICP 各別 y 軸數據分析結果

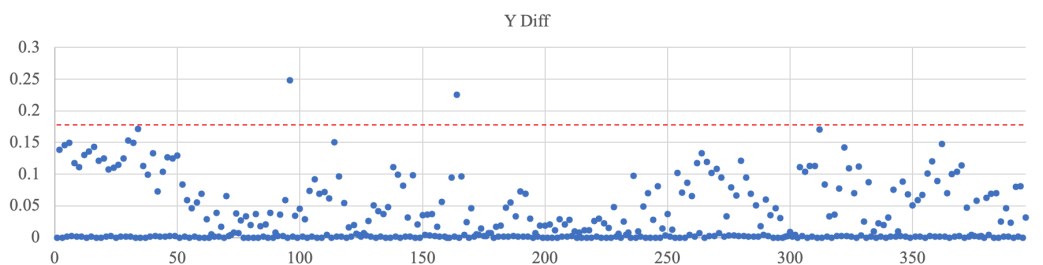


Figure 8: Bag2 NDT 減去 NDT 及 ICP 切換方法各別 y 軸數據分析結果 (驗證有助於降低誤差)

3.2.3 yaw 結果分析

由於 Bag2 為直線前進因此 yaw 無明顯差異

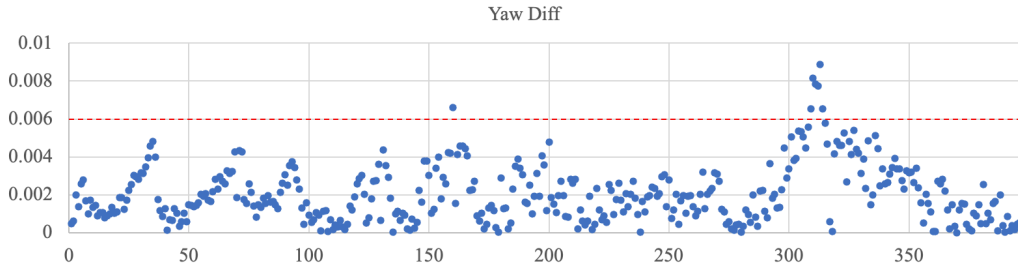


Figure 9: Bag2 NDT 減去 ICP 各別 yaw 軸數據分析結果

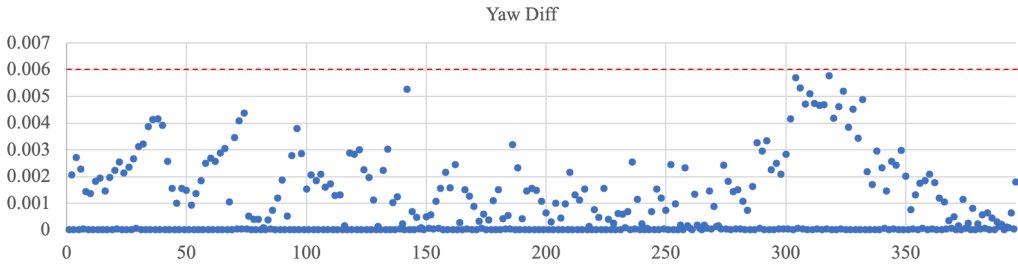


Figure 10: Bag2 NDT 減去 NDT 及 ICP 切換方法各別 yaw 軸數據分析結果 (無明顯降低趨勢)

3.3 nuScenes3 result

由於切換演算法模式精準度還是略差於全 NDT 方法，在本次並不追求效能上的優化差異最終還是使用 NDT 方法作為主要演算法，而在第一步可使用 NDT 作為第一次猜測較佳，綜合上述方法整理以下演算法比較：

- Init guess 重要性：ICP \gg NDT
- Robust：NDT \gg ICP
- 精準度：NDT $>$ ICP (Init guess is not good)
- 效能：ICP \gg NDT

由於上述方法對於 IMU 所提供的數據為原始數據，未經過磁力偏移校正等問題，在此參數轉彎處上對於 tracking 的 Init guess 不佳，因此在此 bag 使用 EKF 方法進行定位，共進行兩次濾波第一次濾波由 IMU 及 wheel odom 做感測融合後，在將經由 Lidar 的姿態轉換至車輛上與 odom filter 進行融合，將融合結果為下一次 Init guess，在此能有效改善轉彎處誤差，但 IMU orientation 無 covariance 濾波結果應該不是最佳結果圖，若要在提升精準度應該需要加入 covariance，EKF pipeline 如圖下所示，

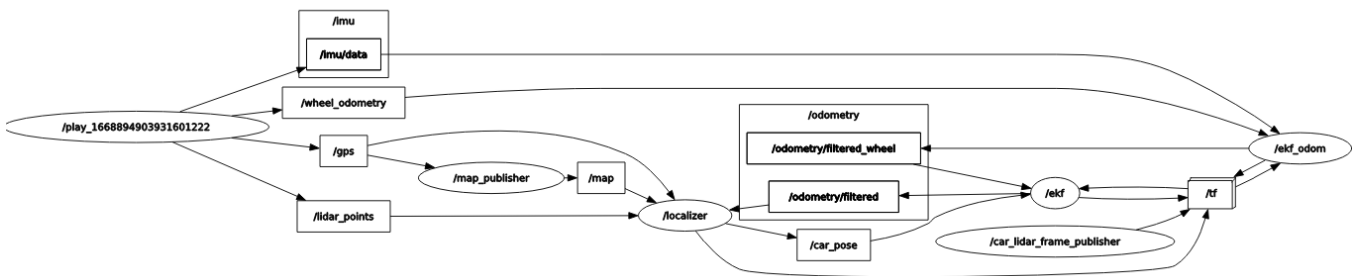


Figure 11: EKF pipeline

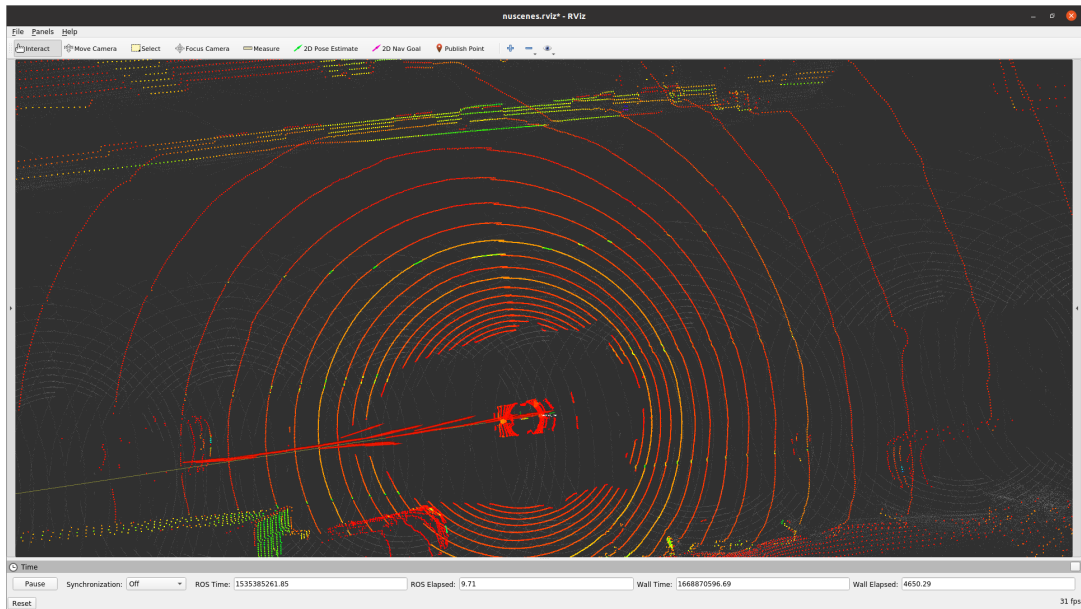


Figure 12: Orientation 濾波不佳問題

4 Problem and Solution

1. 由於採用使用 NDT 方法需要計算相當久的時間，因此在 Bag 播放時間過快會導致 timestamp 無法同步，在雲點圖計算出現跳幀現象導致數據需要重跑，因此使用 NDT 時採用播放 0.5s 讓其計算完成才依序播放 0.5s，因此耗時大概一個工作天，雖然 NDT 理論上需要較長計算時間，但計算時間超乎想像的久亦有可能是參數設定過於緊導致問題。
2. 在前幾步誤差較大原因為使用 ICP 方法需要較好的 Init guess 但剛起步是猜測的 uncertainty 較大因此在前五步可使用 NDT 作為前幾幀方法，後續經由 EKF 濾波後可使用 ICP 方法提高運行速度。

References

- [1] Labbe, R. (2014). Kalman and bayesian filters in python. Chap, 7(246), 4.