SDC Localization

Mid-term Competition

Outline

1. About Mid-term Competition
2. Data
3. Get Started for Localization
4. Kaggle Challenge
5. Supplement : ROS tutorial

1. About Mid-term Competition

Introduction

- The goal of this competition is to develop a **localization** module for estimating the poses of a self-driving car given a map.

- Datasets:
  1. ITRI dataset
  2. nuScenes dataset

- Time:

  2022/10/28 ~ 2022/11/17 23:59

Challenges

- There will be **3** scenes for the localization
  - itri bag * 1 (Easy Case)
  - nuScenes bags * 2 (Advanced Case)
    - Provide wheel_odometry for you to fuse
- Design your own localization algorithm
  - You can design it based on the ICP
  - You can use other methods (e.g. NDT)
- Your algorithm(s) need to have a better performance than our baseline (based on ICP) or you won't get any credit.
- You also need to compare your performance with others to get more credit.
- This is **NOT team competition**, everyone should compete by yourself!
  (Your team name should be your student ID)

2. Data Introduction

Download

- All the data for this competiton are saved on the google cloud directory below

  SDC Localization Competition Data

- The data contains:
  - Maps for ITRI and nuScenes
  - sdc_localization_1.bag
    - ITRI scene, please use the ITRI map
  - sdc_lovalization_2.bag and sdc_lovalization_3.bag
    - nuScenes scene, please use the nuScenes map
  - the bags' name with "lite" mean that they contain only lidar, gps, imu, tf

與我共用 > SDC

名稱 ↑	擁有者	上次修改時間	檔案大小
itri_map.pcd	我	2022年10月26日 我	80.1 MB
localization.zip	我	2022年10月27日 我	25 KB
map_tile_loader.zip	我	2022年10月27日 我	7 KB
nuscenes_maps.zip	我	2022年10月26日 我	645.2 MB
sdc_localization_1.bag	我	2022年10月26日 我	242.7 MB
sdc_localization_2_lite.bag	我	2022年10月26日 我	421.5 MB
sdc_localization_2.bag	我	2022年10月26日 我	1.73 GB
sdc_localization_3_lite.bag	我	2021年10月31日 我	414.1 MB
sdc_localization_3.bag	我	2021年10月31日 我	1.72 GB

ITRI Data - rosbag information

- sdc_localization_1.bag
  - 201 frames
  - <Important> You should play the rosbag with --clock command.

- ROS topics:
  - /lidar_points (sensor_msgs/PointCloud2) ---------- Lidar pointcloud
  - /imu/data (sensor_msgs/Imu) ---------- IMU
  - /gps (geometry_msgs/PointStamped) ---------- Simulated GPS (~10Hz)
  - /tf (tf2_msgs/TFMessage) ---------- Transformations between coordinates

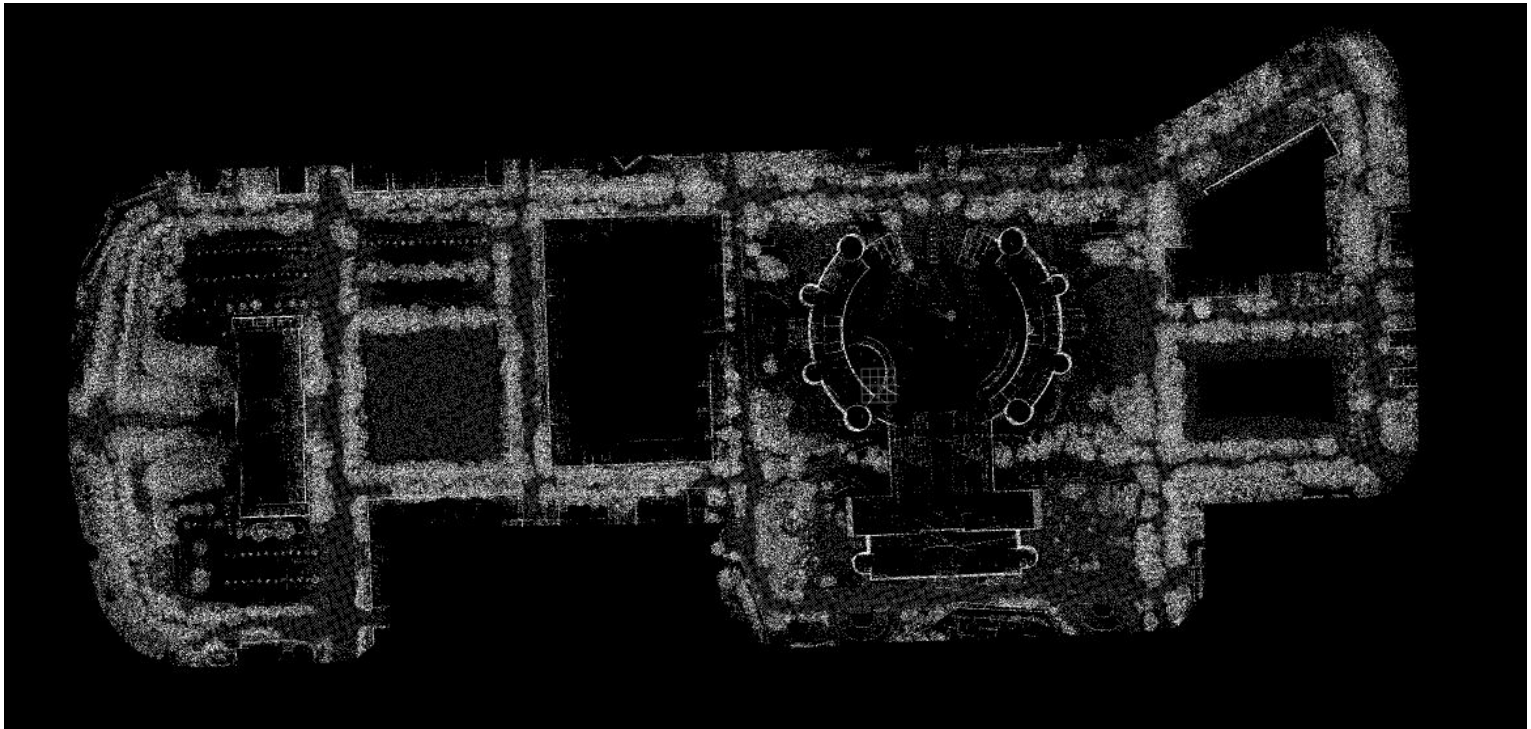
- We don’t provide the rotation initial guess this time, please find it by yourself

ITRI Data - sensor

- ITRI campus
  - LiDAR: velodyne VLP-32C
  - IMU: xsens MTi-G-710
    - orientation_covariance: [0.017453292519943295, 0.0, 0.0, 0.0, 0.017453292519943295, 0.0, 0.0, 0.0, 0.15707963267948966]
    - angular_velocity_covariance: [0.0004363323129985824, 0.0, 0.0, 0.0, 0.0004363323129985824, 0.0, 0.0, 0.0, 0.0004363323129985824]
    - linear_acceleration_covariance: [0.0004, 0.0, 0.0, 0.0, 0.0004, 0.0, 0.0, 0.0, 0.0004]
  - GPS: simulated from vehicle pose in the map frame (standard deviation of noise: 1 meter)

ITRI Data - map

- itri_map.pcd



nuScenes Data - rosbag information

- sdc_localization_2.bag and sdc_localization_3.bag from nuScenes dataset.
  - lite version: localization bag - 2 (lite) and localization bag - 3 (lite)
  - bag 2 : 396 frames, bag 3: 389 frames
  - <important> You should play the rosbag with --clock command.

- ROS topics on lite version:
  - /lidar_points (sensor_msgs/PointCloud2) ---------- Lidar pointcloud
  - /imu/data (sensor_msgs/Imu) ---------- IMU
  - /gps (geometry_msgs/PointStamped) ---------- Simulated GPS (~10Hz)
  - /tf (tf2_msgs/TFMessage) ---------- Transformations between coordinates
  - /wheel_odom (nav_msgs/Odometry) ---------- Ground truth odometry

- We don’t provide the rotation initial guess this time, please find it by yourself

nuScenes Data - additional message topics

- Camera (sensor_msgs/CompressedImage)
  - /image_back/compressed
  - /image_back_left/compressed
  - /image_back_right/compressed
  - /image_front/compressed
  - /image_front_left/compressed
  - /image_front_right/compressed

- Radar (raw data) (conti_radar/Measurement)
  - /radar_back_left
  - /radar_back_right
  - /radar_front
  - /radar_front_left
  - /radar_front_right

- Radar (points cloud) (sensor_msgs/PointCloud2)
  - /nuscenes_radar_back_left
  - /nuscenes_radar_back_right
  - /nuscenes_radar_front
  - /nuscenes_radar_front_left
  - /nuscenes_radar_front_right

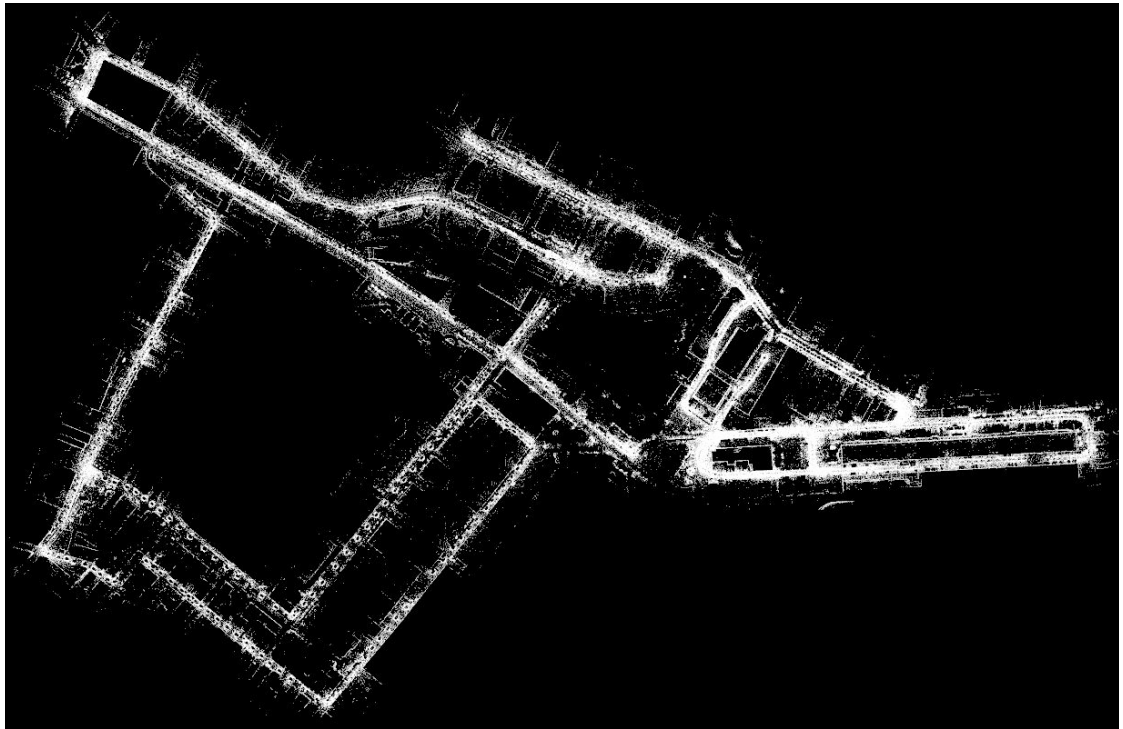
nuScenes Data - sensor

- Here is the sensor setup from nuscenes official.
  - IMU (not available now)
    - orientation_covariance: [0.01, 0.0, 0.0, 0.0, 0.01, 0.0, 0.0, 0.0, 0.04]
    - angular_velocity_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
    - linear_acceleration_covariance: [0.0025, 0.0, 0.0, 0.0, 0.0025, 0.0, 0.0, 0.0, 0.025]
    - <Important> The unit of orientation covariance and angular velocity covariance is degree. To use in EKF/UKF you should convert it to radian.

- Due to the bag size, we offer the original bag and lite bag. The lite bag only contains LiDAR, Wheel odometry, IMU, GPS and TF data which make bag smaller.

nuScenes Data - map

- nuscenes_maps/*.pcd

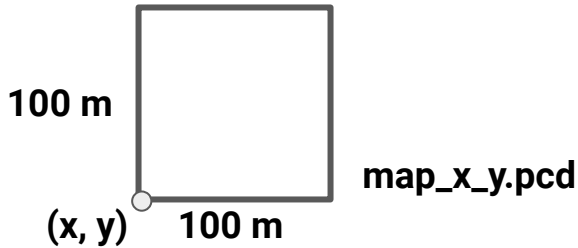


nuScenes Data - map

- Map segmentation :

  Because the map of this dataset is too large, we split whole map to many segments.

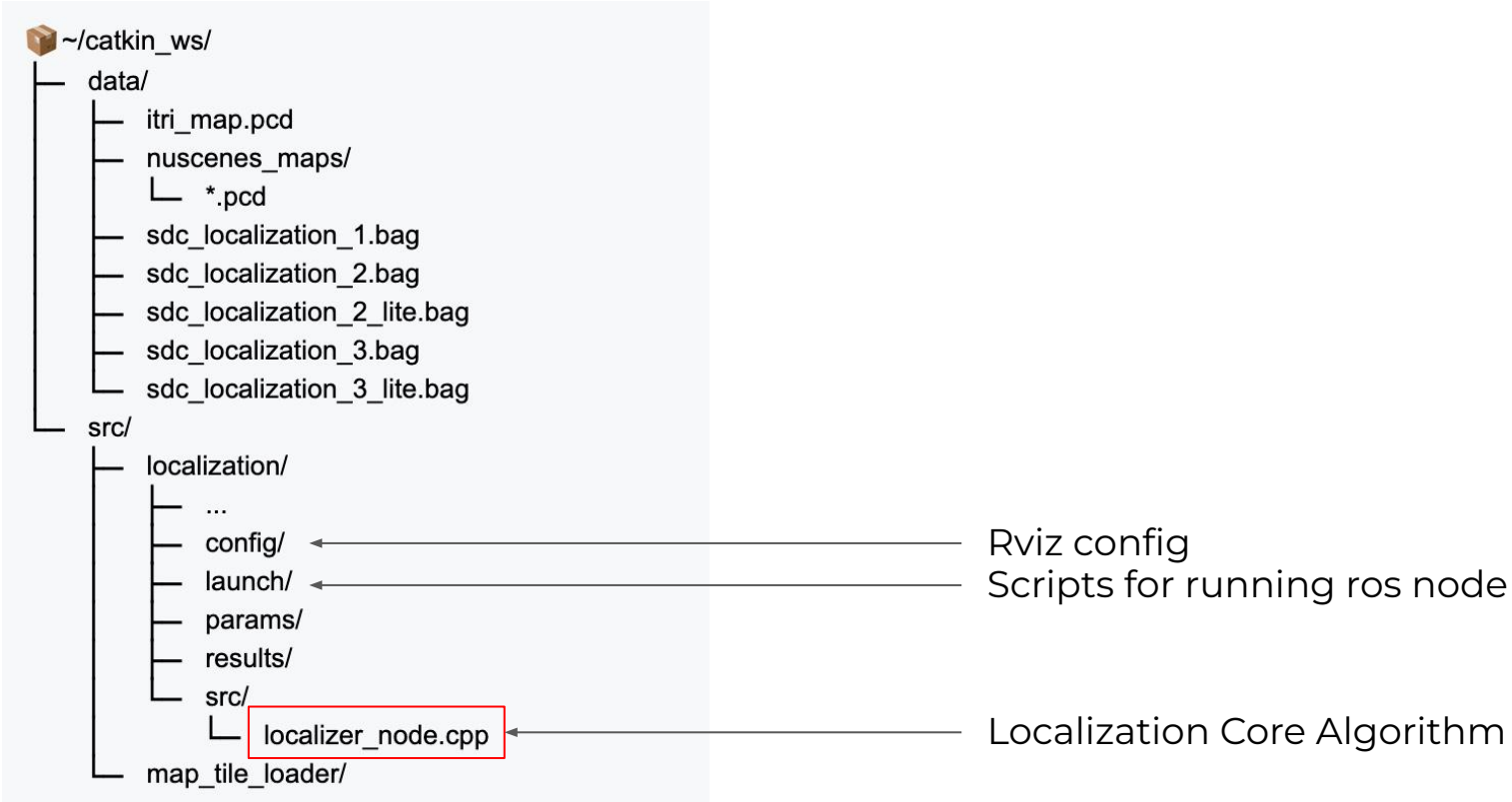
  The file naming rule is like:



- We provide both whole map and segmental map to you, you can use either for your convenience

3. Get Started

Folder structure



Environment : use Docker

In main terminal : create container and enter container

```
> xhost +local:

> docker run \
-it --gpus all --network=host \
--env="DISPLAY" \
--env="QT_X11_NO_MITSHM=1" \
--volume="/tmp/.X11-unix:/tmp/.X11-unix:rw" \
-p 2233:22 \
--rm \
--name ros \
--user root \
-e GRANT_SUDO=yes \
-v ~/catkin_ws:/root/catkin_ws \
softmac/sdc-course-docker:midterm \
bash
```

In other terminal : enter container

```
> docker exec -it ros bash
```

Run localization Code

Competition I (ITRI)

```
(container) > roslaunch localization itri.launch
```

Competition II (nuScenes)

```
(container) > roslaunch localization nuscenes.launch save_path:="/root/catkin_ws/src/localiztion/results/result_2.csv"
```

Competition III (nuScenes)

```
(container) > roslaunch localization nuscenes.launch save_path:="/root/catkin_ws/src/localiztion/results/result_3.csv"
```

Competition I : baseline result demo

Command use in video

```
(container) > roscore
```

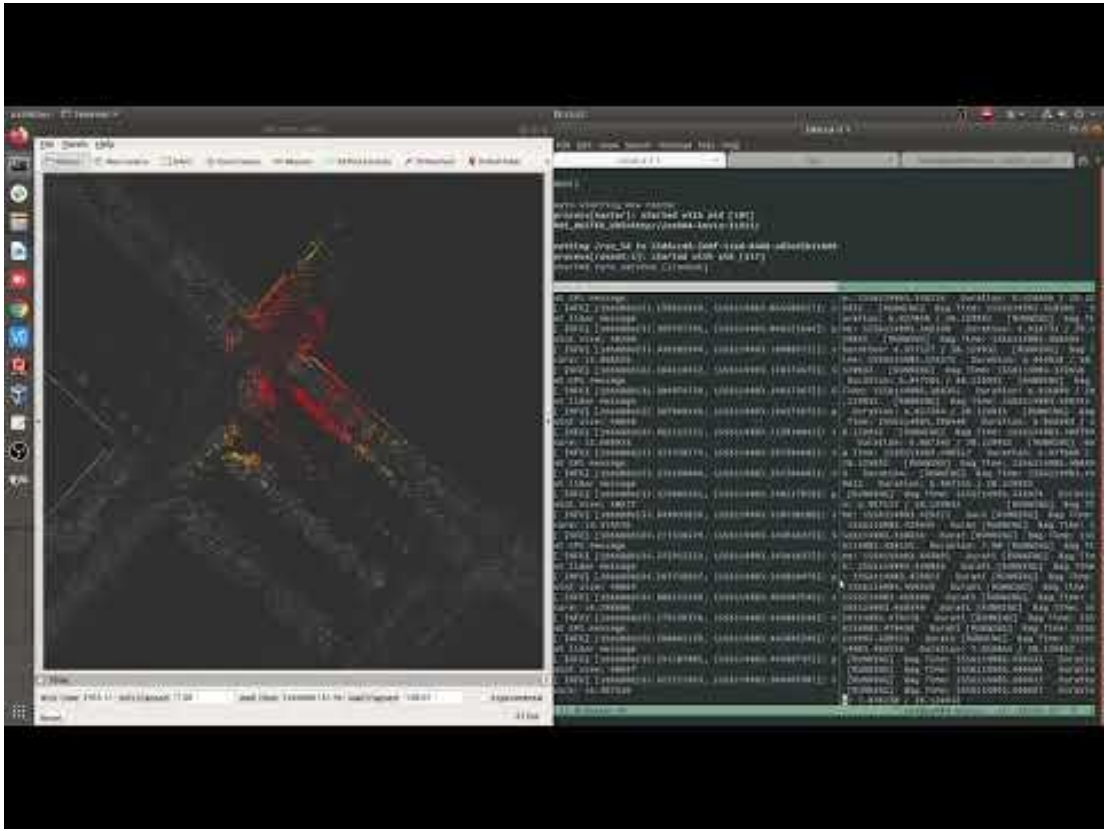
```
(container) > roslaunch localization itri.launch
```

```
> rviz
```

(optional)

```
(container) > rosbag play -r 0.1 --pause sdc_localization_1.bag --clock
```

Competition I : TA baseline demo



Inside localizer_node.cpp

```
182   Eigen::Matrix4f align_map(const pcl::PointCloud<pcl::PointXYZI>::Ptr scan_points){
183     pcl::PointCloud<pcl::PointXYZI>::Ptr filtered_scan_ptr(new pcl::PointCloud<pcl::PointXYZI>());
184     pcl::PointCloud<pcl::PointXYZI>::Ptr filtered_map_ptr(new pcl::PointCloud<pcl::PointXYZI>());
185     pcl::PointCloud<pcl::PointXYZI>::Ptr transformed_scan_ptr(new pcl::PointCloud<pcl::PointXYZI>());
186     Eigen::Matrix4f result;
187
188     /* [Part 1] Perform pointcloud preprocessing here e.g. downsampling use setLeafSize(...) ... */
189
190
191
192     /* Find the initial orientation for fist scan */
193     if(!initialied){
194       pcl::IterativeClosestPoint<pcl::PointXYZI, pcl::PointXYZI> first_icp;
195       float yaw, min_yaw, min_score = std::numeric_limits<float>::max();
196       Eigen::Matrix4f min_pose(Eigen::Matrix4f::Identity());
197           /* [Part 3] you can perform ICP several times to find a good initial guess */
198
199       // set initial guess
200       init_guess = min_pose;
201       initialied = true;
202     }
203
204       /* [Part 2] Perform ICP here or any other scan-matching algorithm */
205       /* Refer to https://pointclouds.org/documentation/classpcl_1_1_iterative_closest_point.html#details */
206
207
208
209       /* Use result as next initial guess */
210     init_guess = result;
211     return result;
212   }
213 };
```

PCL library

Point Cloud Library

https://github.com/PointCloudLibrary/pcl

https://pointclouds.org/documentation/classpcl_1_1_iterative_closest_point.html#details

Kaggle Challenge

Kaggle Results format

- The result contains the pose data of each lidar scan like:

  (id, x , y, z, yaw, pitch, roll)

  Notice that here the pose is car pose in map frame which means you need to transfer your answer from your sensor **/velodyne** to **/car**.

- Result should look like this:

id	x	y	z	yaw	pitch	roll
1	-263.7947083	-68.1448288	-9.884532929	-2.194797215	-0.035989405	0.025829997
2	-264.0217896	-68.44799805	-9.881188393	-2.190142259	-0.033838803	0.025590734
3	-264.2255859	-68.75473785	-9.87323761	-2.184811659	-0.031815543	0.023631885
4	-264.45755	-69.06990051	-9.870077133	-2.178755051	-0.030701873	0.020161531
5	-264.6669006	-69.37965393	-9.877145767	-2.172406787	-0.030660689	0.018140142

Kaggle Evaluation

- Root-Mean-Square Error (RMSE)
  - We use RMSE metrics to evaluate your localization result on the kaggle website
  - Since we have 3 bags for you to compete, we hold 3 kaggle competition:
    - 2022 SDC Localization Competition I for bag - 1 (itri)
    - 2022 SDC Localization Competition II for bag - 2 (nuScenes)
    - 2022 SDC Localization Competition III for bag - 3 (nuScenes)

- Since nuScenes dataset do not provide z-axis of ground truth, the localization competition II&III won't take z-axis into account.
  - Please modify z-axis of your localization result to 0 in competition II&III, or you would get large error due to this.

Kaggle Submission

- Please upload your localization result to the kaggle competition website.
- Maximum Daily Submissions: 20
- The result should contains coordinate data for every LiDAR timestamps.
  - 201 frames / 396 frames / 389 frames
- In the end of this competition, all teams need to upload the codes. We will check if the code can be compiled and executed.
  - Please use roslaunch to execute your program. (rosbag doesn’t need to include in roslaunch file)

Kaggle Ranking and Grading

- Competition Ranking: 60%
- Presentation and Report: 40%

Kaggle Ranking (60%)

- For each localization competition
  - 1^{st} : 20
  - 2^{nd} : 18
  - 3^{rd} : 16
  - Top 20%: 14
  - Top 40%: 12
  - Others above baseline: 10
  - Below baseline: 0
- The total score is the sum of 3 competition ranking result.
- The baseline is on the leaderboard on the kaggle website.

Report (40%)

- Your Report needs contains:
  - Pipeline (How your program works?)
  - Contribution (What's the difference between yours and others?)
  - Problem and Solution
  - Others

The contribution part depends on the idea you implement or how you solve the issue you have faced.

If you use whole open source project without your idea, you will get zero in the contribution part.

e3 Final Submission

1. localization_<student_id>.zip
    - You can make any modifications for TA sample code.
    - You need to pack up the whole localization/ folder or any code you write into .zip
    - Your best submission files of the 3 localization competition named with submit_1.csv, submit_2.csv, submit_3.csv
    - Report named with localization_report_<student_id>.pdf
2. Deadline: 11/17 23:59

Supplement

Reference

- We can use IMU sensor to calculate the path. For more information, you can reference this paper and see the chapter 6.1 and 6.2 for implementation details.

Supplement

ROS

- frames & TF
- roslaunch & launch file
- rviz
- Debug tool: rqt, rosnode, rostopic

PCL library

- data structure
- type conversion
- icp