ME5413: Autonomous Mobile Robotics

Homework 2: SLAM

AY2023/24-Sem 2

Due: 23:59 Sunday, 14 March 2024

Introduction: The aim of this homework is to get you familiarised with the knowledge taught in class and evaluate popular open-sourced SLAM algorithms. After trying out those algorithms, you should be able to appreciate the real-world challenges in SLAM and the pros and cons of each algorithm. These SLAM algorithms will also be used in the final project.

<u>Prerequisite Knowledge</u>

1. Commonly used Evaluation Metrics of SLAM

SLAM performs both localization and mapping at the same time, while we are more interested in the localization performance because it stands for the actual accuracy of ego pose estimation. There are 2 popular metrics that always been used when evaluating the performance of localization.

- Absolute Position Error (APE)
  APE measures the absolute position error. Given that $P_{est}(t)$ is the estimated position by SLAM algorithm at time t, and $P_{gt}(t)$ is the corresponding ground truth position. APE is defined as:
  $$APE(t) = \left\| P_{est}(t) - P_{gt}(t) \right\|$$
- Relative Position Error (RPE)
  RPE measures the local pose error over a specified time interval. Given 2 timestamps t_i and t_j. $\delta p_{est}(t_i, t_j)$ and $\delta p_{gt}(t_i, t_j)$ are the relative translation of estimation and ground truth between t_i and t_j, respectively. RPE is defined as:
  $$RPE(t_i, t_j) = \left\| \delta p_{est}(t_i, t_j) - \delta p_{gt}(t_i, t_j) \right\|$$

Please note:

- Rotation is not considered here because the error of rotation will directly cause the error of translation, thus considering translation is enough at the most time.
- Definition of above metrics only calculate 1 error at certain timestamp. We often use the RMSE of all errors over the entire trajectory to evaluate performance of localization. Suppose we have totally N localization data over the entire sequence, the RMSE of APE or RPE are calculated as:

$$RMSE\ of\ APE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(APE(t))^2}$$

$$RMSE\ of\ RPE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(RPE(t_i, t_j))^2}$$

2. Commonly used Format to Save Odometry Data

We will introduce 2 popular format that saves the localization result given by SLAM.

- KITTI format

KITTI is the most used SLAM benchmark created by Karlsruhe Institute of Technology (KIT) in 2012:https://www.cvlibs.net/datasets/kitti/eval_odometry.php

The pose is saved as the following figure shows:

```
1.000000e+00 1.197625e-11 1.704638e-10 1.110223e-16 1.197625e-11 1.000000e+00 3.562503e-10 0.000000e+00 1.704638e-10 3.562503e-10 1.000000e+00 2.220446e-16
9.999968e-01 -1.958988e-03 -1.630129e-03 3.499723e-03 1.957401e-03 9.999976e-01 -9.745057e-04 -9.789328e-03 1.632035e-03 9.713123e-04 9.999982e-01 5.653511e-01
9.999931e-01 -2.475044e-03 -2.755837e-03 1.173954e-03 2.468662e-03 9.999942e-01 -2.316828e-03 -2.241474e-02 2.761556e-03 2.310009e-03 9.999935e-01 1.127607e+00
9.999863e-01 -2.528164e-03 -4.580897e-03 -7.831705e-03 2.512020e-03 9.999906e-01 -3.526615e-03 -4.141091e-02 4.589771e-03 3.515059e-03 9.999833e-01 1.689225e+00
9.999675e-01 -4.648687e-03 -6.595320e-03 -1.508128e-02 4.612815e-03 9.999745e-01 -5.443974e-03 -5.922720e-02 6.620460e-03 5.413373e-03 9.999634e-01 2.251296e+00
9.999413e-01 -7.134576e-03 -8.152763e-03 -2.040073e-02 7.072901e-03 9.999463e-01 -7.569121e-03 -7.423248e-02 8.206328e-03 7.511012e-03 9.999381e-01 2.816072e+00
9.999200e-01 -8.311338e-03 -9.539068e-03 -2.969353e-02 8.230327e-03 9.999300e-01 -8.500812e-03 -9.078265e-02 9.609054e-03 8.421622e-03 9.999184e-01 3.390281e+00
9.999041e-01 -8.525348e-03 -1.091073e-02 -4.152744e-02 8.441127e-03 9.999344e-01 -7.742225e-03 -1.098544e-01 1.097602e-02 7.649383e-03 9.999105e-01 3.969489e+00
9.998884e-01 -8.811524e-03 -1.206613e-02 -5.593704e-02 8.715868e-03 9.999303e-01 -7.957589e-03 -1.348021e-01 1.213540e-02 7.851533e-03 9.998955e-01 4.556204e+00
9.998780e-01 -8.278892e-03 -1.324786e-02 -7.008098e-02 8.141373e-03 9.999127e-01 -1.040105e-02 -1.575514e-01 1.333281e-02 1.029192e-02 9.998582e-01 5.150463e+00
9.998748e-01 -6.580037e-03 -1.439059e-02 -8.798198e-02 6.374276e-03 9.998774e-01 -1.429781e-02 -1.803479e-01 1.448291e-02 1.420429e-02 9.997942e-01 5.751285e+00
9.998652e-01 -5.082827e-03 -1.561247e-02 -1.054713e-01 4.800847e-03 9.998256e-01 -1.804597e-02 -1.996581e-01 1.570147e-02 1.796859e-02 9.997153e-01 6.363942e+00
9.998522e-01 -3.623834e-03 -1.680538e-02 -1.238576e-01 3.306955e-03 9.998169e-01 -1.884543e-02 -2.187844e-01 1.687060e-02 1.878706e-02 9.996812e-01 6.989627e+00
9.998394e-01 -1.112536e-03 -1.788542e-02 -1.444743e-01 8.907687e-04 9.999227e-01 -1.240253e-02 -2.366165e-01 1.789783e-02 1.238460e-02 9.997631e-01 7.637602e+00
9.998258e-01 -8.672136e-05 -1.866531e-02 -1.645124e-01 5.358731e-05 9.999984e-01 -1.775663e-03 -2.542635e-01 1.866543e-02 1.774354e-03 9.998242e-01 8.302111e+00
9.998120e-01 1.004604e-03 -1.936675e-02 -1.844755e-01 -8.989153e-04 9.999846e-01 5.465141e-03 -2.738044e-01 1.937194e-02 -5.446702e-03 9.997975e-01 8.978383e+00
9.997817e-01 3.389970e-03 -2.061746e-02 -2.077191e-01 -3.275843e-03 9.999791e-01 5.566728e-03 -2.924369e-01 2.063590e-02 -5.497971e-03 9.997719e-01 9.661917e+00
9.997543e-01 5.555763e-03 -2.145926e-02 -2.329992e-01 -5.595211e-03 9.999827e-01 -1.778629e-03 -3.035717e-01 2.144901e-02 1.898261e-03 9.997682e-01 1.035368e+01
9.997369e-01 4.812930e-03 -2.242765e-02 -2.537940e-01 -5.013394e-03 9.999479e-01 -8.890577e-03 -3.091782e-01 2.238369e-02 9.000675e-03 9.997089e-01 1.106423e+01
9.997145e-01 2.620743e-03 -2.375049e-02 -2.701280e-01 -2.819209e-03 9.999613e-01 -8.326642e-03 -3.152238e-01 2.372775e-02 8.391221e-03 9.996832e-01 1.179839e+01
9.997063e-01 1.893748e-03 -2.416094e-02 -2.821162e-01 -1.858131e-03 9.999971e-01 1.496504e-03 -3.207015e-01 2.416371e-02 -1.451169e-03 9.997070e-01 1.255870e+01
9.996966e-01 1.625146e-03 -2.457882e-02 -2.962122e-01 -1.393167e-03 9.999543e-01 9.452361e-03 -3.252216e-01 2.459306e-02 -9.415248e-03 9.996532e-01 1.333520e+01
9.996863e-01 7.583641e-04 -2.503532e-02 -3.128296e-01 -4.521018e-04 9.999250e-01 1.223661e-02 -3.292075e-01 2.504272e-02 -1.222145e-02 9.996117e-01 1.412399e+01
9.996792e-01 -1.382699e-03 -2.529212e-02 -3.270868e-01 1.667549e-03 9.999354e-01 1.124476e-02 -3.300205e-01 2.527494e-02 -1.128333e-02 9.996169e-01 1.492763e+01
9.996851e-01 -2.198699e-03 -2.499779e-02 -3.451203e-01 2.413900e-03 9.999602e-01 8.581850e-03 -3.298600e-01 2.497793e-02 -8.639487e-03 9.996507e-01 1.574882e+01
9.997041e-01 5.747479e-05 -2.432655e-02 -3.585571e-01 6.660546e-05 9.999870e-01 5.099768e-03 -3.310784e-01 2.432653e-02 -5.099877e-03 9.996911e-01 1.658175e+01
9.997240e-01 -2.636624e-03 -2.334610e-02 -3.661096e-01 2.632565e-03 9.999965e-01 -2.046126e-04 -3.374209e-01 2.334656e-02 1.430967e-04 9.997274e-01 1.742810e+01
9.997263e-01 -5.544119e-03 -2.272786e-02 -3.686442e-01 5.399776e-03 9.999649e-01 -6.407463e-03 -3.506532e-01 2.276259e-02 6.282984e-03 9.997212e-01 1.828806e+01
9.997553e-01 -5.085930e-03 -2.153036e-02 -3.812873e-01 4.819452e-03 9.999113e-01 -1.241078e-02 -3.643232e-01 2.159157e-02 1.230397e-02 9.996912e-01 1.916967e+01
9.997662e-01 -3.838836e-03 -2.127843e-02 -3.931436e-01 3.555293e-03 9.999046e-01 -1.334729e-02 -3.836280e-01 2.132764e-02 1.326852e-02 9.996845e-01 2.007060e+01
```

Each row stands for 1 pose w.r.t. world coordinate with totally 12 values. From the left to right, they are the first 3 rows in the homogeneous transformation matrix. I.e., if consider data in a row v0 ~ v11, the transformation matrix would be:

$$T = \begin{bmatrix} v0 & v1 & v2 & v3 \\ v4 & v5 & v6 & v7 \\ v8 & v9 & v10 & v11 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

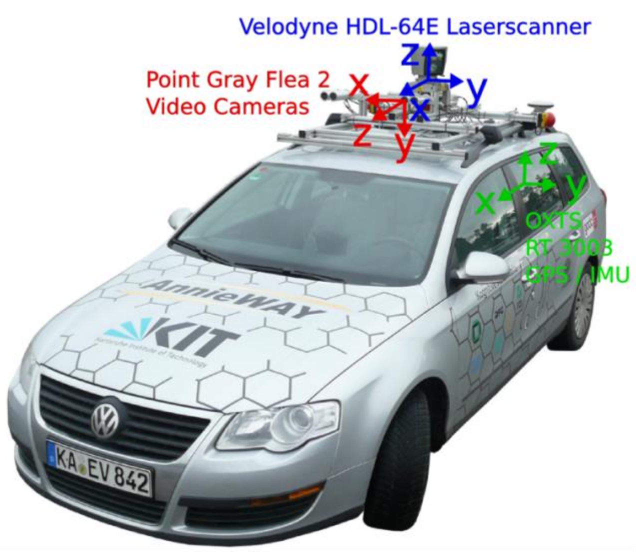
- TUM format
  Computer Vision Group of The Technical University of Munich (TUM) has proposed several SLAM datasets and the ground truth pose is saved as the following format:

```
# mocap poses: time(us) px py pz qx qy qz qw
1.079726972999572754e+06 -4.460786581039428711e-01 2.074798047542572021e-01 1.101052522659301758e+00 2.269580215215677432e-02 9.129061549901959505e-02 -1.637333631515502930e-01 -9.820093512535094105e-01
1.088060972999572754e+06 -4.457169473171234131e-01 2.071367800235748291e-01 1.101303696632385254e+00 2.247850224375713690e-02 9.112996608018869571e-02 -1.635881960391998291e-01 -9.820535182952879749e-01
1.096393972999572754e+06 -4.452580511569976807e-01 2.067252695560455322e-01 1.101626634597778320e+00 2.283553220331668854e-02 9.166159480810164017e-02 -1.634240597486495972e-01 -9.820231199264526367e-01
1.104726972999572754e+06 -4.448015689849853516e-01 2.062993049621582031e-01 1.102000713348388672e+00 2.306929603219032288e-02 9.211968630552289095e-02 -1.632729023694992065e-01 -9.819998741149902344e-01
1.113059972999572754e+06 -4.443445801734924316e-01 2.058703601360321045e-01 1.102428078651428223e+00 2.326454780995845795e-02 9.252475202083586259e-02 -1.631019264459609985e-01 -9.819856882095336914e-01
1.121393972999572754e+06 -4.439297914505004883e-01 2.054362446069717407e-01 1.102889060974121094e+00 2.286909148097027167e-02 9.214881062507625231e-02 -1.628354936838150024e-01 -9.820744991302489124e-01
1.129726972999572754e+06 -4.434683024883270264e-01 2.049844712018966675e-01 1.103419780731201172e+00 2.250417321920400449e-02 9.177161753177640047e-02 -1.625385731458663940e-01 -9.821674227714538574e-01
1.138060972999572754e+06 -4.429821074008941650e-01 2.045129537582397461e-01 1.104005694389343262e+00 2.219199202954763583e-02 9.152916073799130525e-02 -1.622220659255598145e-01 -9.822494983673096813e-01
1.146393972999572754e+06 -4.424700438976287842e-01 2.040316313505172729e-01 1.104646325111389160e+00 2.195614017546176910e-02 9.130321443080899324e-02 -1.618712395429611206e-01 -9.823337197303771973e-01
1.154726972999572754e+06 -4.419224858283996582e-01 2.035397440195083618e-01 1.105353951454162598e+00 2.183155901730066129e-02 9.117845445871350374e-02 -1.614091247320175171e-01 -9.824241399765014648e-01
1.163060972999572754e+06 -4.413415789604187012e-01 2.030336558818817139e-01 1.106105923652648926e+00 2.176023274660116025e-02 9.103868156671519885e-02 -1.608176380395889282e-01 -9.825356006622314453e-01
1.171393972999572754e+06 -4.407398987373335205e-01 2.025247961282730103e-01 1.106921672821044922e+00 2.171898633241653442e-02 9.080920368432994494e-02 -1.601556390523910522e-01 -9.826658964157104492e-01
1.179726972999572754e+06 -4.401119947433471680e-01 2.020191550254821777e-01 1.107778906822204590e+00 2.170297317206854038e-02 9.073738008737561311e-02 -1.594922095537185669e-01 -9.827806949615478516e-01
1.188059972999572754e+06 -4.394069015979766846e-01 2.014944851398468018e-01 1.108677625656127930e+00 2.186343818902969360e-02 9.062135964632030138e-02 -1.586204618215560913e-01 -9.829290509223937988e-01
1.196393972999572754e+06 -4.386878311634063721e-01 2.009950727224349976e-01 1.109633684158325195e+00 2.181023545563215427e-02 9.040065854787822375e-02 -1.577388197183609009e-01 -9.830923676490783691e-01
1.204726972999572754e+06 -4.379289746284484863e-01 2.005181461572647095e-01 1.110639572143554688e+00 2.186471410095686130e-02 9.008179605007170243e-02 -1.567436307668686191e-01 -9.832795262336728737e-01
1.213060972999572754e+06 -4.371254742145538330e-01 2.000606507062911987e-01 1.111659884452819824e+00 2.194842696189869269e-02 8.977594226598735461e-02 -1.557226777076721191e-01 -9.834678173065184437e-01
1.221393972999572754e+06 -4.362567961215972900e-01 1.995828002691268921e-01 1.112688302993774414e+00 2.227940782904630490e-02 8.974616974592206087e-02 -1.548020094633102417e-01 -9.836084842681884766e-01
1.229726972999572754e+06 -4.354312419891357422e-01 1.991521865129470825e-01 1.113768577575683594e+00 2.256415039300918579e-02 8.932127803564068880e-02 -1.537606418132781982e-01 -9.838040471076965332e-01
1.238059972999572754e+06 -4.345148503780364990e-01 1.987321227788925171e-01 1.114839076995849609e+00 2.280841208994388580e-02 8.914905041456218371e-02 -1.528153717517852783e-01 -9.839612841606140137e-01
1.246393972999572754e+06 -4.336457848548889160e-01 1.983416378498077393e-01 1.115964293479919434e+00 2.300428412854677029e-02 8.871012926101681795e-02 -1.518432497978210449e-01 -9.841468930244445801e-01
1.254726972999572754e+06 -4.327254295349121094e-01 1.979936510324478149e-01 1.117101669311523438e+00 2.305142767727375031e-02 8.818843215703961458e-02 -1.508419662714004517e-01 -9.843465089797972523e-01
1.263060972999572754e+06 -4.318259954452514648e-01 1.976770311594009399e-01 1.118264794349670410e+00 2.326453104615211487e-02 8.777831494808194246e-02 -1.499223113059997559e-01 -9.845187067985534668e-01
1.271393972999572754e+06 -4.309252500534057617e-01 1.974221915006637573e-01 1.119422554969787598e+00 2.336310781538492032e-02 8.737465739250181718e-02 -1.490784734487533847e-01 -9.846804141998289905e-01
1.279726972999572754e+06 -4.300123453140258789e-01 1.971915811300277710e-01 1.120597243309020996e+00 2.360894903540616818e-02 8.700986951589581575e-02 -1.483134031295776645e-01 -9.848223924636839710e-01
1.288060972999572754e+06 -4.290879964828491211e-01 1.970121860504150391e-01 1.121776103973388672e+00 2.379963919520378113e-02 8.664593100547787752e-02 -1.476185768842697144e-01 -9.849542379379272461e-01
1.296393972999572754e+06 -4.281323552131652832e-01 1.968863904476165771e-01 1.122931241989135742e+00 2.386083453893661499e-02 8.621776849031444201e-02 -1.470815986394882202e-01 -9.850707054138183594e-01
1.304726972999572754e+06 -4.271509647369384766e-01 1.967905908823013306e-01 1.124059677124023438e+00 2.407484874129300900e-02 8.607356250286099519e-02 -1.466743350028991699e-01 -9.851387739181517444e-01
1.313059972999572754e+06 -4.262389838695526123e-01 1.967308372259140015e-01 1.125203728675842285e+00 2.412013150751590729e-02 8.592325448989865389e-02 -1.465092003345489502e-01 -9.851753711700439453e-01
```

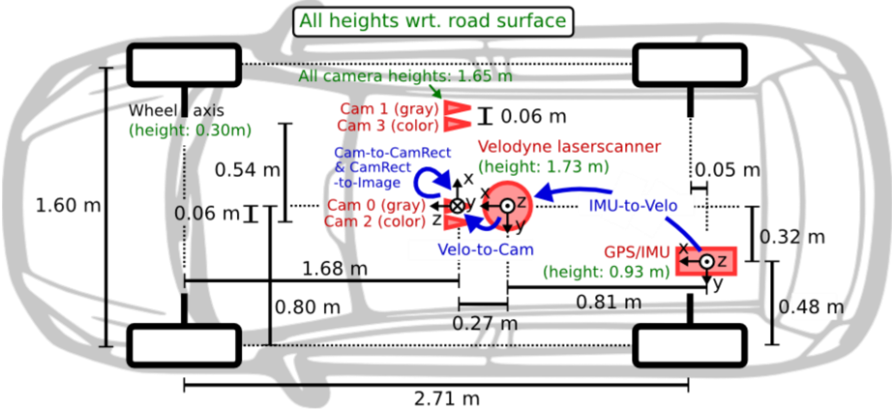
  - Each line in the text file contains a single pose.
  - The format of each line is '**timestamp tx ty tz qx qy qz qw**'
  - **timestamp** (float) gives the number of seconds since the Unix epoch.
  - **tx ty tz** (3 floats) give the position of the optical center of the color camera with respect to the world origin as defined by the motion capture system.
  - **qx qy qz qw** (4 floats) give the orientation of the optical center of the color camera in form of a unit quaternion with respect to the world origin as defined by the motion capture system.
  - The file may contain comments that have to start with "#".

3. Coordinate Transformation in SLAM evaluation

The coordinates systems may be different between the SLAM algorithm and the ground truth. For example, ground truth of KITTI dataset is recorded in camera frame. If we use LiDAR data of KITTI, we should transform the poses given by LIDAR SLAM to camera frame before performing evaluation:



The bag we provide in Task 2 is part of KITTI dataset, the extrinsic transformation between sensors is given by:



Please note, by default, ROS follows the right-hand coordinate convention:

- x forward
- y left
- z up

But for the output of SLAM algorithm, you need to check the corresponding repository page.

Task 1: Running Cartographer on the given ROS bag

`<2dlidar.bag>` records the message generated by a mobile robot moving in a static environment. In this task, you are expected to try out Cartographer (https://google-cartographer-ros.readthedocs.io/en/latest/index.html) on the given ROS bag (`<2dlidar.bag>`) and save the .pgm format map built by Cartographer.

Bonus:

We suggest you use EVO (mentioned in task2) to test the performance of the Cartographer.

1. Ground truth of the robot's trajectory is stored in the `<2dlidar.bag>` itself in topic `/odom`
2. Refer to the following command to use EVO: `evo_ape bag <bag_file_name> /odom /tf:map.base_footprint –plot`

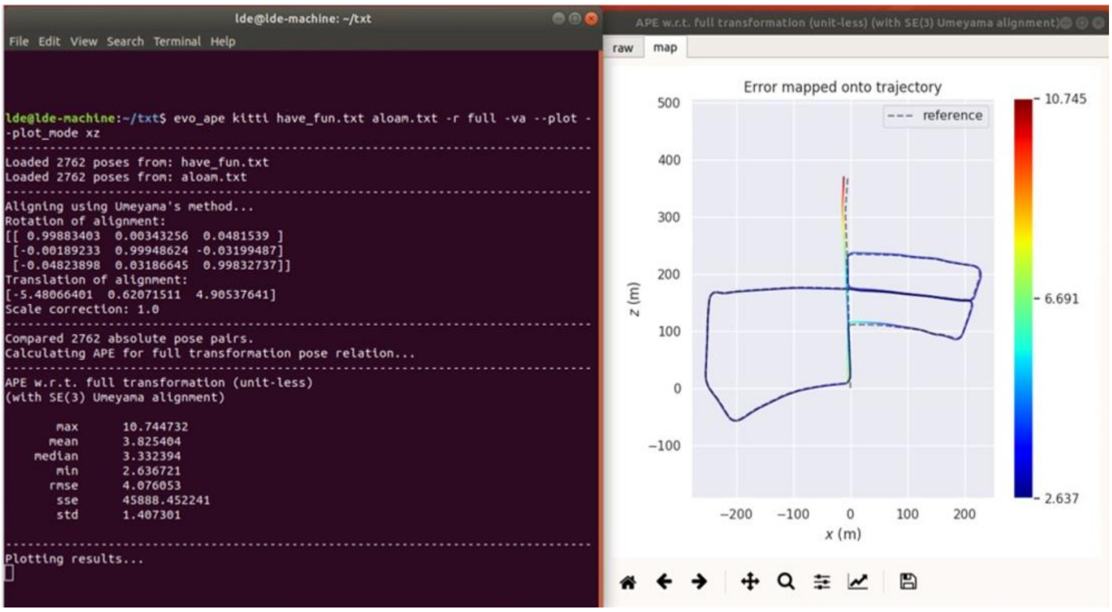
Task 2: Running 3D SLAM algorithms on the given ROS bag

Choose some SLAM algorithms to run the `<have_fun.bag>` and evaluate the performance of your SLAM algorithms by the EVO tool (https://github.com/MichaelGrupp/evo). The performance of the algorithms are assessed based on the Absolute Root Mean Square Error (RMSE) over the entire sequence. There is no limit on what SLAM algorithms you use, your goal is minimizing the RMSE on the given sequence, and the Ground truth of the `<have_fun.bag>` is the `have_fun.txt` that saved in KITTI format. You are required to use at least two SLAM algorithms and one of them is pure LIDAR SLAM algorithm. We encourage you to use pure Lidar SLAM algorithms like ALOAM, LeGO-LOAM, etc., or multi-sensor fusion SLAM algorithms like VINS, FAST-LIO, etc.

The pipeline you should complete is:

- Test your SLAM algorithms on the given bag and record the odometry output as a bag.
- Read the recorded bag and get the data of odometry (you can use bagpy, Matlab or any open-sourced rosbag processing package).
- Analyze the coordinate system and transform the odometry data to camera frame (ground truth is recorded in camera system). Save it in TUM or KITTI format:
    - `evo_ape have_fun_tum.txt <my_result>.txt -r full --plot --plot_mode xz`
    - `evo_ape have_fun_kitti.txt <my_result>.txt -r full --plot --plot_mode xz`

Please attach the figure like the following one to show your result in the report.



In your report (for both Task 1 and 2):

- Briefly describe the SLAM algorithm you use
- Describe the detailed process of running the SLAM algorithms, give a screenshot of your algorithm running in Rviz
- Highlight your modification/tuning done on the algorithms to achieve better results
- Show the Absolute RMSE, as well as the plots generated by the EVO tool
- Discuss the drawbacks/failures in your tests, provide some analysis and propose possible solutions (illustrate with figures and provide your hypothesis)

Submitting your completed Homework Assignment:

You are expected to summarise your observations, assumptions, and your own implementation details in a 5-page report (the maximum length of the report is 5 pages, while there is no page limit for appendices, if you wish to attach more of your results, but only images and tables can be put in appendices).

Do note that you should practice good code styles in your own code, including proper naming conventions, informative documentations, etc (please refer to the Google Python Style Guide).

Generate a non-password-protected zipfile of this folder and upload it to CANVAS – under Homework 2

We will use the latest version, regardless of who uploads. Name of Zipfile: "YourHomeWorkGroupNumber_Homework2.zip" (e.g., 43_Homework2.zip - for group 43)

Late submissions will be penalised.

1. Report details:
   a. Homework Group number
   b. Matric numbers of group members (e.g. A0123456X.)
   c. Maximum number of **5 pages for the report. There's no limit on pages for appendix**, **but only figures and tables are allowed in appendix.**
2. All code and results are to be stored in the following three folders

   Task 1:
   - The .pgm format map generated by Cartographer
   - Images generated by "evo_ape" command (if you have)

   Task 2:
   - The code you wrote to perform coordinate transformation.
   - The .pcd format map generated by your chosen SLAM algorithms
   - Images generated by "evo_ape" command
3. Evaluation of your homework will be based on
   a. Technical explanations
   b. Code executability
   c. Performance/Accuracy
   d. Report
   e. Peer review