Run EVO tool on KITTI sequences

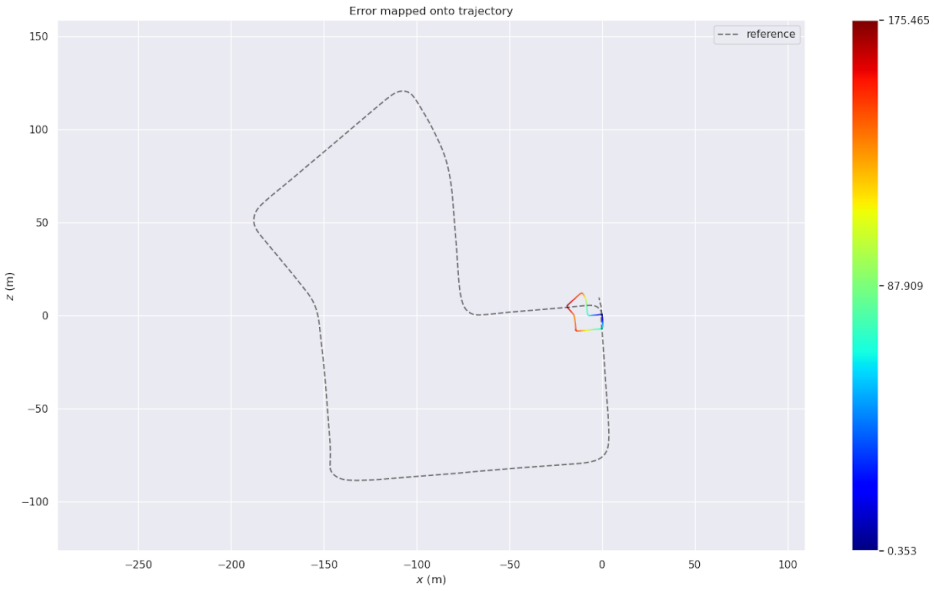
In the second coursework, you will evaluate the SLAM system using the EVO tool. The installation is very straightforward, and you can install via pip directly. They also provided a detailed wiki page on their GitHub:
https://github.com/MichaelGrupp/evo
https://github.com/MichaelGrupp/evo/wiki

You also have access to some useful scripts here:
https://github.com/UCL/COMP0249_24-25_ORB_SLAM2.git

~~Or here:~~
~~https://github.com/UCL/COMP0130_22-23_Topic_03/tree/main/Coursework_03~~

This section will give you an example on how to evaluate the camera trajectory estimation from monocular ORB-SLAM2 on KITTI sequence using the EVO tool. Before we dive into the codes and commands, it is important to understand different error metrics used for evaluating a SLAM system. The most common metrics are **Absolute Trajectory Error** (ATE) and **Relative Pose Error** (RPE). For detailed definitions you can refer to the paper of TUM dataset [1]. The KITTI dataset also has its own defined metrics [2] but the ideas behind them are similar. In this coursework, we will ask you to use **ATE** only for evaluation, but you are welcome to explore other metrics if you are interested.

When comparing the estimated camera poses and the ground-truth poses, the first step is to align them to the same coordinate system. For stereo and depth systems, this could be done via solving a 6-DoF transformation between the two trajectories (similar idea as ICP). For monocular systems, 7-DoF alignment (pose + scale) is required as scale is also unknown. Without the alignment you might end up with something like this:



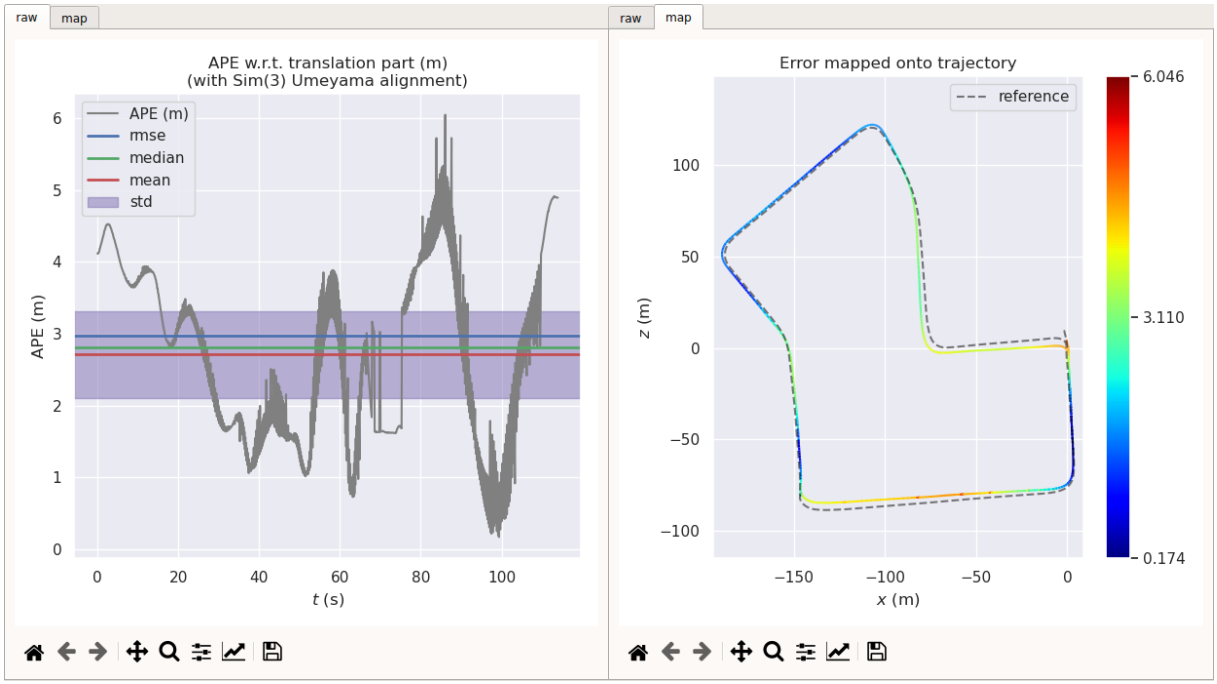
You can do the 7-DoF alignment using EVO by specifying the options “--align --correct_scale” or simply “-as”. The alignment process requires timestamps in the ground-truth pose file. In the KITTI dataset, poses and timestamps are provided in separate files. Thus, our first step is to convert KITTI ground-truth data into a format with timestamps (also known as TUM format). This is the reason that on line123 of mono_kitti.cc we are saving the estimated trajectory in TUM format. We have provided a python script under /Evaluation to do this for you. For example, by running:

python kitti_to_tum.py 07.txt times.txt 07_tum.txt

This will convert the ground-truth poses of KITTI sequence 07 into TUM format and save to 07_tum.txt. Next, you can run the EVO tool to evaluate the estimated trajectory. You can do this by running:

evo_ape tum gt.txt esimate.txt -as --plot --plot_mode xz

The first option “tum” specifies the data format, and the second and third options stand for the ground-truth trajectory file (07_tum.txt) and the estimated trajectory. You can refer to the wiki page for more details. After running the above command, you should get the results in your command line and a window as shown in the figure below. Everything is interactive and you can edit and save the figures by clicking on the buttons below..



To summarize, you will need to follow 2 steps when evaluating **monocular** SLAM on **KITTI** dataset:

1. Convert the KITTI ground-truth file into TUM format.
2. Save estimated trajectory in TUM format and evaluate using EVO.

[1] A Benchmark for the Evaluation of RGB-D SLAM Systems (J. Sturm, N. Engelhard, F. Endres, W. Burgard and D. Cremers), In Proc. of the International Conference on Intelligent Robot Systems (IROS), 2012.
https://vision.in.tum.de/_media/spezial/bib/sturm12iros.pdf

[2] Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite (Andreas Geiger and Philip Lenz and Raquel Urtasun), In Conference on Computer Vision and Pattern Recognition (CVPR), 2012.
http://www.cvlibs.net/publications/Geiger2012CVPR.pdf