# Robot Vision and Navigation

## Coursework 3 - Description

Lourdes Agapito (l.agapito@cs.ucl.ac.uk), J. Wang, Z. Lu

# Overview

**Assignment Submission Date:** 22nd April 2022
**Weighting:** 33% of module total
**Final Submission Format:** For this coursework **you will continue to work in pairs as you did for the previous courseworks.** Each group will submit a report in PDF format (see details below).
**Assessment Format:** the assessment will consist of a written component based on a PDF report.

# Installing ORB-SLAM2 on your machine

For this coursework you will need to download and install ORB-SLAM2 on your machines and you will be running it in **Monocular mode**. We recommend that you use this release of ORB-SLAM2, refactored by Simon and Ziwen, which will make it much easier for you to install the code than using the original repository from the authors.

https://github.com/sjulier/Refactored_ORB_SLAM2

Please go to the Coursework 3 folder on Moodle to find the KITTI and TUM sequences we would like you to work with.

Please tell us which operating system you are currently running by filling in this poll:

https://moodle.ucl.ac.uk/mod/choice/view.php?id=3431419&forceview=1

# Coursework Description: Download, run and evaluate a SLAM system and write a report (100%)

We want you to run the **MONOCULAR** ORB-SLAM system under four different conditions. You must produce the evaluation plots and write your interpretation of the effects caused on the quality of the camera tracking in each case:

1. Run the system with off-the-shelf options, evaluate and obtain the ATE plot for both sequences.

2. Reduce the number of ORB features used by the system and test the impact on the estimation of the camera trajectory. Choose 3 levels of number of features (you will end up with 3 graphs per sequence). Your job is to go into the code and find the place where you can change the parameter/s that govern the number of features that are detected by the system.
3. Turn off the outlier rejection stage and evaluate again. Your job is to go into the code and find the place where you can switch off the function that rejects outliers and run the system without it. The performance of the system should suffer substantially.
4. Turn off the loop closure and evaluate again. Your job is to go into the code and find the place where you can switch off the loop closure and run the system without it. The performance of the system should suffer substantially.

You will submit a PDF report that contains:

a. **A detailed description of the *tracking* thread of MONOCULAR ORB-SLAM [(~2000 words)]**. Only the tracking, not the mapping. You are expected to understand and describe how the camera trajectory (rotation and translation parameters) are estimated including a description of the energy minimization or the parameter estimation. You can refer to the paper associated with the system but do not plagiarise the text. Please always use your own words and always cite or acknowledge any sources that you have used.
b. The **results of the 4 evaluations that you have run**. (listed above as 1, 2, 3 and 4.) You should include a description of each evaluation (see the list above) and the corresponding plots. You should add an explanation for the behaviour you observe.
c. A paragraph at the end of your report should summarise your conclusions.

You will need to run ORB-SLAM2 on two sequences:

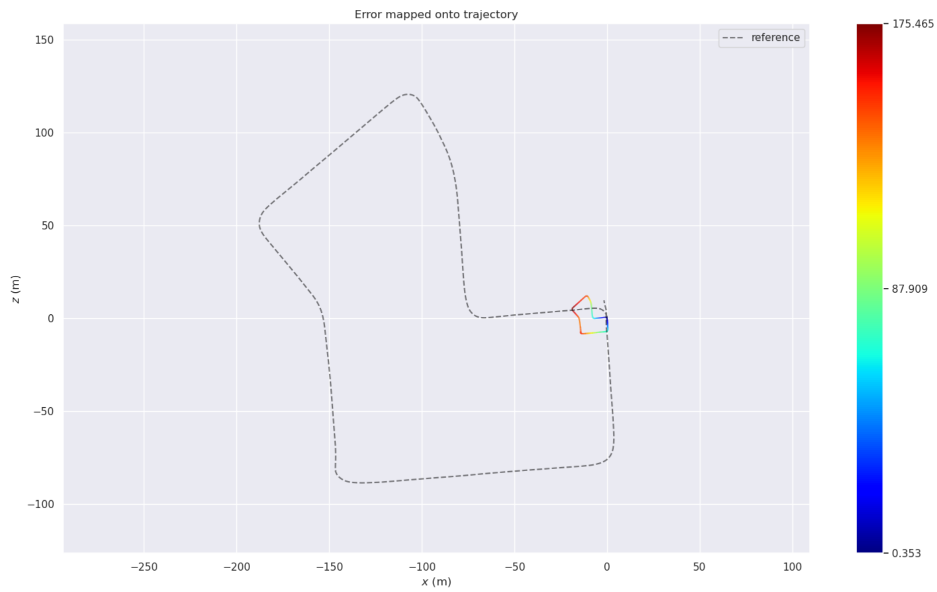
1. KITTI 07: This one is compulsory. We have uploaded this sequence on Moodle. You MUST run Monocular ORB-SLAM2 on this sequence and complete the tasks specified above.
2. For the second sequence, you can choose from the KITTI or TUM datasets. We have provided an extra KITTI sequence and the link to a TUM sequence on Moodle. Please choose from one of these two.

# Run EVO tool on KITTI sequences

In this coursework, you will evaluate the SLAM system using the EVO tool. The installation is very straight forward, 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

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 definition you can refer to the paper of TUM dataset [1]. KITTI dataset also has its own defined metrics [2] but the ideas behind 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:



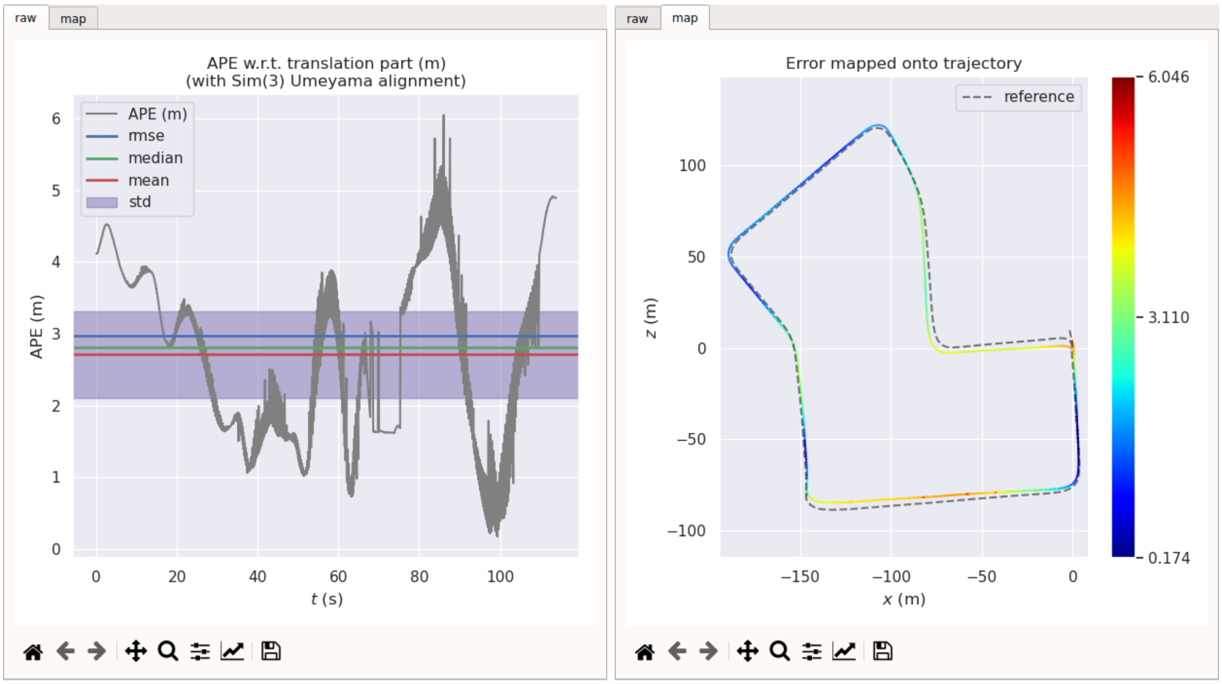

You can do the 7-DoF alignment using EVO by specifying the options “--align --correct_scale” or simply “-as”. The alignment process requires timestamps in the ground-truth pose file. In the KITTI dataset, poses and timestamps are provided in separate files. Thus, our first step is to convert KITTI ground-truth 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