

Final Project CS 498 GC

- ☐ This project consists of a written and coding part. Both must be submitted on canvas.
- ☐ The coding part must be developed in python with ROS2. Upload the python files only.
- ☐ The written part must be submitted in PDF format.

Learning outcomes

- ☐ Understand the main principles of visual SLAM systems.
- ☐ Evaluate the performance of a SLAM system in different environments.
- ☐ Process data collected from a mobile robot using Robot Operating System (ROS)

Description

RTAB-Map (Real-Time Appearance-Based Mapping) is a loop closure detection approach for SLAM. In this exercise, the students must perform RGB-D SLAM using recorded image data. You are provided with two data sequences (one collected on campus, the other collected in a cornfield from the terrasentia robot). The ZED camera was used to collect RGB and depth images. In addition, the robot was equipped with a GPS module, an inertial measurement unit, and wheel encoders. These sensors' measurements were fused by an Extended Kalman Filter (EKF), which computes an approximation of the ground-truth trajectory.

1. Run RTABMap on [rosbag1](#) and [rosbag2](#) (see a tutorial in the appendix section). Add the 3D maps to your report.
2. Implement a ROS node to get the coordinates (x,y,z) of the estimated trajectories by RTAB map (/rtabmap/odom) and the ground truth trajectories (/terrasentia/ekf). Store the data in .txt files. Add the 3D plots of these trajectories to your report.
3. Compute the Root mean square error between the estimated trajectories ($y \in R^3$) and the ground truth trajectories ($\hat{y} \in R^3$).

$$RMSE = \sqrt{\frac{\sum_{i=1}^N \|y(i) - \hat{y}(i)\|^2}{N}},$$

4. Discuss the results.

Appendix A. RTAB-Map installation

```
$ sudo apt install ros-humble-nav2-msgs
$ cd ~/ros2_ws
$ git clone https://github.com/introlab/rtabmap.git src/rtabmap
$ git clone --branch ros2 https://github.com/introlab/rtabmap_ros.git src/rtabmap_ros
$ export MAKEFLAGS="-j6" # Can be ignored if you have a lot of RAM (>16GB)
$ colcon build --symlink-install --cmake-args -DRTABMAP_SYNC_MULTI_RGBD=ON
-DRTABMAP_SYNC_USER_DATA=ON
```

Further info: https://github.com/introlab/rtabmap_ros/tree/ros2#rtabmap_ros

Appendix B. Tutorial to run RTAB-Map

Run on different terminals:

```
>> ros2 run tf2_ros static_transform_publisher 0.0 0.0 0.0 -1.5708 0.0 -1.5708 zed2_imu_link  
zed2_left_camera_optical_frame
```

```
>> ros2 launch rtabmap_ros rtabmap.launch.py tabmap_args:="--delete_db_on_start"  
rgb_topic:=/z/left/color depth_topic:=/terrasentia/zed2/zed_node/depth/depth_registered  
camera_info_topic:=/terrasentia/zed2/zed_node/left/camera_info frame_id:=zed2_imu_link  
approx_sync:=false wait_imu_to_init:=true imu_topic:=/terrasentia/zed2/zed_node/imu/data qos:=1
```

```
>> ros2 bag play my_rosbag
```