

ICP-BASED LOCALIZER SYSTEM IN ROS

Anna Fumagalli

Sapienza University of Rome

1. OVERVIEW

In this project, an ICP-based localization system was developed using the ROS (Robot Operating System) framework. The main objective was to estimate the robot's pose by comparing distance measurements to obstacles and walls obtained from a laser scan with a map of the environment.

The ICP is an iterative algorithm that aligns the scan data with the predefined map by finding the optimal transformation (orientation and translation). With the latter, the ICP prediction of the robot's position in the map is updated.

The developed program consists of several modules that cooperate within ROS: a node that listens to the robot's laser measurements and map of the environment (provided through a map server); the initialization of the robot's pose through a dedicated topic (/initialpose); the management of the map through a KD-Tree structure - which optimizes the search for obstacles and points of interest -; the ICP algorithm that aligns the laser sensor data with the map to calculate the updated robot pose. The project was structured using a Catkin-type ROS workspace, with the partitioning into folders separating source code, configuration files, and test data. Localization is simulated in a virtual environment using the Stage ROS simulator.

2. HOW TO COMPILE

For project development, it is required to have installed the map server, stage-ros and teleop - twist - keyboard packages. I have worked with *ROS Noetic*.

Create the folder `catkin_loc_ws/src` and then initialize the workspace:

```
cd ~
mkdir -p catkin_loc_ws/src
cd catkin_loc_ws/src
catkin_init_workspace
```

Clone the repository linked here and build the project in `catkin_loc_ws`:

```
git clone https://github.com/
AnnaFumagalli98/ICP-based-Localization
cd catkin_loc_ws
```

```
catkin build
```

3. HOW TO RUN AND TEST

Open a new terminal and once in `catkin_loc_ws` source it and run the `roscore`:

```
cd catkin_loc_ws
source devel/setup.bash
roscore
```

Once `roscore` is executed, enter the `catkin_loc_ws` folder, source it, then enter the `catkin_loc_ws/src/ICP-based-Localization/02_icp_localization` and finally run the `map_server` node:

```
cd catkin_loc_ws
source devel/setup.bash
```

```
cd catkin_loc_ws /src/ICP-based-Localization
/02_icp_localization /
```

```
roslaunch map_server map_server test_data /
cappero.yaml
```

Open a terminal and once in `catkin_loc_ws`, sourced it and entered in `catkin_loc_ws/src/ICP-based-Localization/02_icp_localization`, launch the `rviz` configuration:

```
rviz -d test_data/rviz.rviz
```

Open a terminal to launch the simulator, once entered in `catkin_loc_ws`, sourced it, entered in `catkin_loc_ws/src/ICP-based-Localization/02_icp_localization`:

```
roslaunch stage_ros stageros
test_data/cappero.world
```

Open the last terminal, to run the `localizer` node after being in `catkin_loc_ws` and having sourced it:

```
roslaunch icp_localization localizer_node
```

The initial position has to be set using RViz tool *2D Pose Estimate* which is then passed to the ICP as a message through a callback function. In the simulator you can move the robot with the mouse.