Lab2 - Line features with Split & Merge

1   Introduction

This lab is designed to learn how to extract line features from point data. The data is extracted from the **sensor_msgs/LaserScan** messages saved in a bagfile dataset and are converted to (x, y) points in a 2D plane. Your goal is to create an algorithm that creates lines from the given points. The Split & Merge algorithm explained in the attached scientific paper will be the base of its implementation.

The resulting algorithm will be used on the following labs as a base for feature extraction in Particle Filter, Extended Kalman Filter and Simultaneous Localization and Mapping.

2   Pre-lab

Read and understand the Split & Merge algorithm explained in the attached paper. Instead of fitting a line to all points, we are going to implement the *Iterative-End-Point-Fit* for its simplicity.

Lines will be represented in the form (1) because simplifies operations.

$$ax + by + c = 0 \tag{1}$$

The questions for the prelab (answer in UdG moodle) are:

- Given two points (x_1, y_1) and (x_2, y_2), compute the line parameters (a, b, c).

- Given a line (a, b, c) and a point (x, y) compute the distance from the point to the line.

Answer with a single formula containing only the given data.

3   Lab work

The ROS package **lab2_splitandmerge** (from the metapackage **probabilistic_labs** that you cloned during the lab0) contains all necessary code for running this lab (it is recommendable to pull the last version from the repository before starting the lab). Test the code using the following command:

```
roslaunch lab2_splitandmerge lab2_splitandmerge.launch
```

This will playback the data recorded into the bagfile **probabilistic_basics/bags/dataset3.bag** and two nodes. One will handle the odometry of the robot (**lab2_splitandmerge/src/odometrynode.py**) and the other one handling the laser scan data (**lab2_splitandmerge/src/splitandmergenode.py**).

As you can see the dummy Split & Merge algorithm implemented, only provides lines from the first to the last point of each scan without any other computation. Also you can observe how the odometry is not perfect and the accumulation of the scans does not represent properly the map (Fig. 1).
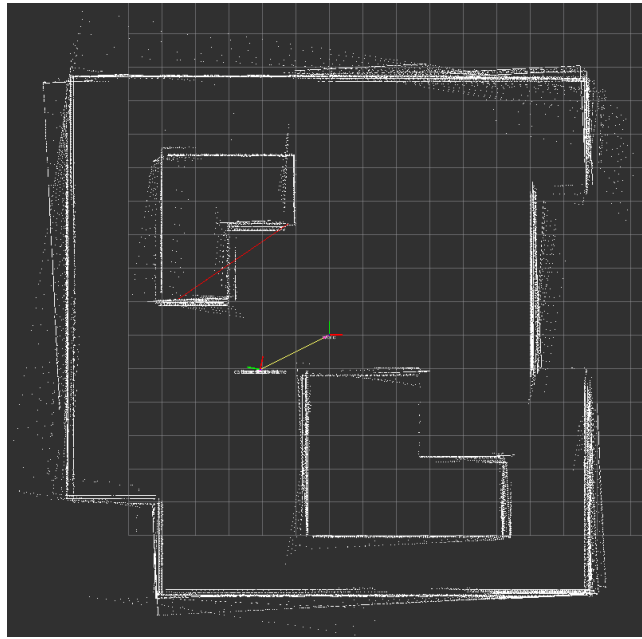
Figure 1: All points plotted with respect to odometry (white) and the line from the Split & Merge.

For completing the lab you have to modify the file `lab2_splitandmerge/src/splitandmerge.py`. The functions that have to be modified are highlighted. You can tune the parameters of the Split & Merge algorithm by changing the values assigned by default when the `splitandmerge` function is called.

For viewing properly each scan in Rviz, decrease the `LaserScan` decay time to 0, so it will only show the last published scan. Also change the fixed frame to `camera_depth_frame` to see the position of the robot fixed.

4   Optional

- Test the algorithm with both datasets `probabilistic_basics/bags/dataset3.bag` (the one by default) and `probabilistic_basics/bags/dataset2.bag`.

  (You will need to modify the file: `launch/lab2_splitandmerge.launch`).

- Represent all the lines that have been obtained by the Split & Merge algorithm in a single image, showing the discrepancies caused by the odometry. (You will probably need to modify the file: `splitandmergenode.py`).

5   Lab report

Write a brief report (to be submitted in a pdf of maximum 3 pages) explaining your solution and problems faced. Include only the final `splitandmerge.py` file for us to test. If you have tested it with both datasets and different parameters are needed leave the `splitandmerge.py` with the parameters for `dataset3.bag` and report the ones used for `dataset2.bag` in the submitted pdf. The deadline for the submission of this lab is March 14th at 23:55.