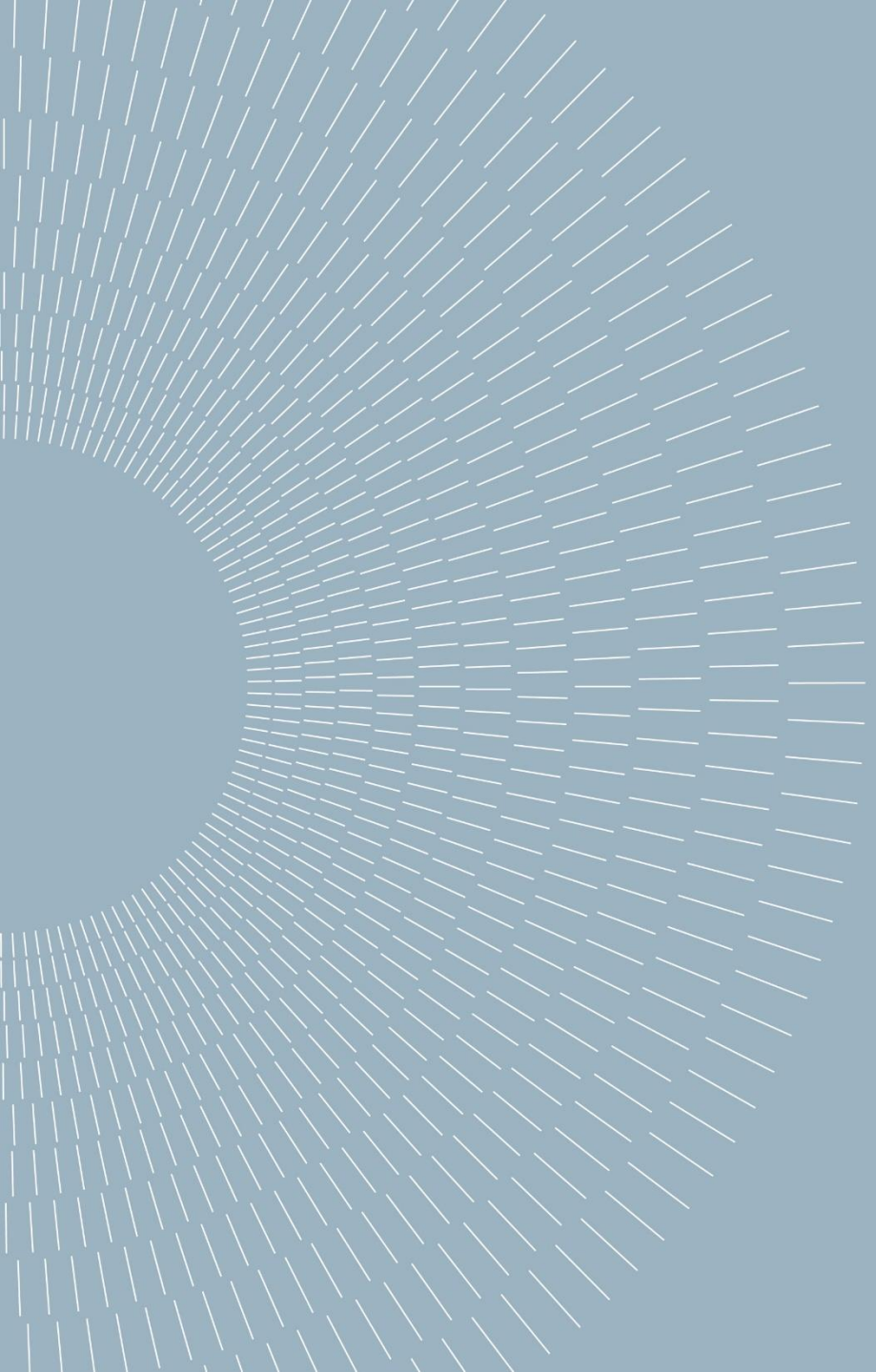
POLITECNICO DI MILANO

AUTONOMOUS VEHICLES

S. Arrigoni



Localization in ROS

Introduction to SLAM

POLITECNICO MILANO 1863

Introduction

What is Simultaneous Localization and Mapping (SLAM)?

In order to plan trajectories for a vehicle we need:

- To known our position (global? relative in a given map?)
- To know the surrounding environment (Map?)

Introduction

What is Simultaneous Localization and Mapping (SLAM)?

Position:

- Global position not always available (underwater, GNSS in a urban area or tunnel,…)

Known Map:

- Not always available;
- Can vary over time

Introduction

What is Simultaneous Localization and Mapping (SLAM)?

Localization: Robot needs to estimate its location with respects to objects in its environment (Map provided).

Mapping: Robot need to map the positions of objects that it encounters in its environment (Robot position known)

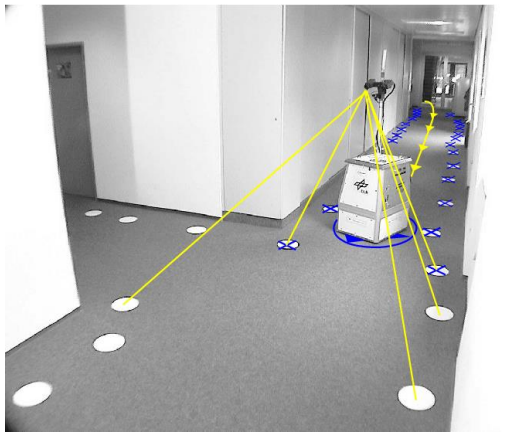
SLAM: Robot simultaneously maps objects that it encounters and determines its position (as well as the position of the objects) using noisy sensors

Introduction

MAP: different representations available in literature

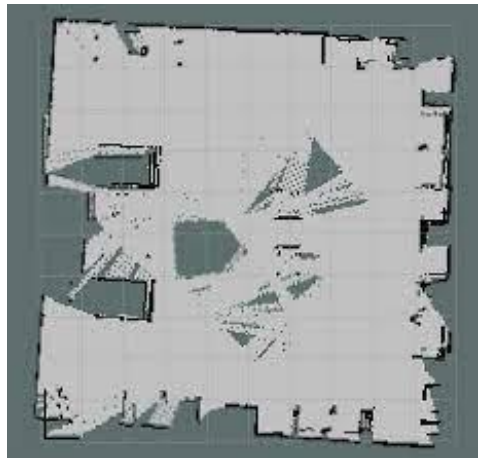
Landmark based: set of fixed point of interest
[Leonard et al., 98; Castelanos et al., 99: Dissanayake et al., 2001; Montemerlo et al., 2002]

- ***sparse***
- ***usually for localization***



Grid Map / Scan: environment discretized into cells (occupied or empy)
[Lu & Milios, 97; Gutmann, 98: Thrun 98; Burgard, 99; Konolige & Gutmann, 00; Thrun, 00; Arras, 99; Haehnel, 01; Grisetti et al., 05; …]

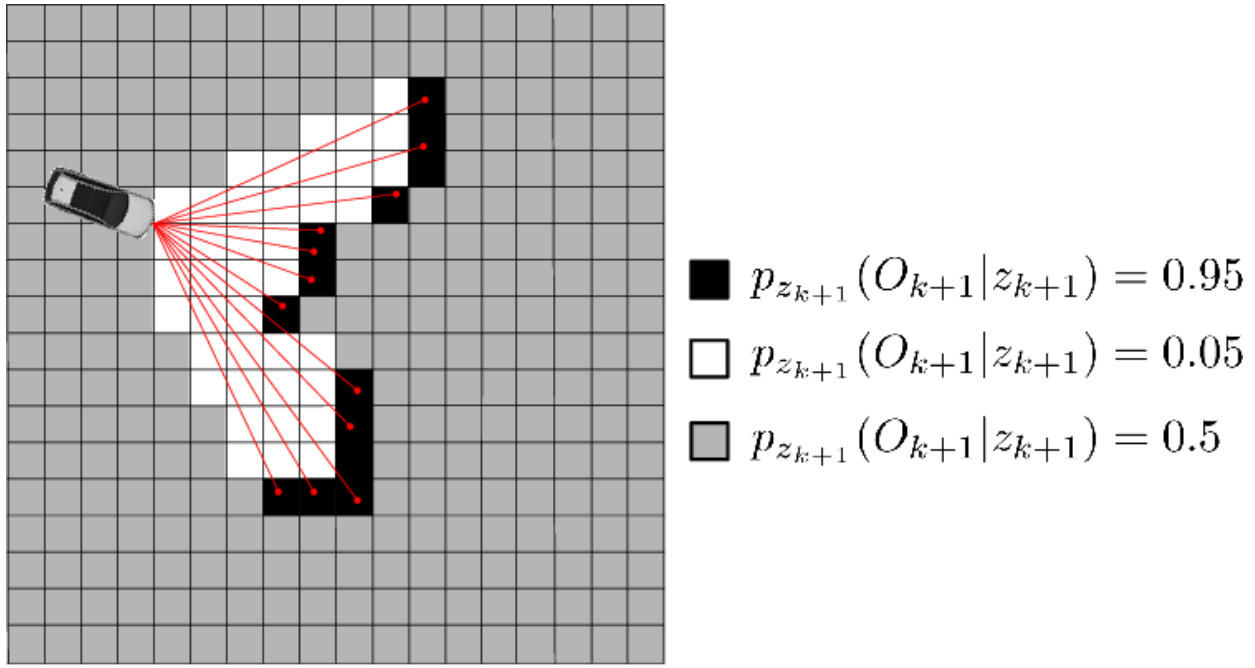
- ***Dense (info of all points)***
- ***usually for planning***



Introduction

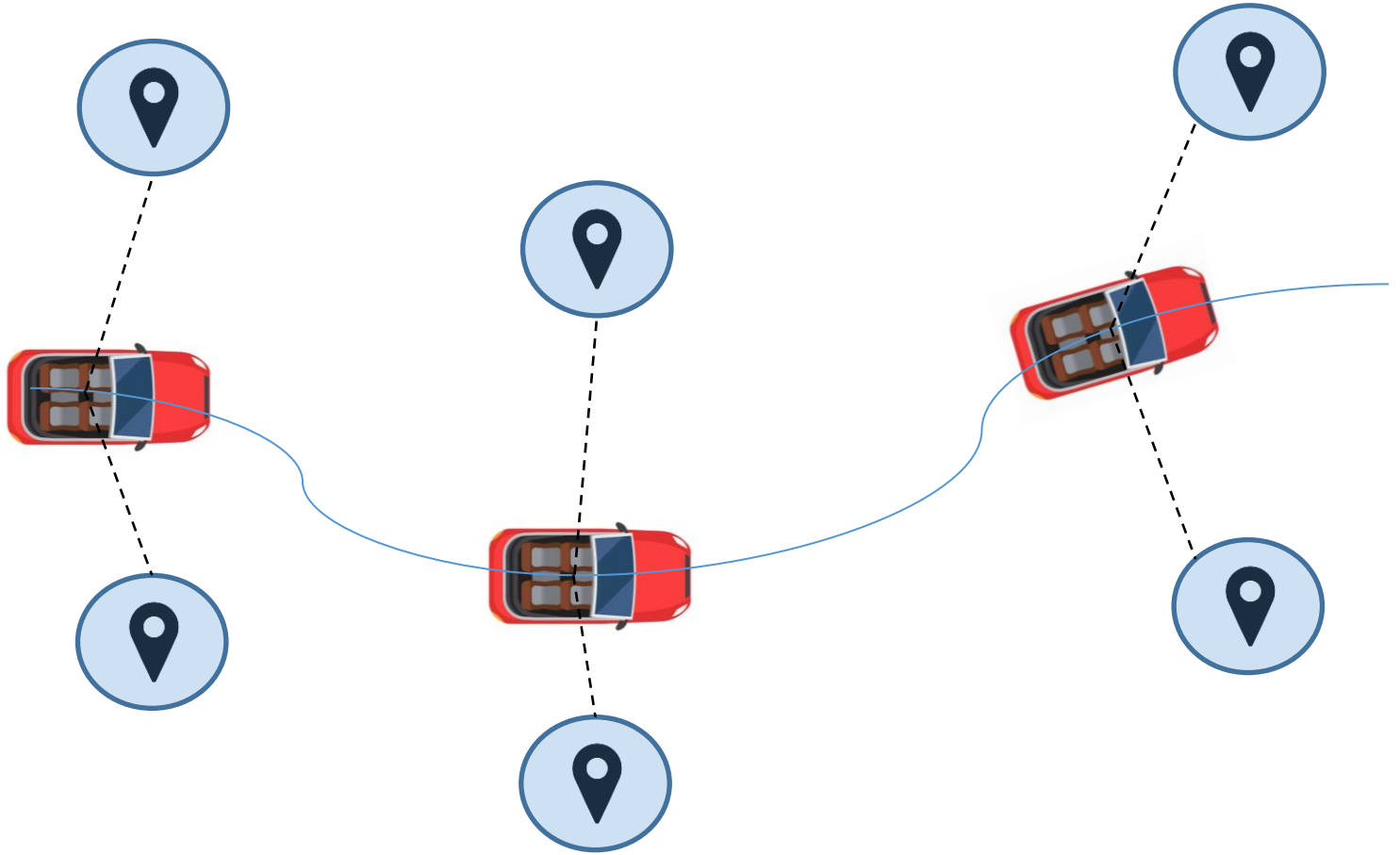
Occupancy Grid: 2D representation for maps where:

each cell is considered independent and its value reflects the probability of being occupied



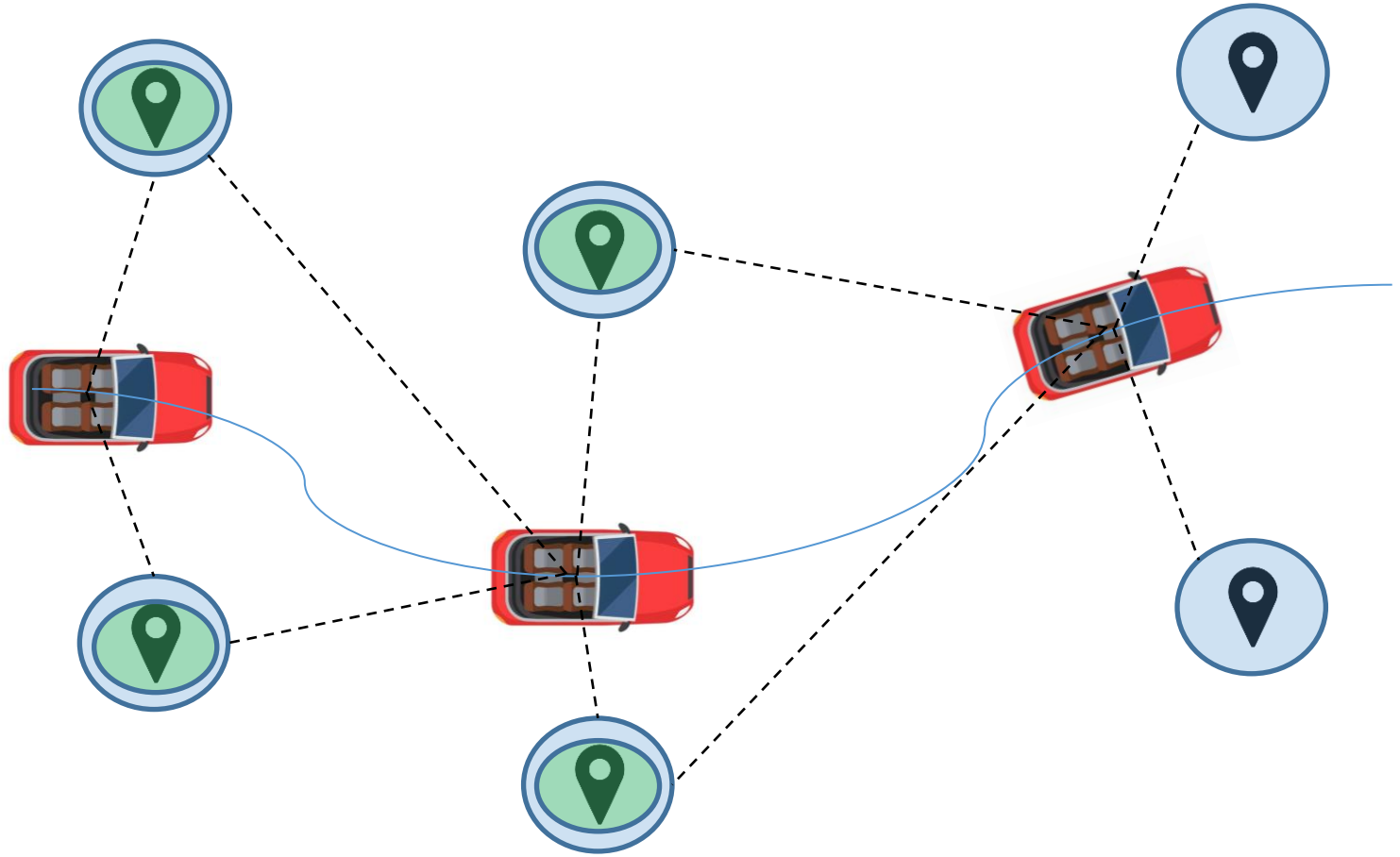
Introduction

MAPPING from ideal localization



Introduction

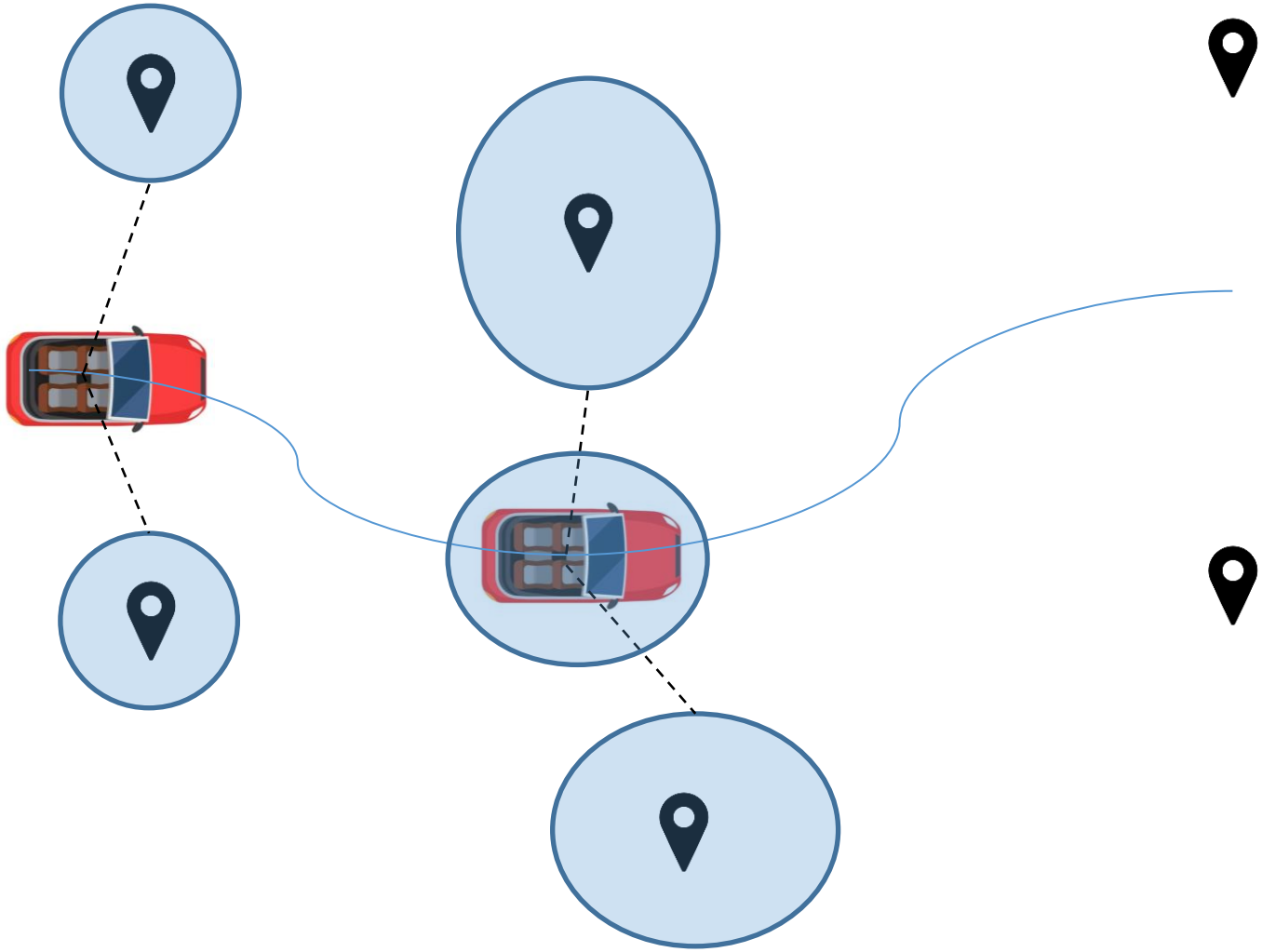
MAPPING from ideal localization



PROBLEM: odometry is usually noisy!

SLAM

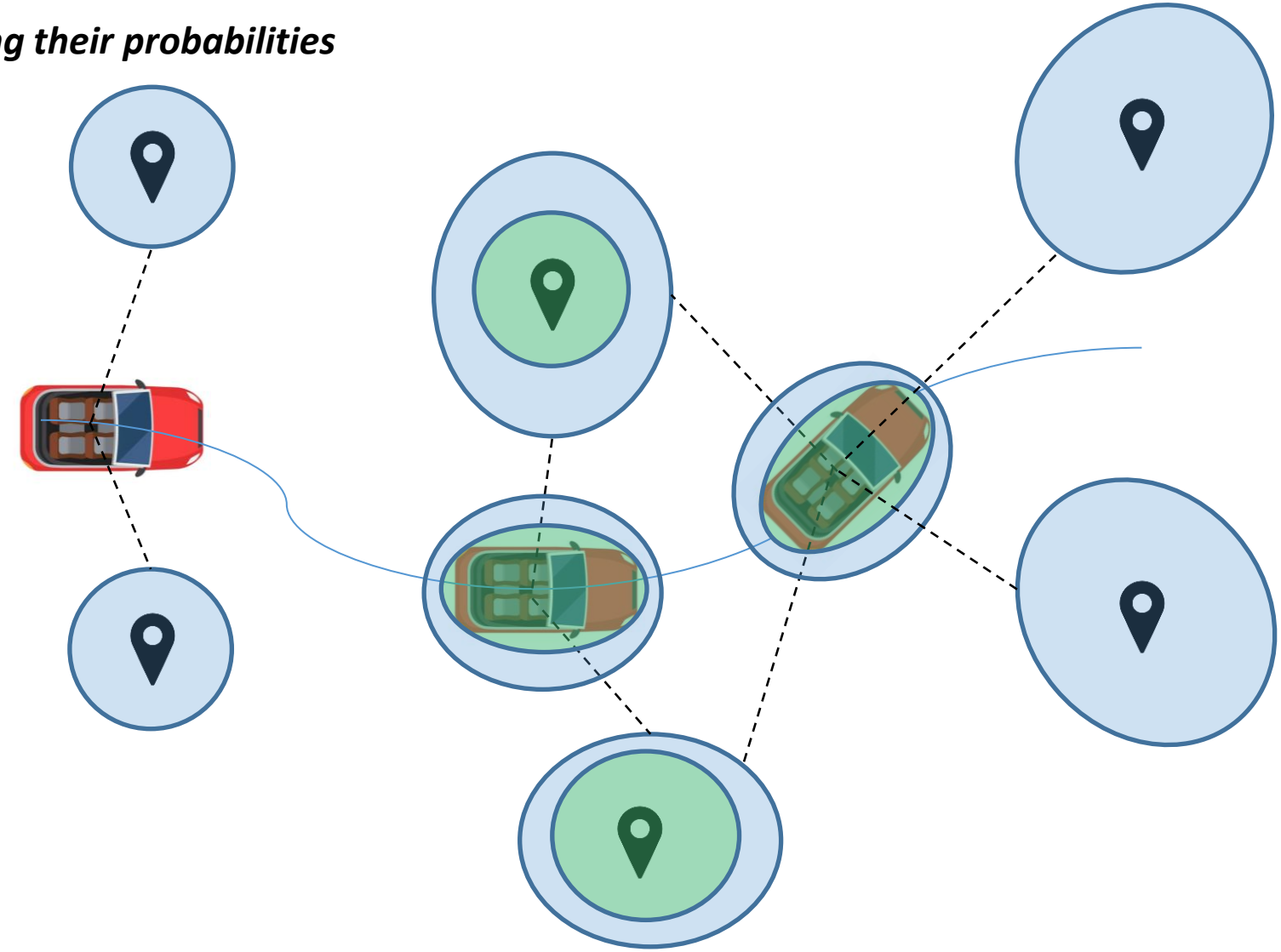
Start to localize and map…



- *localization is affected by noise*
- *next landmarks are affected by loc. noise + meas. noise*

SLAM

update odometry and map considering their probabilities

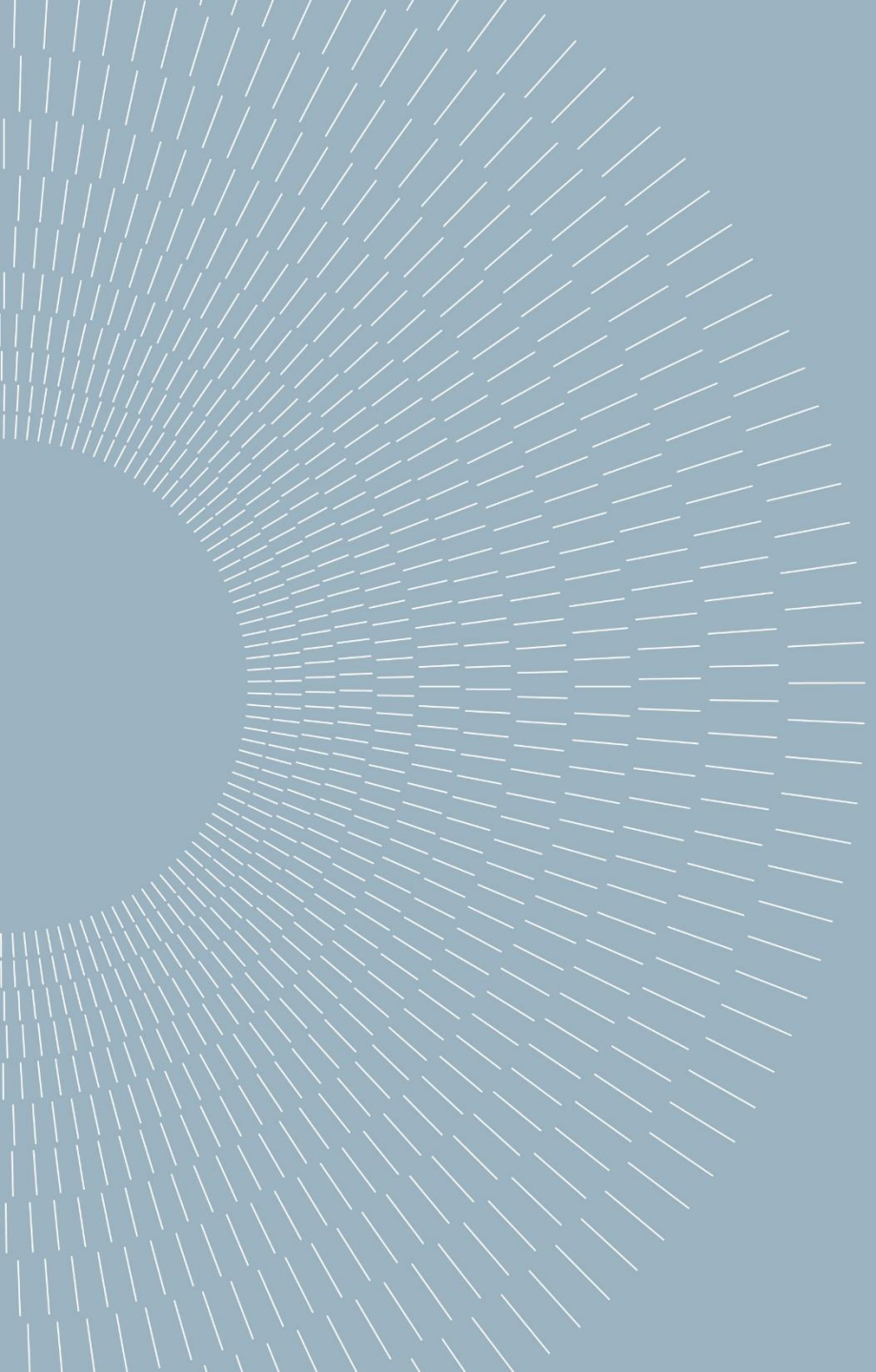


- *same landmarks are measure again: this reduces meas. Noise*
- *localization noise is so also reduced*

SLAM

If you want to go deeper…

- *http://dspace.mit.edu/bitstream/handle/1721.1/36832/16-412JSpring2004/NR/rdonlyres/Aeronautics-and-Astronautics/16-412JSpring2004/A3C5517F-C092-4554-AA43-232DC74609B3/0/1Aslam_blas_report.pdf*
- *http://ais.informatik.uni-freiburg.de/teaching/ss12/robotics/slides/12-slam.pdf*
- *or other online-courses…*



ROS

Navigation stack

POLITECNICO MILANO 1863

Introduction

ROS Navigation stack

The Navigation Stack takes in information from odometry and sensor streams and outputs velocity commands to send to a mobile base.

Adapt it for an arbitrary robot requires some tuning. It is mandatory that:

- the robot runs ROS;
- To have a tf transform tree in place;
- To publish sensor data using the correct ROS Message types.

Introduction

ROS Navigation stack

inputs
- Odometry;
- Sensor data;
- Goal pose.

outputs
- (safe) velocity commands.
  From A to B in a safe way

limitations
- Just for 2D navigation
- Just for differential drives/ holonomic robots;
- Requires a planar laser;
- Better with squared (circular) robots

Introduction

ROS Navigation stack

- move_base
- nav_core

Main node & class definition for custom code

- amcl
- robot_pose_ekf

Localization methods

- base_local_planner
- carrot_planner
- dwa_local_planner

Local planning algorithms

- navfn
- global_planner

Global planning algorithms

- move_slow_and_clear
- rotate_recovery
- clear_costmap_recovery

Recovery algorithms

- costmap_2d
- map_server
- voxel_grid

Tools for mapping

- fake_localization
- move_base_msgs

Debug Tools

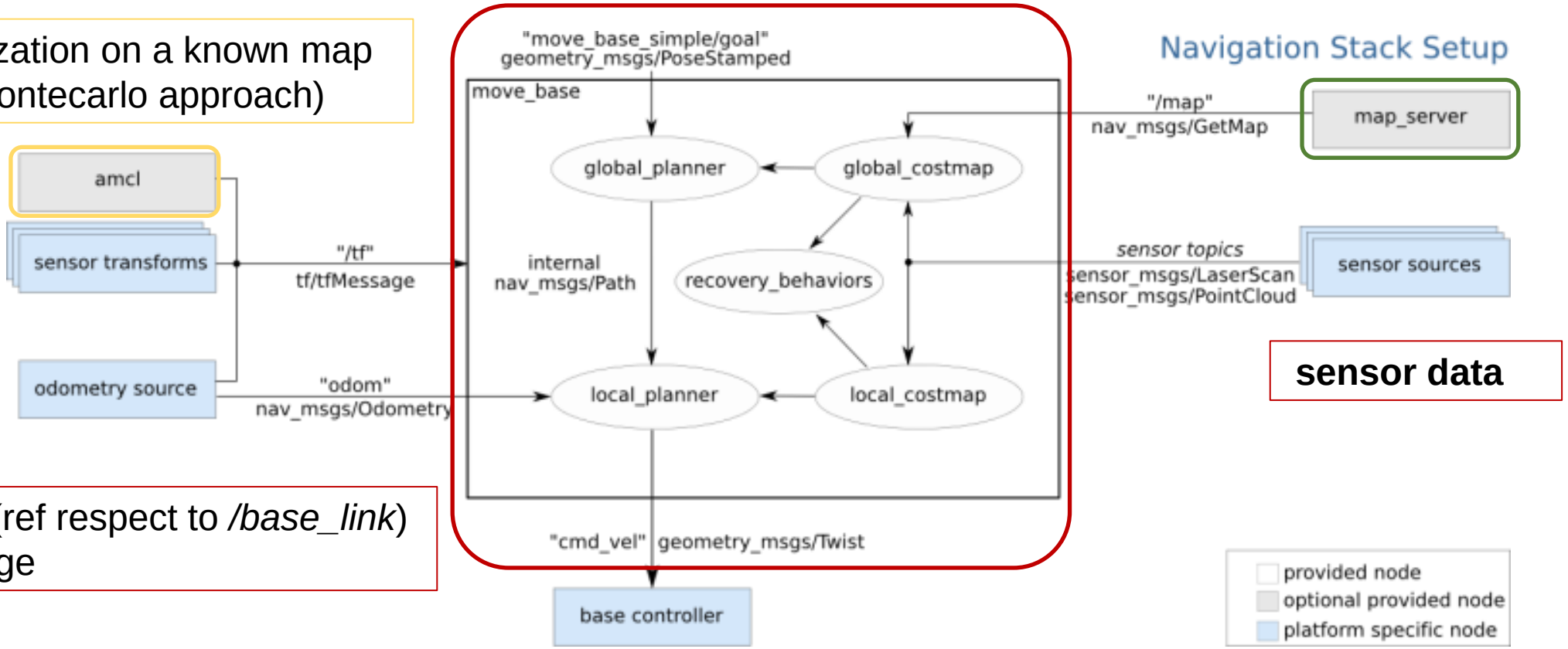
ref. *"**Robotics**" course by M.Matteucci, S.Mentasti*

Introduction

ROS Navigation stack

Ufficial diagram

amcl localization on a known map (adaptive montecarlo approach)



sensor data

/tf of sensors (ref respect to */base_link*)
/odom message

Introduction

MAP Server

- Tool of ROS navigation stack for publishing and saving maps.
- Generates maps both via topic and via service.
- Can be used to save dynamically generated maps.

- ***We'll use it to save the MAP***

```
rosrun map_server map_saver -f ~/<choose a directory>/map_name
```

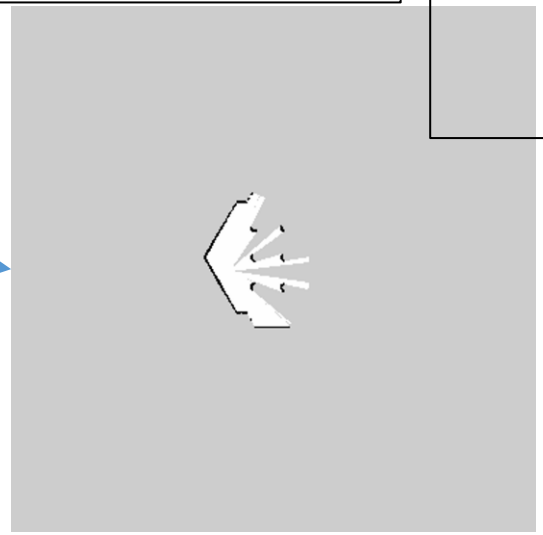
Introduction

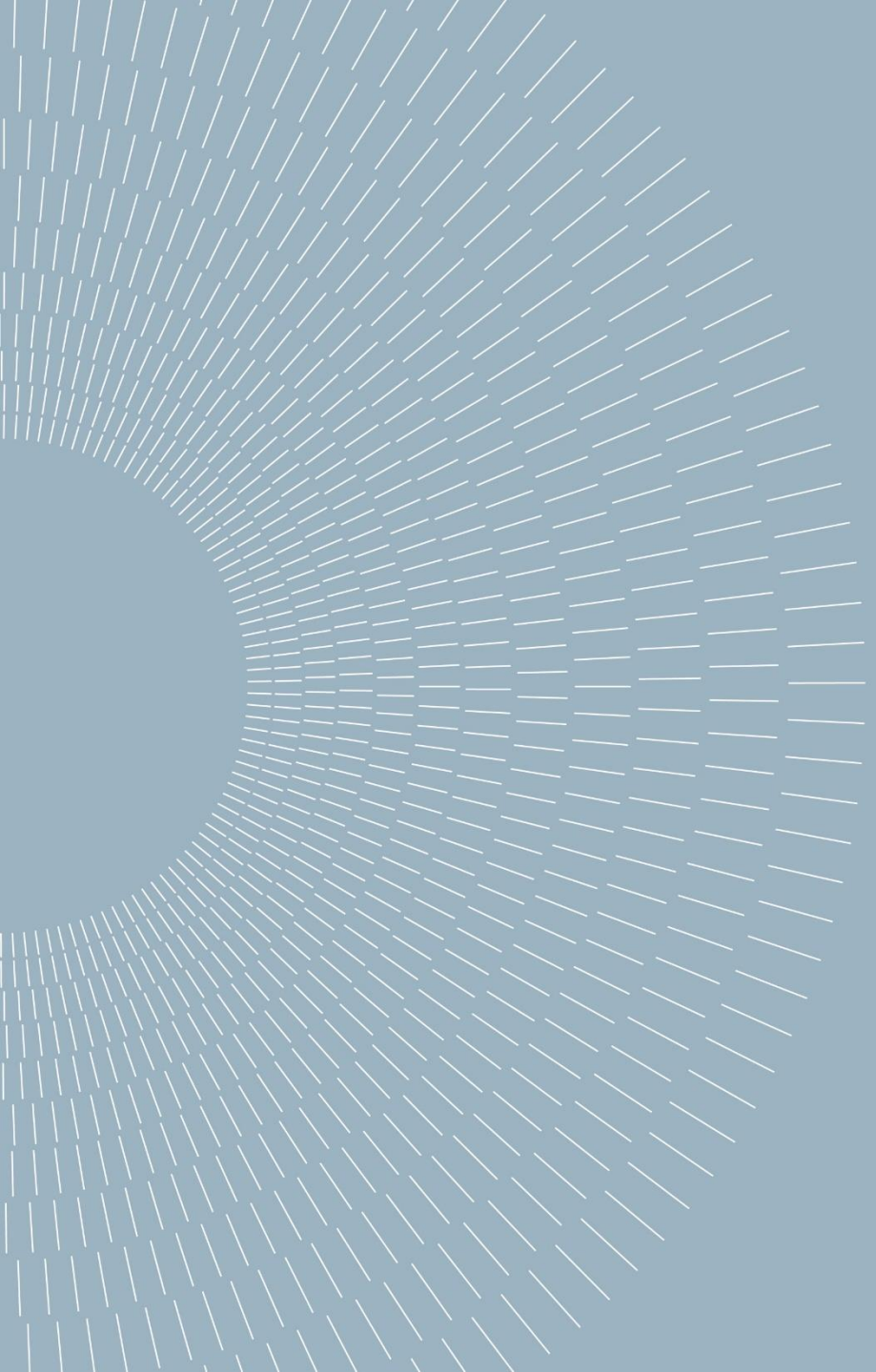
MAP Server

- Map is composed by:
  - YAML file = map meta-data
  - PGM file = encoded data of occupancy grid

```yaml
image: test_mappa.pgm
resolution: 0.050000
origin: [-10.000000, -10.000000, 0.000000]
negate: 0
occupied_thresh: 0.65
free_thresh: 0.196
```

Map name and directory
Map resolution m/px
Lower-left pixel as (x,y,angle)
White free, black occupied
Th limit to occupied (>)
Th limt to free (<)





ROS

GMAPPING

POLITECNICO MILANO 1863

How to map a new environment

Gmapping

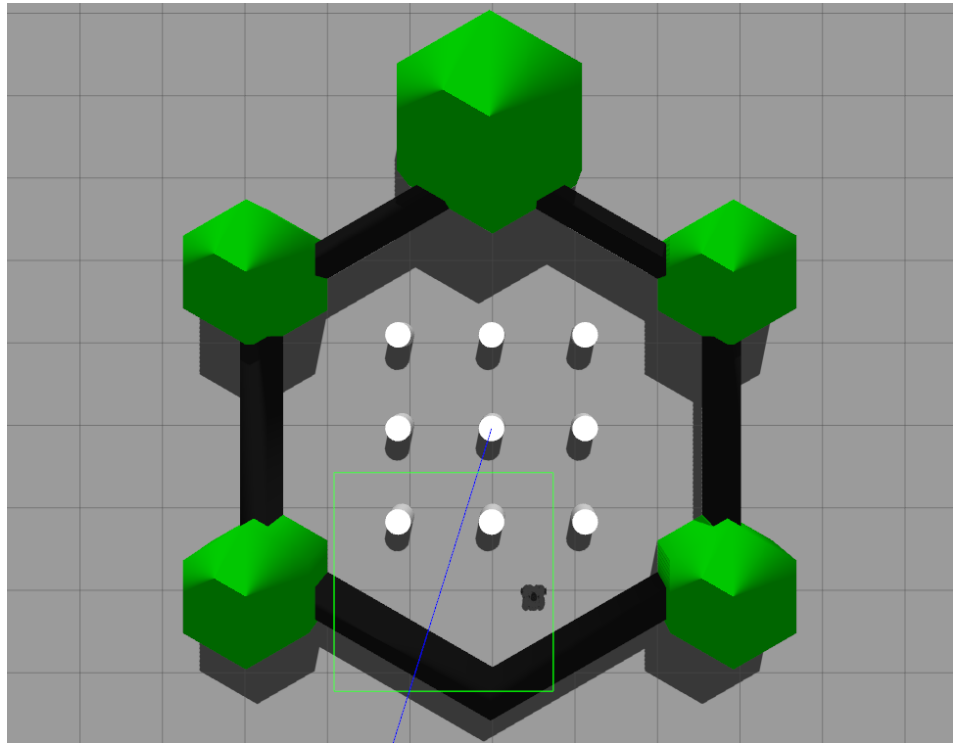
Mapping is usually done by means of SLAM:
we'll specifically use «Gmapping» package available

Using ***laser scan*** messages, ***slam_gmapping node***
generates a ***map*** as topic

How to map a new environment

example

Let’s initialize Gazebo:



```
roslaunch turtlebot_gazebo turtlebot_world.launch
```

How to map a new environment

example

Let’s initialize Gmapping node:

```
roslaunch turtlebot3_slam turtlebot3_gmapping.launch
```

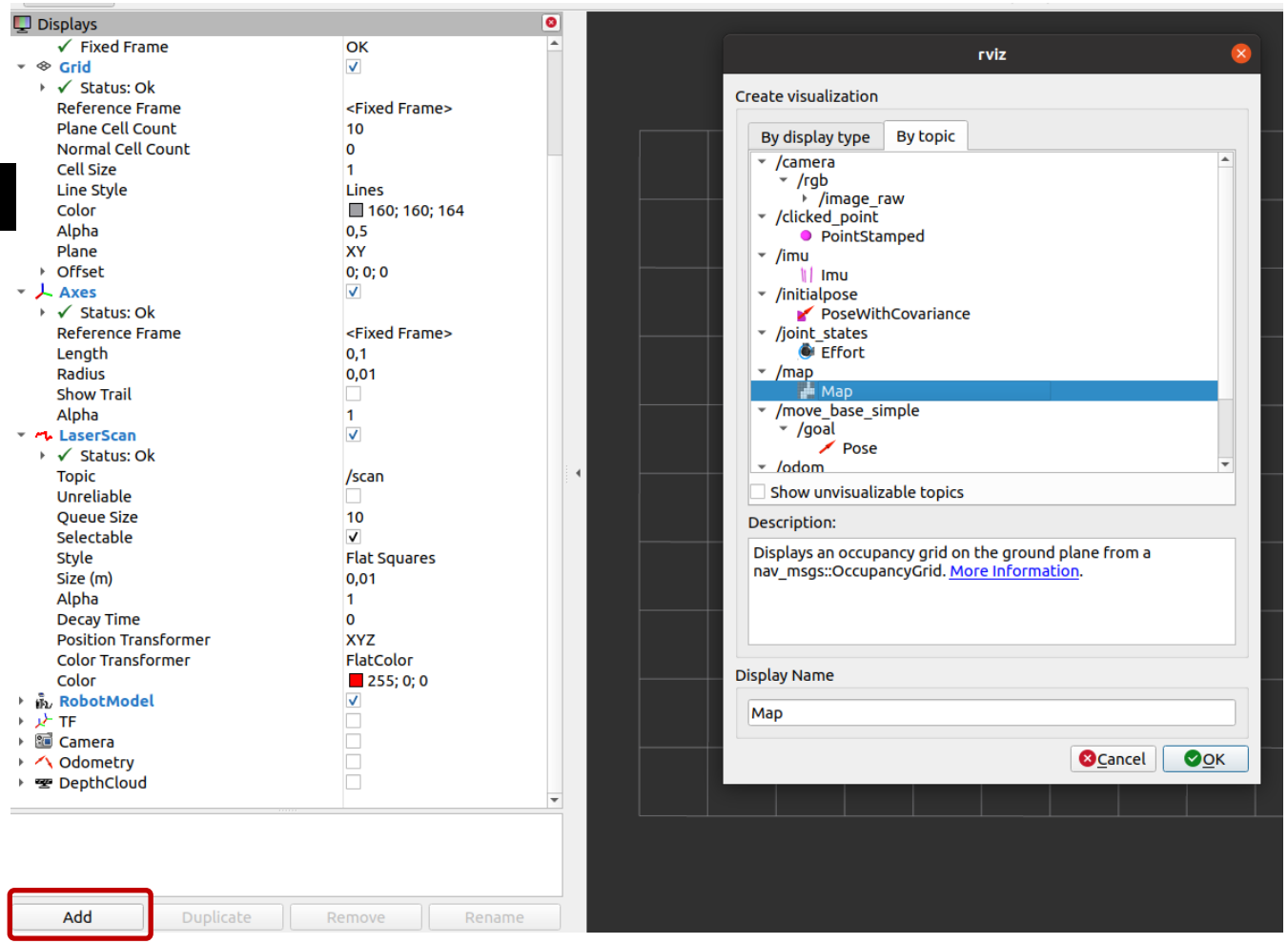
How to map a new environment

Let’s open visualizer (RVIZ):

```
roslaunch turtlebot3_gazebo turtlebot3_gazebo_rviz.launch
```

Let’s add map message:

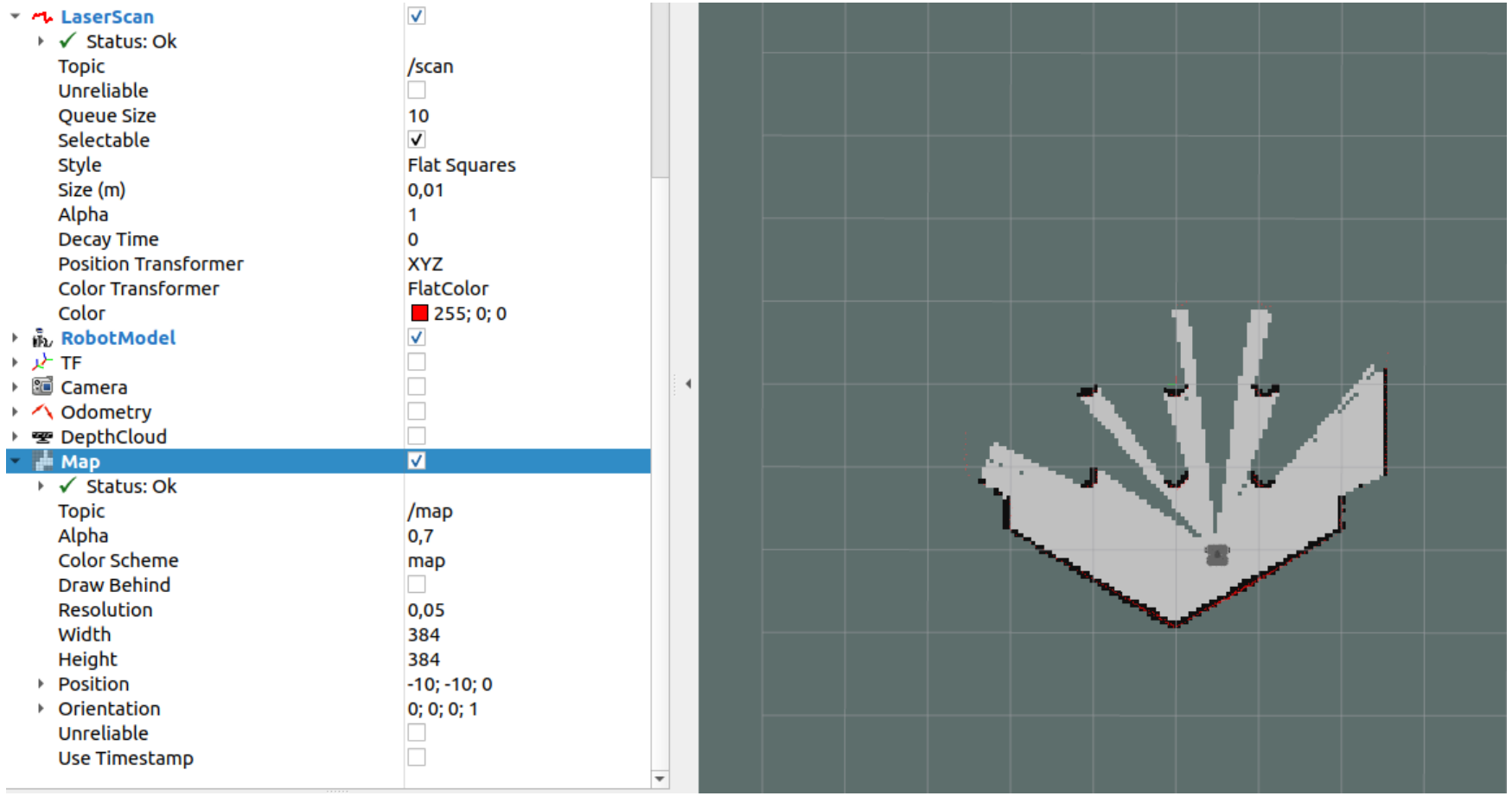
example

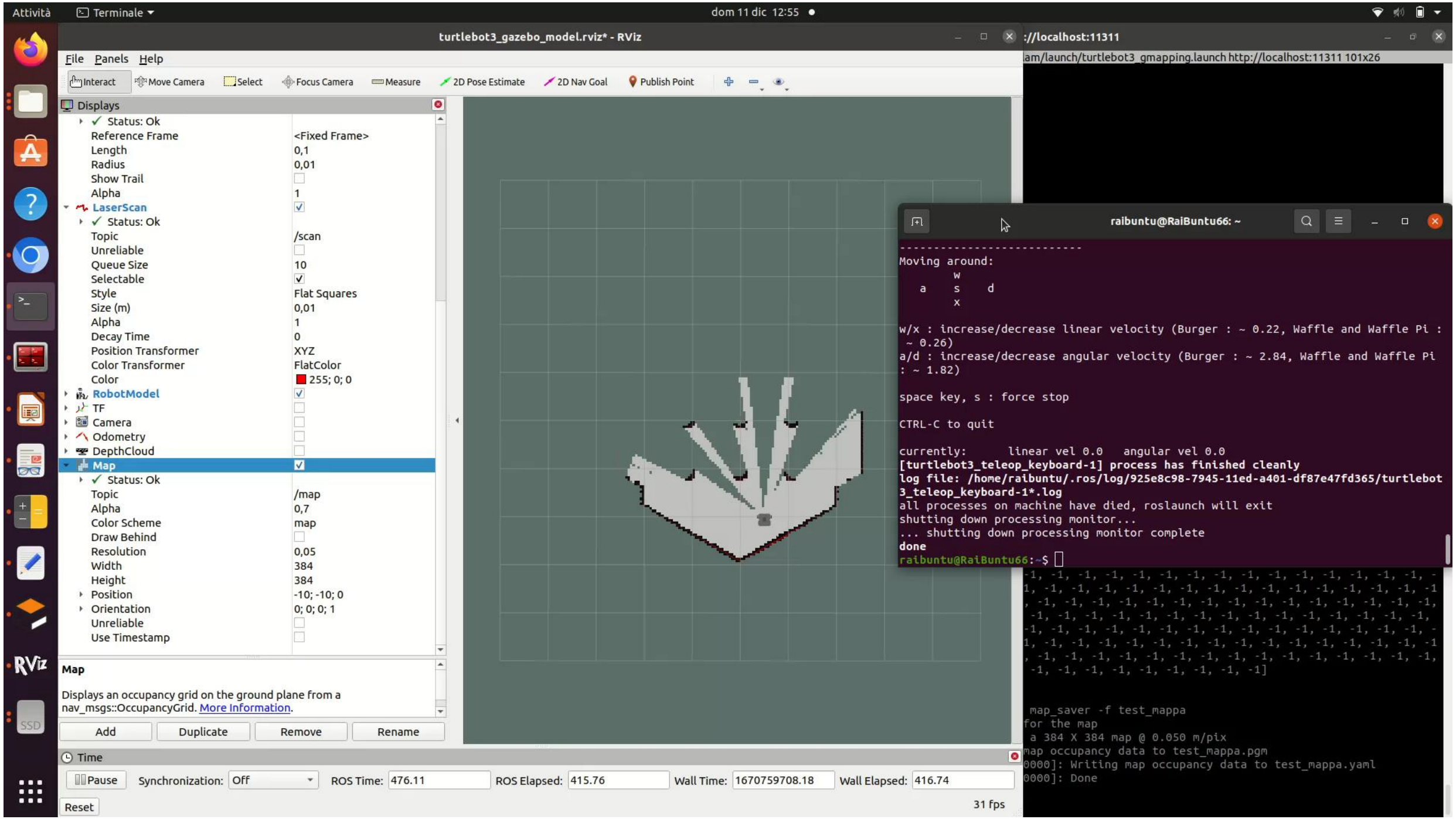


How to map a new environment

example

Result




Attività
Terminale
dom 11 dic 12:55
turtlebot3_gazebo_model.rviz* - RViz
File Panels Help
Interact Move Camera Select Focus Camera Measure 2D Pose Estimate 2D Nav Goal Publish Point
Displays
Status: Ok
Reference Frame <Fixed Frame>
Length 0,1
Radius 0,01
Show Trail
Alpha 1
LaserScan
Topic /scan
Unreliable
Queue Size 10
Selectable
Style Flat Squares
Size (m) 0,01
Alpha 1
Decay Time 0
Position Transformer XYZ
Color Transformer FlatColor
Color 255; 0; 0
RobotModel
TF
Camera
Odometry
DepthCloud
Map
Topic /map
Alpha 0,7
Color Scheme map
Draw Behind
Resolution 0,05
Width 384
Height 384
Position -10; -10; 0
Orientation 0; 0; 0; 1
Use Timestamp
Map
Displays an occupancy grid on the ground plane from a nav_msgs::OccupancyGrid. More Information.
Add Duplicate Remove Rename
Time
Pause Synchronization: Off ROS Time: 476.11 ROS Elapsed: 415.76 Wall Time: 1670759708.18 Wall Elapsed: 416.74
Reset
31 fps
://localhost:11311
am/launch/turtlebot3_gmapping.launch http://localhost:11311 101x26
raibuntu@RaiBuntu66: ~
Moving around:
w
a s d
x
w/x : increase/decrease linear velocity (Burger : ~ 0.22, Waffle and Waffle Pi : ~ 0.26)
a/d : increase/decrease angular velocity (Burger : ~ 2.84, Waffle and Waffle Pi : ~ 1.82)
space key, s : force stop
CTRL-C to quit
currently: linear vel 0.0 angular vel 0.0
[turtlebot3_teleop_keyboard-1] process has finished cleanly
log file: /home/raibuntu/.ros/log/925e8c98-7945-11ed-a401-df87e47fd365/turtlebot3_teleop_keyboard-1*.log
all processes on machine have died, roslaunch will exit
shutting down processing monitor...
... shutting down processing monitor complete
done
raibuntu@RaiBuntu66:~$
map_saver -f test_mappa
for the map
a 384 X 384 map @ 0.050 m/pix
map occupancy data to test_mappa.pgm
0000]: Writing map occupancy data to test_mappa.yaml
0000]: Done

How to map a new environment

example

Let’s finally save the map

```
rosrun map_server map_saver -f ~/<choose a directory>/test_map
```

```
raibuntu@RaiBuntu66:~$ rosrun map_server map_saver -f test_mappa
[ INFO] [1670759456.890322363]: Waiting for the map
[ INFO] [1670759457.158102077]: Received a 384 X 384 map @ 0.050 m/pix
[ INFO] [1670759457.158121262]: Writing map occupancy data to test_mappa.pgm
[ INFO] [1670759457.160242050, 225.752000000]: Writing map occupancy data to test_mappa.yaml
[ INFO] [1670759457.160289518, 225.752000000]: Done
```

This creates 2 files:

- test_map.pgm
- test_map.yaml

That’s it for today…

See you next time!

S. Arrigoni