

Turtlebot3 SLAM与导航虚拟仿真实验-基本实验

- 本次实验课适用ROS系统版本为 `melodic` 和 `kinetic`。如果有同学使用其他版本ROS系统而在实验过程中出现问题，恐怕无法给予帮助。
- 仿真实验和实际环境实验有较大不同，本实验仿真运行效果也许不错，但实际环境中要差很多。
- 从PDF文件复制终端命令的时候会行末的回车换行符一并复制，可先将复制内容粘贴到一文本文件中，删除行尾换行符后再复制、粘贴到终端中执行。

目录

Turtlebot3 SLAM与导航虚拟仿真实验-基本实验

一、基本实验-激光SLAM建图与导航

1.1 激光SLAM建图

1.2 自主导航

一、基本实验-激光SLAM建图与导航

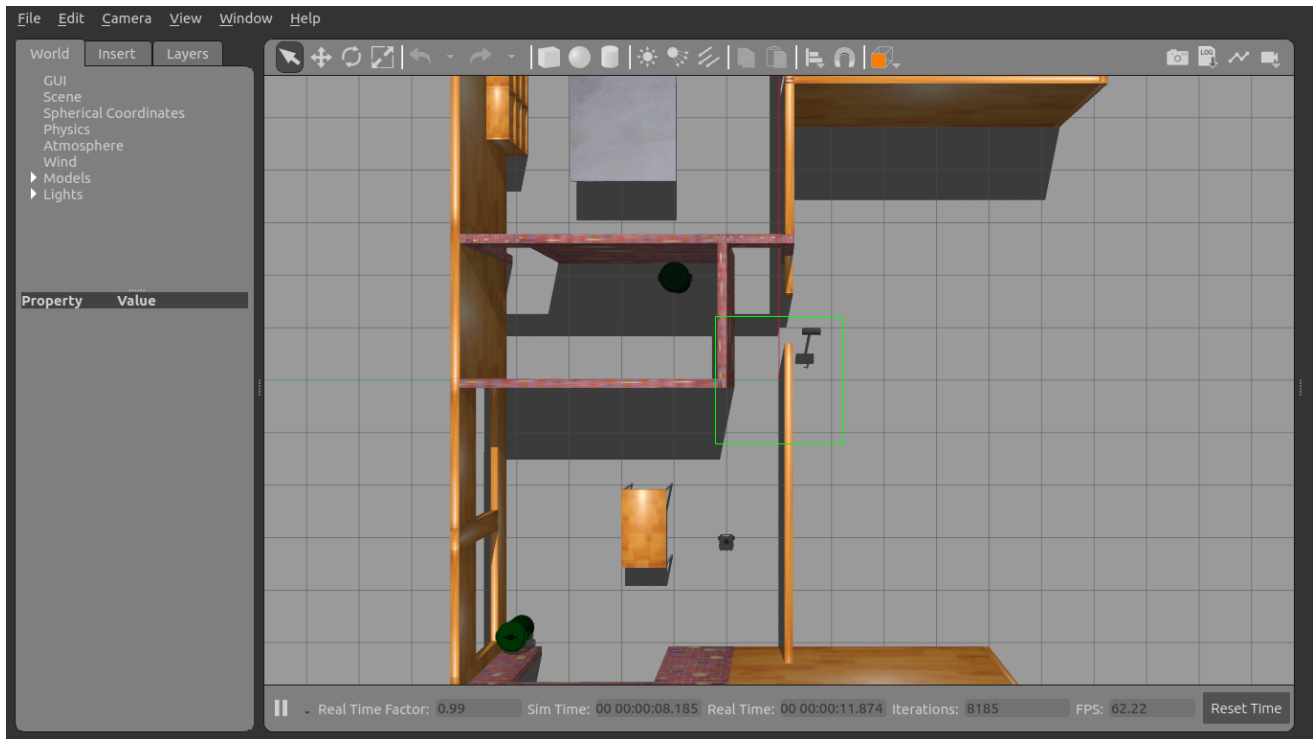
基本实验的目的是使用现有的功能包，利用激光雷达数据进行SLAM（同时定位与建图）得到环境地图，并在该地图的基础上进行Turtlebot3机器人的自主导航。

1.1 激光SLAM建图

首先运行虚拟仿真环境：

```
$ roslaunch turtlebot3_gazebo turtlebot3_house.launch
```

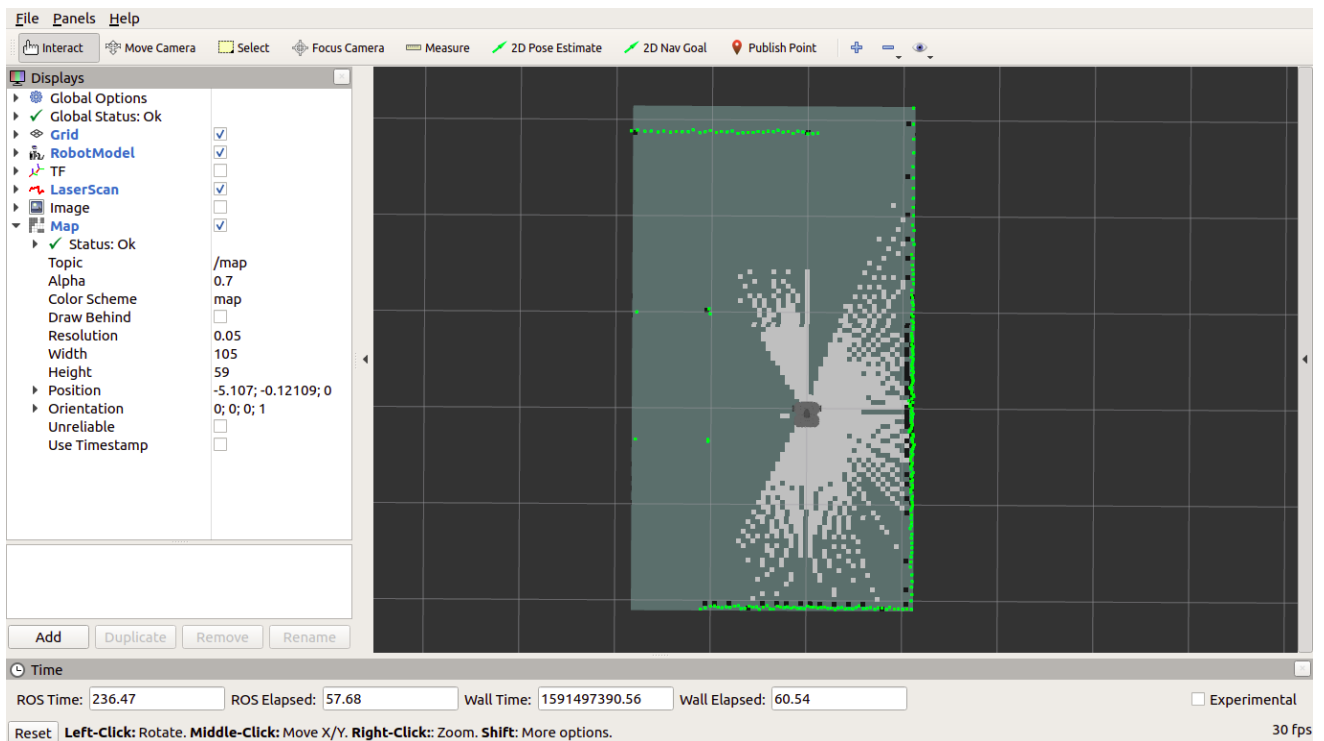
如果gazebo顺利启动并显示如下房间模型，终端输出有 `Physics dynamic reconfigure ready` 字样，则说明虚拟仿真环境启动完成。



打开新终端窗口，启动SLAM建图相关节点：

```
$ roslaunch turtlebot3_slam turtlebot3_slam.launch slam_methods:=karto
```

正确启动后，rviz显示内容如下：



打开新终端窗口，启动Turtlebot3机器人键盘运动控制节点：

```
$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

成功运行的截图如下：

```
guoqing@GQLU18:~$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
... logging to /home/guoqing/.ros/log/1a7b3f70-a867-11ea-90e7-28d244cd935f/roslaunch-GQLU18-7629.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://GQLU18:33649/

SUMMARY
=====
PARAMETERS
* /model: waffle
* /rostdistro: melodic
* /rosversion: 1.14.3

NODES
/
  turtlebot3_teleop_keyboard (turtlebot3_teleop/turtlebot3_teleop_key)

ROS_MASTER_URI=http://localhost:11311

process[turtlebot3_teleop_keyboard-1]: started with pid [7644]

Control Your TurtleBot3!
-----
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)

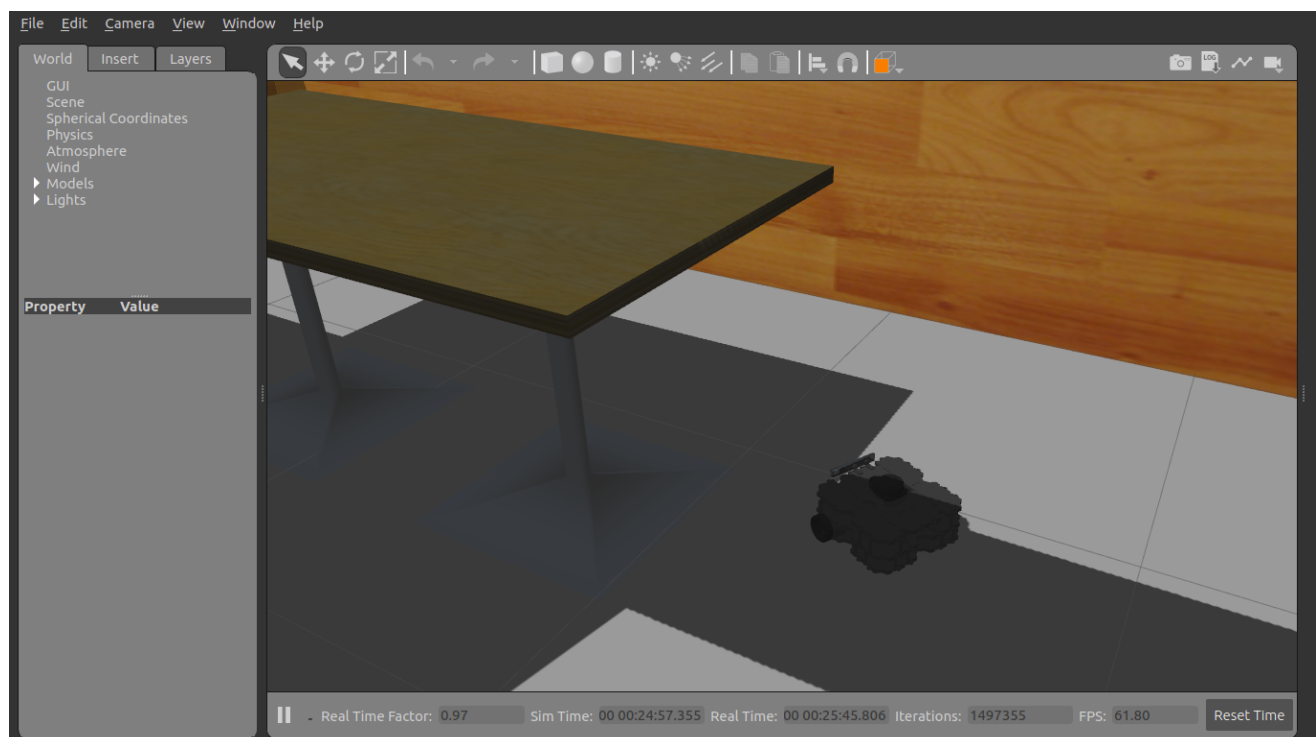
space key, s : force stop

CTRL-C to quit
```

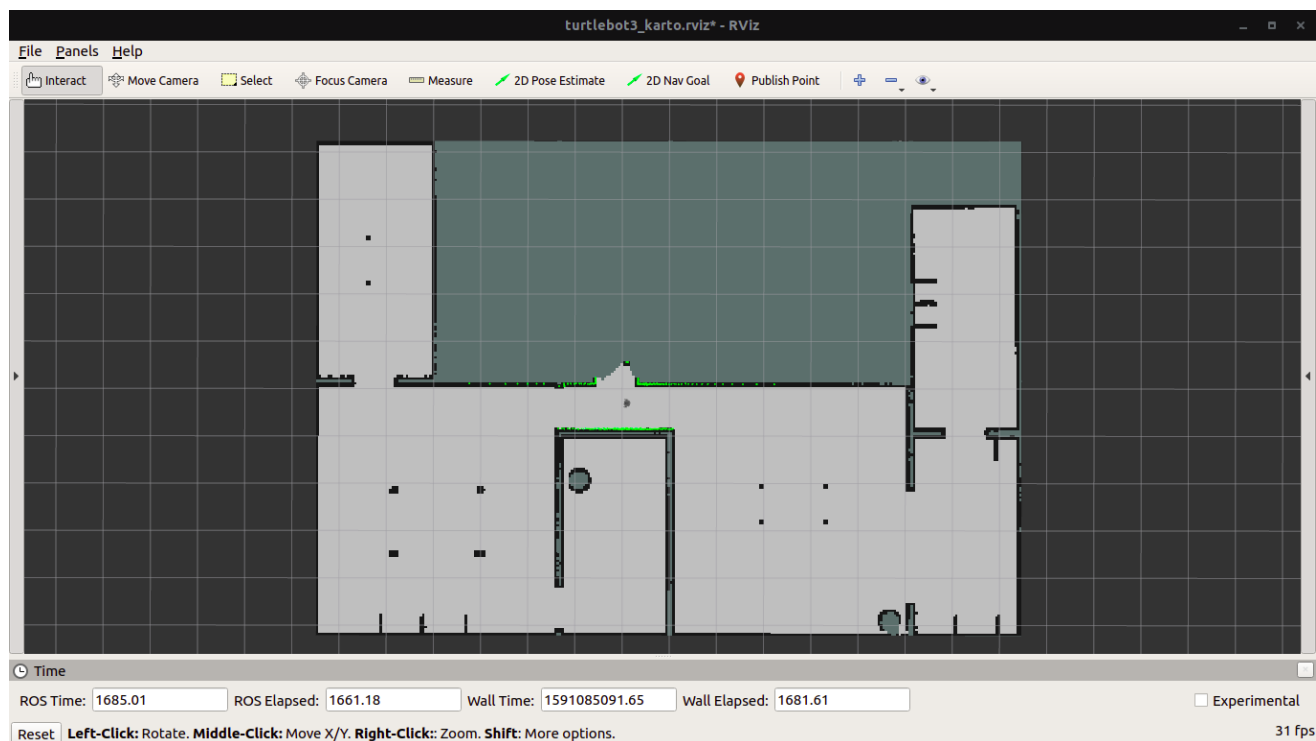
键盘控制的方式和大家刚接触ROS时运行 `turtlesim` 包下的 `turtle_teleop_key` 操作方式几乎相同：

- `w`：机器人加速/前进
- `x`：机器人减速/后退
- `a`：机器人加速左转
- `d`：机器人加速右转
- `s`：机器人急停

接下来同学们需要控制机器人在虚拟环境中走动，SLAM算法会不断根据激光雷达的数据构建环境地图。Gazebo仿真环境中的桌子下面一般是可以通行的，但是下面这个桌子支撑腿附近的区域除外：



本实验只需要构建室内部分的地图即可，最终构建成功的地图如下所示：



此时执行下面的命令将地图保存起来，讲义中将地图保存为 `~/mymap/housemap`：

```
$ roslaunch map_server map_saver -f ~/mymap/housemap
```

保存成功后，查看目录 `~/mymap`，可以看到map_server创建了以 `housemap` 为名的地图文件：



其中 `*.pgm` 即为以图片格式保存的地图，`*.yaml` 中保存了额外的地图信息如地图坐标系原点、分辨率等。

思考：

- 构建完成室内地图之后，尝试控制机器人在房间外围沿着墙壁走一圈，构建室外地图。出现了什么现象？你觉得为什么会出现这样的现象？

建图工作完成后，关闭上面启动的所有终端。

1.2 自主导航

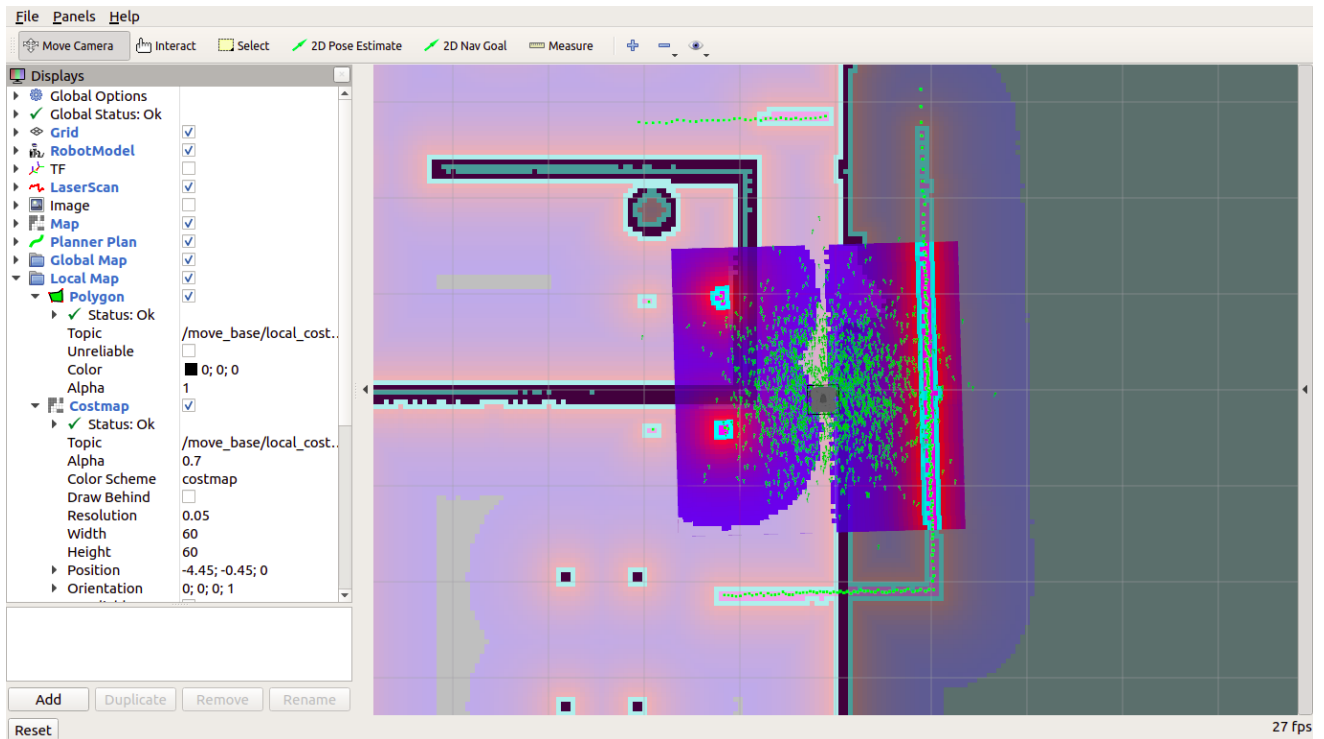
接下来进行Turtlebot3机器人的自助导航实验，首先打开终端窗口，启动虚拟仿真环境：

```
$ roslaunch turtlebot3_gazebo turtlebot3_house.launch
```

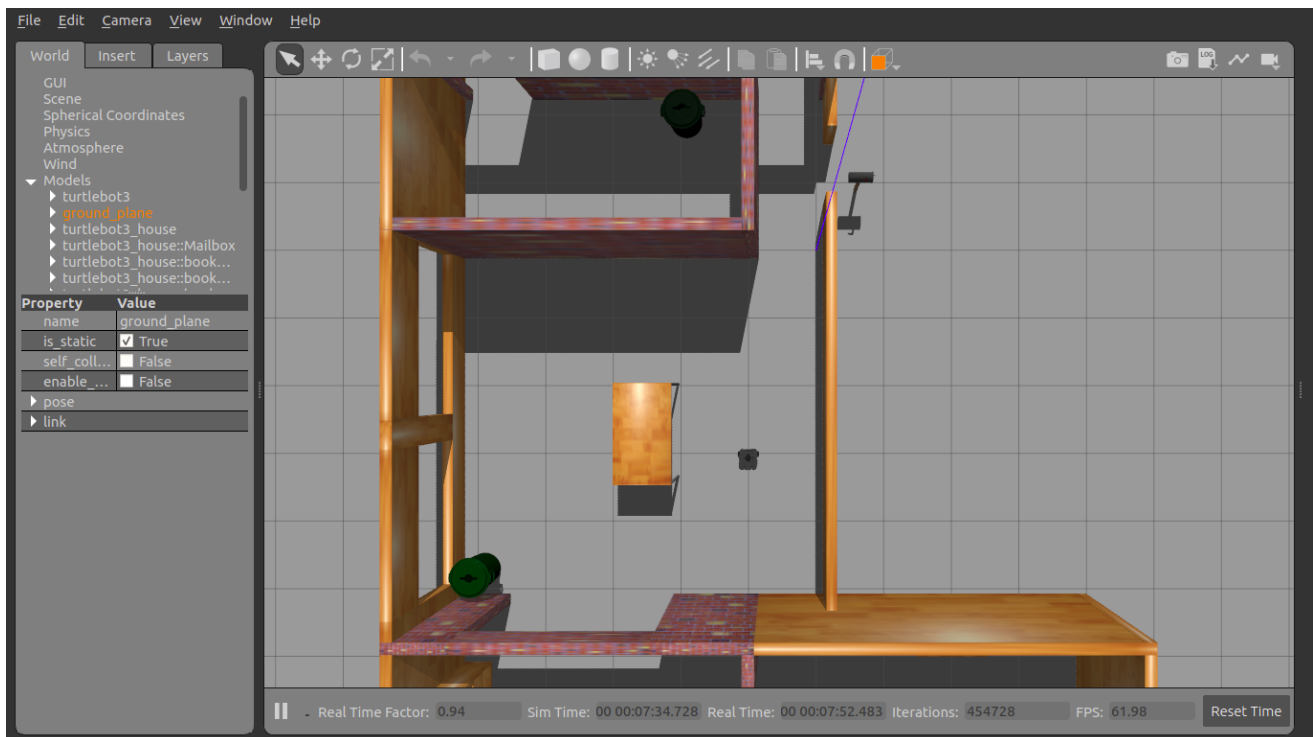
等待Gazebo启动成功后，执行下述命令启动Turtlebot3的导航功能包：

```
$ roslaunch turtlebot3_navigation turtlebot3_navigation.launch map_file:=/home/你的用户名/mymap/housemap.yaml
```

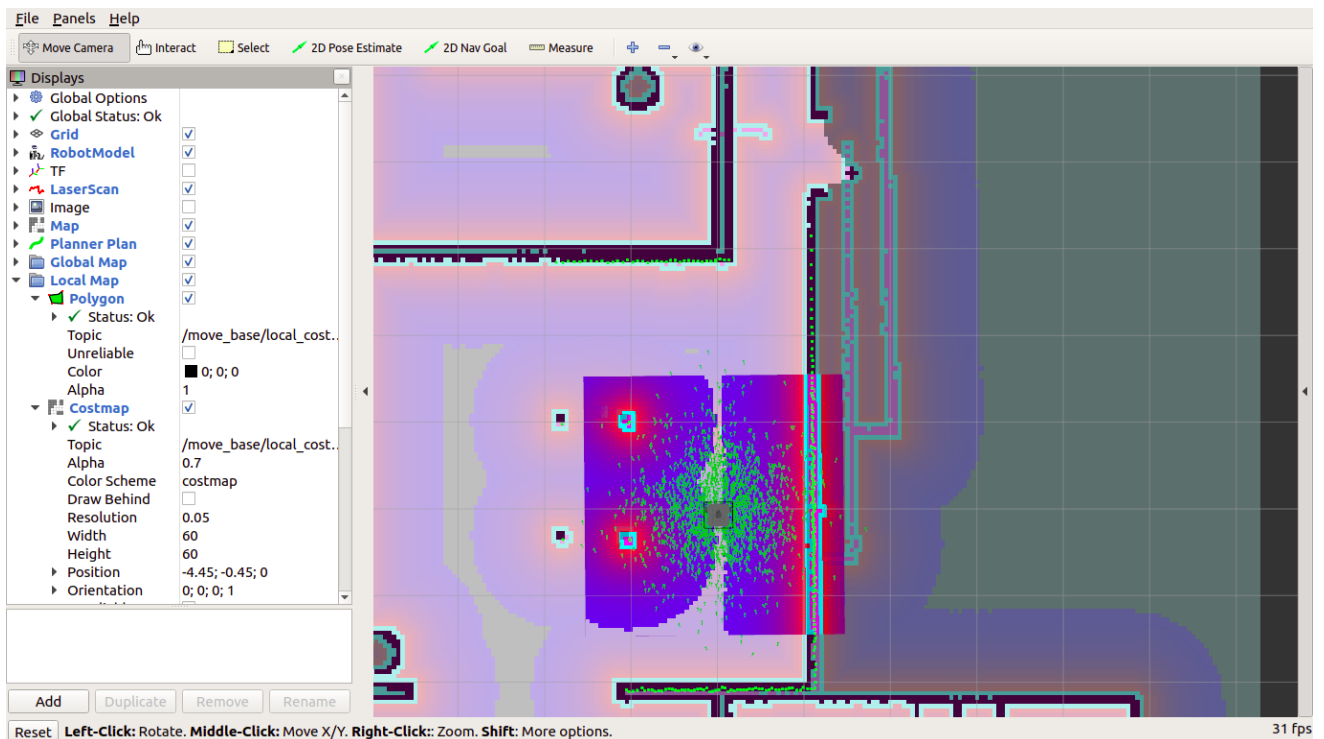
注意launch文件的参数 `map_file` 要改成你的地图保存位置，填写的是yaml文件；另外最好填写完整路径，`~/mymap/housemap.yaml` 不一定能够被正确识别，会出现“XXXX.yaml文件打不开”的错误提示。如果一切顺利，上述命令执行后rviz将会显示如下画面：



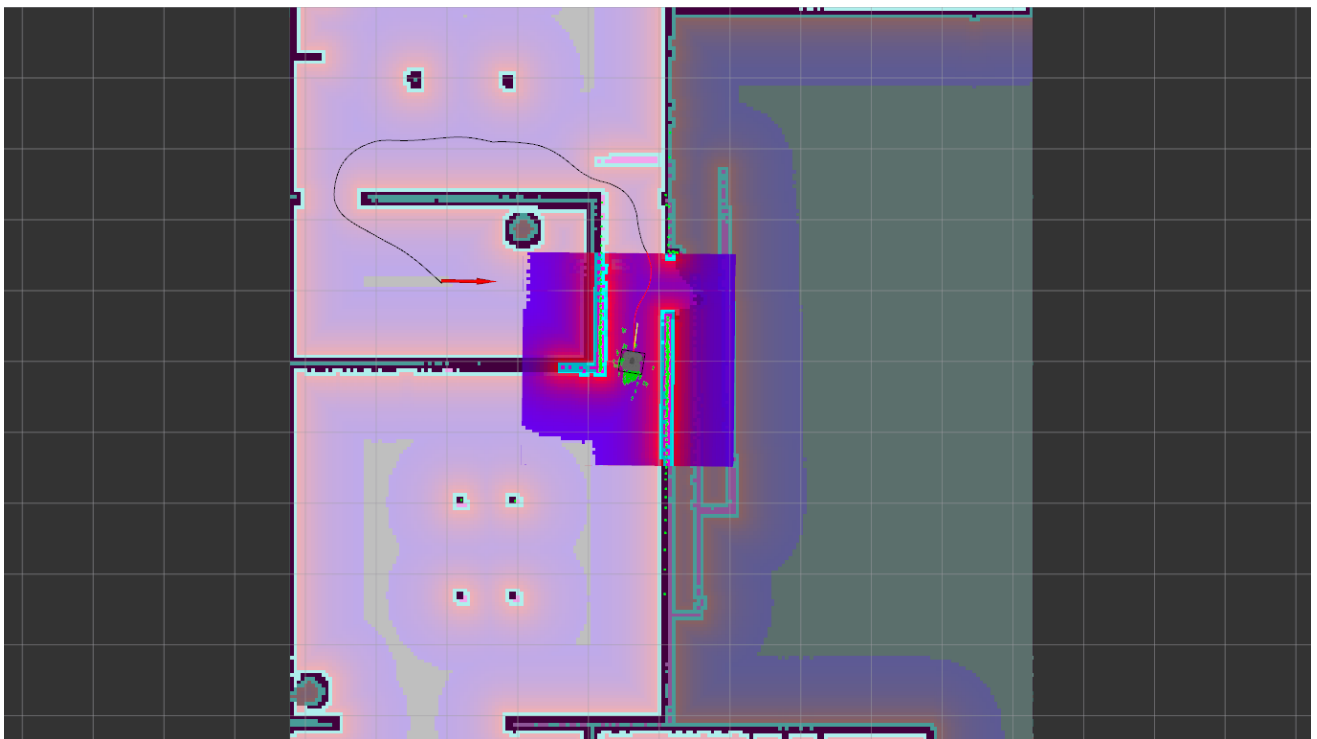
此时机器人位于地图坐标系 `/map` 的原点处，图中每个小网格代表1m的长度。注意此时gazebo中机器人的位置：



和rviz中的机器人位置并不相同。因此在执行导航实验前，需要首先设定机器人的初始位姿。在rviz窗，通过网格我们可以发现机器人的初始位姿在rviz显示位置处的左1、下3个网格处，点rviz窗口上端工具栏中的 2D Pose Estimate 按钮，然后在rviz中点击对应的网格点，按下鼠标左键后确定机器人的初始位姿，此时不要松开，拖动鼠标更改箭头的方向即可修改机器人的朝向，对于本次实验保持箭头垂直向上。完成机器人的初始位姿设置后的效果：



然后点击rviz工具栏的 2D Nav Goal 设定导航目标点，操作方式和设置机器人初始位姿类似。设定后机器人会计算一条全局路径，机器人将会基本沿着全局路径前进到目标点：



以上就是基本实验的内容。

思考：

- 导航实验依赖Cost Map实现导航。查看rviz左侧的显示部件并尝试打开或者关闭对应内容的显示，哪些内容是全局地图？哪些内容是局部地图？它们的相同和不同之处都有什么？
 - 在导航过程中，尝试在规划的全局路径上，放置Gazebo提供的基本模型（立方体、圆柱、球）作为动态障碍，观察路径规划器的应对措施；如果障碍物把路堵死，rviz的终端有何输出？
-

刘国庆

2020.06.07