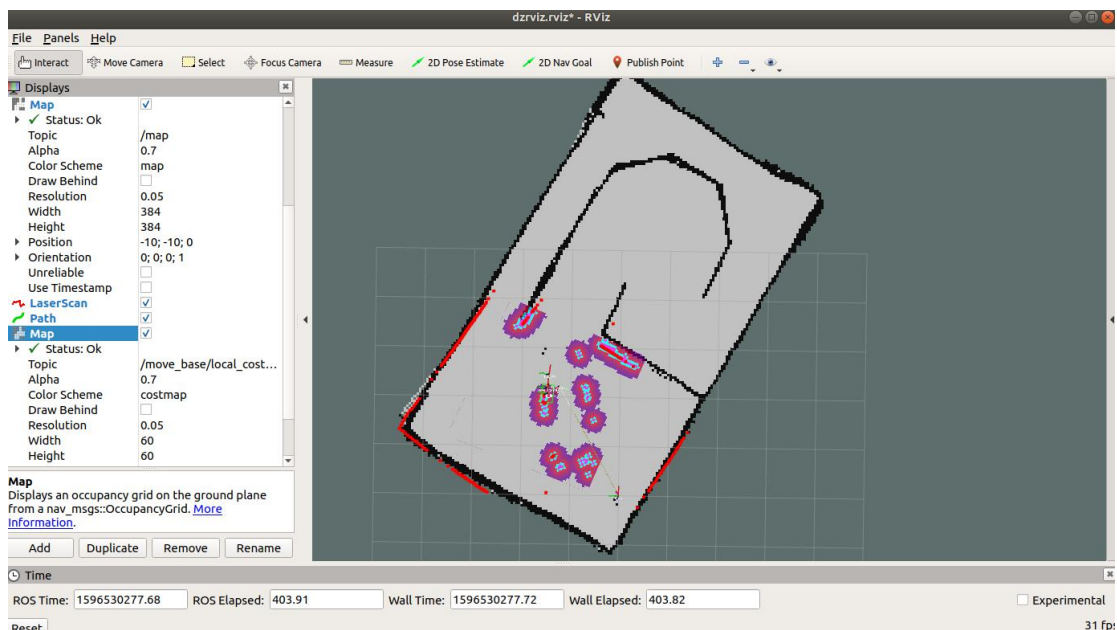


# 地图介绍

AI 航团队

## 1. 栅格地图

ROS 中的地图很好理解，就是一张普通的灰度图像，通常为 `pgm` 格式。这张图像上的黑色像素表示障碍物，白色像素表示可行区域，灰绿色是未探索的区域。如下图所示，SLAM 需要机器人在未知的环境中逐步建立起地图，然后根据地图确定自身位置，从而进一步定位。



在 SLAM 建图的过程中，你可以在 RViz 里看到一张地图被逐渐建立起来的过程，类似于一块块拼图被拼接成一张完整的地图。这张地图对于我们定位、路径规划都是不可缺少的信息。事实上，地图在 ROS 中是以 Topic 的形式维护和呈现的，这个 Topic 名称就叫做 `/map`，它的消息类型是 `nav_msgs/OccupancyGrid`。

## 2. 地图锁存

由于 `/map` 中实际上存储的是一张图片，为了减少不必要的开销，这个 Topic 往往采用锁存（latched）的方式来发布。什么是锁存？其实就是：地图如果没有更新，就维持着上

次发布的内容不变，此时如果有新的订阅者订阅消息，这时只会收到一个/map 的消息，也就是上次发布的消息；只有地图更新了（比如 SLAM 又建出来新的地图），这时/map 才会发布新的内容。锁存器的作用就是，将发布者最后一次发布的消息保存下来，然后把它自动发送给后来的订阅者。这种方式非常适合变动较慢、相对固定的数据（例如地图），然后只发布一次，相比于同样的消息不定的发布，锁存的方式既可以减少通信中对带宽的占用，也可以减少消息资源维护的开销。

### 3. nav\_msgs/OccupancyGrid

然后我们来看一下地图的 OccupancyGrid 类型是如何定义的，你可以通过 `rosmmsg show nav_msgs/OccupancyGrid` 来查看消息，或者直接 `roscd nav_msgs OccupancyGrid.msg` 来查看 `srv` 文件。

#### nav\_msgs/OccupancyGrid Message

File: `nav_msgs/OccupancyGrid.msg`

##### Raw Message Definition

```
# This represents a 2-D grid map, in which each cell represents the probability of
# occupancy.

Header header

#MetaData for the map
MapMetaData info

# The map data, in row-major order, starting with (0,0).    Occupancy
# probabilities are in the range [0,100].    Unknown is -1.
int8[] data
```

##### Compact Message Definition

```
std_msgs/Header header
nav_msgs/MapMetaData info
int8[] data
```

*autogenerated on Sun, 09 Feb 2020 03:18:28*

```
std_msgs/Header header #消息的报头
  uint32 seq
  time stamp
  string frame_id #地图消息绑定在TF的哪个frame上,一般为map
nav_msgs/MapMetaData info #地图相关信息
  time map_load_time #加载时间
  float32 resolution #分辨率 单位: m/pixel
  uint32 width #宽 单位: pixel
  uint32 height #高 单位: pixel
  geometry_msgs/Pose origin #原点
    geometry_msgs/Point position
      float64 x
      float64 y
      float64 z
    geometry_msgs/Quaternion orientation
      float64 x
      float64 y
      float64 z
      float64 w
  int8[] data #地图具体信息
```

这个 `srv` 文件定义了 `/map` 话题的数据结构, 包含了三个主要的部分: `header`, `info` 和 `data`。 `header` 是消息的报头, 保存了序号、时间戳、`frame` 等通用信息, `info` 是地图的配置信息, 它反映了地图的属性, `data` 是真正存储这张地图数据的部分, 它是一个可变长数组, `int8` 后面加了 `[]`, 你可以理解为一个类似于 `vector` 的容器, 它存储的内容有 `width*height` 个 `int8` 型的数据, 也就是这张地图上每个像素。