一、RRT Exploration

1. <https://blog.csdn.net/m0_37931718/article/details/90692686>
2. buaa是雷达？rrt是视觉？？ 答： 都是雷达。
3. 依赖库：

sudo apt-get install ros-kinetic-gmapping ros-kinetic-navigation ros-kinetic-kobuki ros-kinetic-kobuki-core ros-kinetic-kobuki-gazebo

sudo apt-get install python-opencv python-numpy python-scikits-learn

1. 启动的命令行

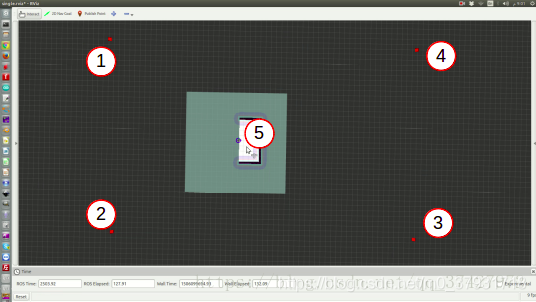
（1）Gazebo simulation和gmapping

roslaunch rrt\_exploration\_tutorials single\_simulated\_house.launch

（2）Start Exploration

roslaunch rrt\_exploration single.launch

1. 在rviz里点5个点



1. 去掉namespace的命令行：
   1. roslaunch rrt\_exploration\_tutorials single\_simulated\_house\_frontier.launch
   2. roslaunch rrt\_exploration single\_frontier.launch
2. subcribe的topic
   1. /map
   2. /move\_base\_node/global\_costmap/costmap
   3. /detected\_points
3. frontier\_exploration
4. 链接
   1. <http://wiki.ros.org/frontier_exploration>
   2. 结合turtlebot <https://github.com/130s/turtlebot_samples>
5. turtlebot\_samples的仿真环境有问题：

//export TURTLEBOT\_GAZEBO\_WORLD\_FILE=/home/zhjd/turtlebot\_buaa\_ws/src/IndoorRobotSimulation/gazebo\_worlds/square\_hall.world

1. gazebo和gmapping和rviz： roslaunch rrt\_exploration\_tutorials single\_simulated\_house\_frontier.launch

注：和原RRT仿真环境的区别：修改了topic，取消了前缀/robot\_1

1. frontier exploration： roslaunch turtlebot\_samples exploration\_gazebo\_RRTenv.launch

注：在rviz中maker的topic改为exploration\_polygon\_marker；

（3）报错：[ WARN] [1632388143.344389959, 2809.790000000]: Illegal bounds change, was [tl: (-17.750000, -17.750000), br: (4.650000, 17.450001)], but is now [tl: (-17.800000, -17.800000), br: (4.550000, 17.350001)]. The offending layer is explore\_costmap/explore\_boundary

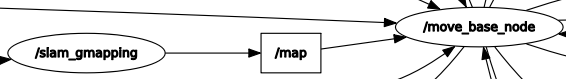
现象：给机器人呢规划了探索的范围，但是它到达目标点之后，就不动了？？

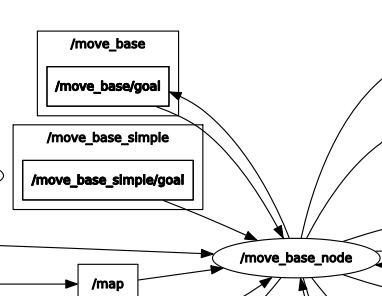
1. 其他：
2. explore\_lite
3. 链接：
4. 运行
   1. gazebo和gmapping和rviz： roslaunch rrt\_exploration\_tutorials single\_simulated\_house\_frontier.launch

注：和原RRT仿真环境的区别：修改了topic，取消了前缀/robot\_1

* 1. 运行： roslaunch explore\_lite explore.launch

1. 三个问题
   1. movebase+gmapping中能做到，手动指定目标点，路径规划。gmapping提供了相机位姿吗？？但我看到只有/map啊。

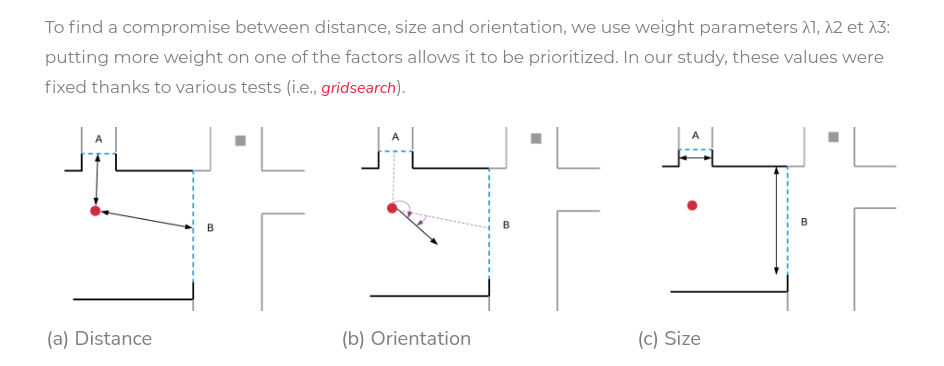


* 1. movebase对接收的goal的topic的名称没有规定吗？？ 
  2. 把expore.cpp看完，就可以不再用这个expore lite了。因为他的导航中位姿的准确性太差。但为什么位姿准确性这么差，是因为gmapping中的定位模块，比amcl效果要差？？能不能用cartography代替呢（cartographer中能提供位姿信息给movebase吗）？

1. cartographer
2. 命令行：
   1. 仿真环境和rviz： roslaunch rrt\_exploration\_tutorials single\_simulated\_house\_normal\_no\_gammping\_and\_movebase.launch
   2. cartographer编译方法： catkin\_make\_isolated --install --use-ninja
3. sudo boo <https://www.youtube.com/watch?v=tySFQuSL8Ls> up主没说 源码在哪？

<https://github.com/tyuownu/nearest_frontier_planner> 似乎是这个Nearest frontier planner。不确定

1. 原理介绍：<https://awabot.com/en/autonomous-exploration-method-frontiers/>



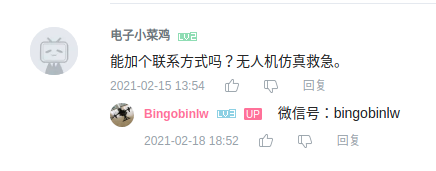
1. 要不要 直接用 rrt exploration（RRT到底比frontier planner好在哪？？）
2. <https://www.youtube.com/watch?v=45f4QF6Crtc> 方法很简陋。

开始建图前，会问操作者，建图的形式。

1. ICRA2021 三维的：Three-Dimensional Terrain Aware Autonomous Exploration for Confined Spaces
2. 三维的 <https://www.youtube.com/watch?v=1_nyFn5u2rs>
3. 三维无人机 <https://www.youtube.com/watch?v=tH2VkVony38>
4. movebase是否需要”机器人位姿”的思考: 经过测试可知，gmapping+movebase一样可以在rviz中，通过navigation goal，控制机器人的导航

|  |  |
| --- | --- |
| IMG_256 | ~~从左图可知,是用不到 机器人位姿的.~~  ~~所以有以下两种可能：~~  ~~1.movebase用不到机器人位姿。则机器人只打开movebase和gazebo，也是可以导航的。~~  ~~2.movebase内部自进行了 amcl。但也不对啊~~  （1）是否用到了 机器人位姿： 是的。  tf里面包含了 baselink相对map的坐标关系（即机器人位姿）。但没有amcl，这个是怎么生成的？？ （在slam中，也可以把机器人位姿以tf形式发送）。 |
| （2）tf是怎么生成的？  Odom信息的去向。第一个是广播了tf关系，即每次机器人移动以后，/odom坐标系和/base\_link的关系，(关于为什么要发布这tf关系，见第三篇博文)；第二个是将消息发布到odom topic上。这两个东西都将是move\_base需要的 |  |
| 1. gampping涉及到的tf变换   接受的：  sensor→baselink：使用 静态tf 发布。  baselink→odom：由odometry系统提供？？就是机器人平台吧。  发布的：  map→odom：the current **estimate of the robot's pose** within the map frame  map和odom是什么关系？？ |  |
| 1. 三步走： 2. cartographer配 movebase 3. cartographer的主动slam。cartographer的map能不能提取边界。 4. cartographer怎么和视觉slam（今儿quaric slam结合），主要是二者的地图怎么尺寸匹配，能联合优化自然更好。vio slam。（甚至是quric slam在cartographer基础上优化？怎么做，两者每一帧/观测的时间戳不同啊？） |  |

1. cartographer配 movebase
2. cartographer发布的位姿 是map→odom，还是map→baselink、
3. movebase接受的机器人位姿，是 <https://zhuanlan.zhihu.com/p/117381264>
4. 链接 ：[视觉slam建图导航中建立map-＞odom的tf关系](https://blog.csdn.net/jUst3Doit/article/details/110792571)
5. odom：里程计坐标系，这里要区分开odom topic，这是两个概念，一个是坐标系，一个是根据编码器（或者视觉等）计算的里程计。 但是两者也有关系，odom topic 转化得位姿矩阵是odom–>base\_link的tf关系。这时可有会有疑问，odom和map坐标系是不是重合的？可以很肯定的告诉你，机器人运动开始是重合的。但是，随着时间的推移是不重合的，而出现的偏差就是里程计的累积误差。那map–>odom的tf怎么得到?就是在一些校正传感器合作校正的package比如gmapping会给出一个位置估计（localization），这可以得到map–>base\_link的tf，所以估计位置和里程计位置的偏差也就是odom与map的坐标系偏差。所以，如果你的odom计算没有错误，那么map–>odom的tf就是0（最后一句话是不是有点问题）



1. explore lite的代码原理
2. ros consle是做什么的？

if (ros::console::set\_logger\_level(ROSCONSOLE\_DEFAULT\_NAME,

ros::console::levels::Debug)) {

ros::console::notifyLoggerLevelsChanged();

}

1. explore：
   1. Costmap2DClient costmap\_client\_;
   2. actionlib::SimpleActionClient<move\_base\_msgs::MoveBaseAction> move\_base\_client\_;
      1. costmap\_client\_.getCostmap(), 获取其中的map？？
      2. move\_base\_client\_.waitForServer();
   3. frontier\_exploration::FrontierSearch search\_;
   4. ros::Timer exploring\_timer\_;
   5. ros::Timer oneshot\_;
   6. ros::Time last\_progress\_;
   7. size\_t last\_markers\_count\_;
   8. private\_nh\_.advertise<visualization\_msgs::MarkerArray>("frontiers", 10);
      1. private\_nh\_.param("planner\_frequency", planner\_frequency\_, 1.0);
   9. const std::vector<frontier\_exploration::Frontier>& frontiers
   10. 这是启动了make plan吗？？

exploring\_timer\_ =

relative\_nh\_.createTimer(ros::Duration(1. / planner\_frequency\_),

[this](const ros::TimerEvent&) { makePlan(); });

1. frontier\_exploration::Frontier
   1. costmap\_2d::Costmap2D\* costmap
   2. 结构体：struct Frontier {
      1. middle 和 centroid 什么区别
   3. FrontierSearch函数： 再耽搁线程里，根据输入的costmap，生成frontier
   4. costmap\_->worldToMap(position.x, position.y, mx, my)) { /\* position不就是地图的坐标吗？ 而且地图的坐标和世界的坐标有什么区别？？？ 答：是不是 地图中有分辨率（栅格）的原因。 \*/
   5. // make sure map is consistent and locked for duration of search

std::lock\_guard<costmap\_2d::Costmap2D::mutex\_t> lock(\*(costmap\_->getMutex()));

* 1. fdf

1. costmap\_client（好难！！）
   1. 通过接受topic来构建 costmap。但什么多了一个 wartformessage

/\* initialize costmap \*/

costmap\_sub\_ = subscription\_nh.subscribe<nav\_msgs::OccupancyGrid>(

costmap\_topic, 1000,

[this](const nav\_msgs::OccupancyGrid::ConstPtr& msg) {

updateFullMap(msg);

});

ROS\_INFO("Waiting for costmap to become available, topic: %s",

costmap\_topic.c\_str());

auto costmap\_msg = ros::topic::waitForMessage<nav\_msgs::OccupancyGrid>(

costmap\_topic, subscription\_nh); /\* 为什么会接收两个 map的topic？？ \*/

updateFullMap(costmap\_msg);

* 1. map\_msgs::OccupancyGridUpdate 实时更新显示栅格地图
     1. <https://zhuanlan.zhihu.com/p/359342938>
     2. gazebo\_rrt官方图中。explore lite运行时的节点和topic。可以发现/map updates没有用到
     3. static\_cast<unsigned char> ？？
     4. void Costmap2DClient::updatePartialMap 是怎么更新地图的

size\_t idx = costmap\_.getIndex(x, y);

unsigned char cell\_cost = static\_cast<unsigned char>(msg->data[i]);

costmap\_data[idx] = cost\_translation\_table\_\_[cell\_cost];

* + 1. // lock as we are accessing raw underlying map

auto\* mutex = costmap\_.getMutex();

std::lock\_guard<costmap\_2d::Costmap2D::mutex\_t> lock(\*mutex);

* 1. msg到costmap的转变：

std::array<unsigned char, 256> init\_translation\_table()

{

std::array<unsigned char, 256> cost\_translation\_table;

// lineary mapped from [0..100] to [0..255]

for (size\_t i = 0; i < 256; ++i) {

cost\_translation\_table[i] =

static\_cast<unsigned char>(1 + (251 \* (i - 1)) / 97);

}

// special values:

cost\_translation\_table[0] = 0; // NO obstacle

cost\_translation\_table[99] = 253; // INSCRIBED obstacle

cost\_translation\_table[100] = 254; // LETHAL obstacle

cost\_translation\_table[static\_cast<unsigned char>(-1)] = 255; // UNKNOWN

return cost\_translation\_table;

}

疑问：

1. unsigned char a = static\_cast<unsigned char>(-1); /\* -1为什么转为无法好char之后， 等于255 \*/
2. 根据结果可知，从0～100转化为了0～255，但是之后开始循环。这是因为unsigned char是8个字节，最高只能到255。
   1. 体现ros service的句子：
      1. move\_base\_client\_.waitForServer(); 作为client，等待service的相应。
      2. action-server/action-client的定义方法： <https://blog.csdn.net/Will_Ye/article/details/79589319>
      3. 请问action-server和action-client的名称不一样 也能通信吗？如图 我运行的三个跟movebase相关的node，名称分别为move\_base、move\_base\_node、mobile\_base，但是它们相互之间却能通信（表现为gazebo中机器人可以被控制移动）。
      4. 变化过程:
         1. actionlib::SimpleActionClient<move\_base\_msgs::MoveBaseAction> move\_base\_client\_; move\_base\_client\_("move\_base")
         2. SimpleActionClient(const std::string & name, bool spin\_thread = true)

{ initSimpleClient(nh\_, name, spin\_thread); }

* + - 1. initSimpleClient(nh\_, name, spin\_thread)

{ ac\_.reset(new ActionClientT(n, name, &callback\_queue)); }

* + - 1. new ActionClientT(n, name, &callback\_queue)

ActionClient<ActionSpec> ActionClientT

* + - 1. ros::NodeHandle n\_; n\_(name)
      2. 测试下 https://www.cnblogs.com/21207-iHome/p/8297099.html 的程序.更改下node的名字.

1. ROS actionlib学习
2. 链接 <https://www.cnblogs.com/21207-iHome/p/8297099.html>
3. 优势：假如某个请求执行时间很长，在此期间用户想查看执行的进度或者取消这个请求的话，service机制就不能满足了，但是actionlib可满足用户这种需求。例如，控制机器人运动到地图中某一目标位置，这个过程可能复杂而漫长，执行过程中还可能强制中断或反馈信息，这时actionlib就能大展伸手了。
4. ActionClient 和ActionServer之间使用action protocol通信，action protocol就是预定义的一组ROS message，这些message被放到ROS topic上在 ActionClient 和ActionServer之间进行传实现二者的沟通。
5. action有三种消息：goal, result, and feedback
6. 10月8日的安排:
   1. 导入一张/map,实现这个功能。
      1. global pose：

Pose\_()

: position()

, orientation() {

}

* 1. 找到边界的挑选准则,是直接选择最近的吗？？
     1. FrontierSearch::searchFrom是按照宽度 搜索，所以frontier序列是按照距离机器人的远近排序的。
     2. Explore::makePlan是从frontier序列的begin向end搜索，将第一个不在黑名单中的frontier->centroid作为goal。
     3. 所以 边界的挑选准则,就是选择最近的
  2. 怎么把,物体信息，添加到frontier的挑选中。
     1. txt版的物体信息
     2. 程序中，物体信息：位姿+scale
     3. 距离是一个数字。物体信息怎么变为 数字呢？？
  3. eplloidslam的运行： opencv的问题
     1. 都用cv3。 尝试用cv3编译system。但不确定是否可行。问题是：
        1. darknet ros 的opencv版本不确定
        2. kinect v2 的opencv版本：
        3. 使用cv2编译/home/zhjd/thirdparty/vision\_opencv/cv\_bridge。从而修改cv bridge的头文件位置
           1. 在/home/zhjd/catkin\_ws/src/EllipsoidSLAMv1/Example/ROS/EllipsoidSLAM/CMakeLists.txt中修改

set(cv\_bridge\_DIR "/home/zhjd/thirdparty/vision\_opencv/cv\_bridge/build")

从而修改了头文件的位置。

* + - 1. 修改/opt/ros/kinetic/share/cv\_bridge/cmake/cv\_bridgeConfig.cmake 中使用的opencv的头文件include和库文件lib的位置 [链接](https://blog.csdn.net/weixin_41632448/article/details/103019748?utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-3.no_search_link&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-3.no_) 。 从而重新编译了cv\_bridge的so文件（库文件）。
         1. 这里我在/usr/local中安装了opencv2.4.10（lib中是so文件，include中头文件） [安装日志](https://user.qzone.qq.com/995333216/blog/1634568717)。 以便于修改cv\_bridgeConfig.cmake。
      2. 但是ros\_ellipsoid\_slam\_real还有用到了一些kinetic的cv3.但似乎不影响
  1. 椭球体 slam和 栅格 地图 ，怎么 尺度对齐。
  2. 除了goal的位置确定,增加位姿确定。可以使面向未知区域，也可以是面向物体。

1. 展望未来：激光slam和quaric slam 的联合优化。现在第一步，还是先完全使用cartographer的位姿估计。

|  |  |
| --- | --- |
| gazebo\_rrt官方图中。explore lite运行时的节点和topic。  可以发现/map updates没有用到。 |  |