

嵌入式物联网综合实践

——智能车竞赛部分

背景

- 以智能汽车为研究对象的创意性科技竞赛，是面向大学生的一种具有探索性工程的实践活动
- 室外光电组比赛主要学习和考核以ROS为主的机器人相关知识、非常符合目前机器人产业界的人才需求

实践内容

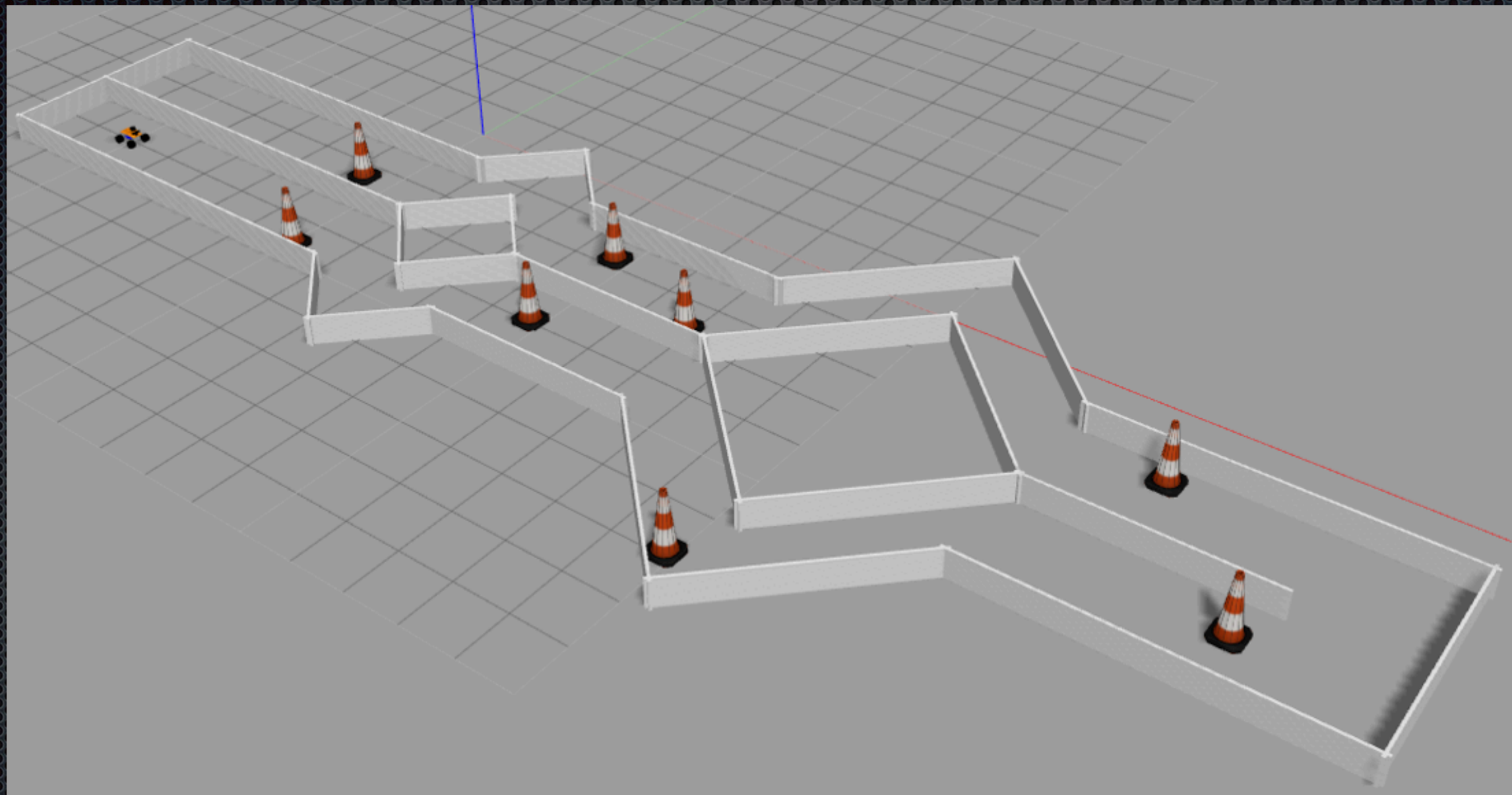
- 基于ROS，在已构建赛道地图中，通过自主导航算法实现无人车完成从起点到终点的运动。



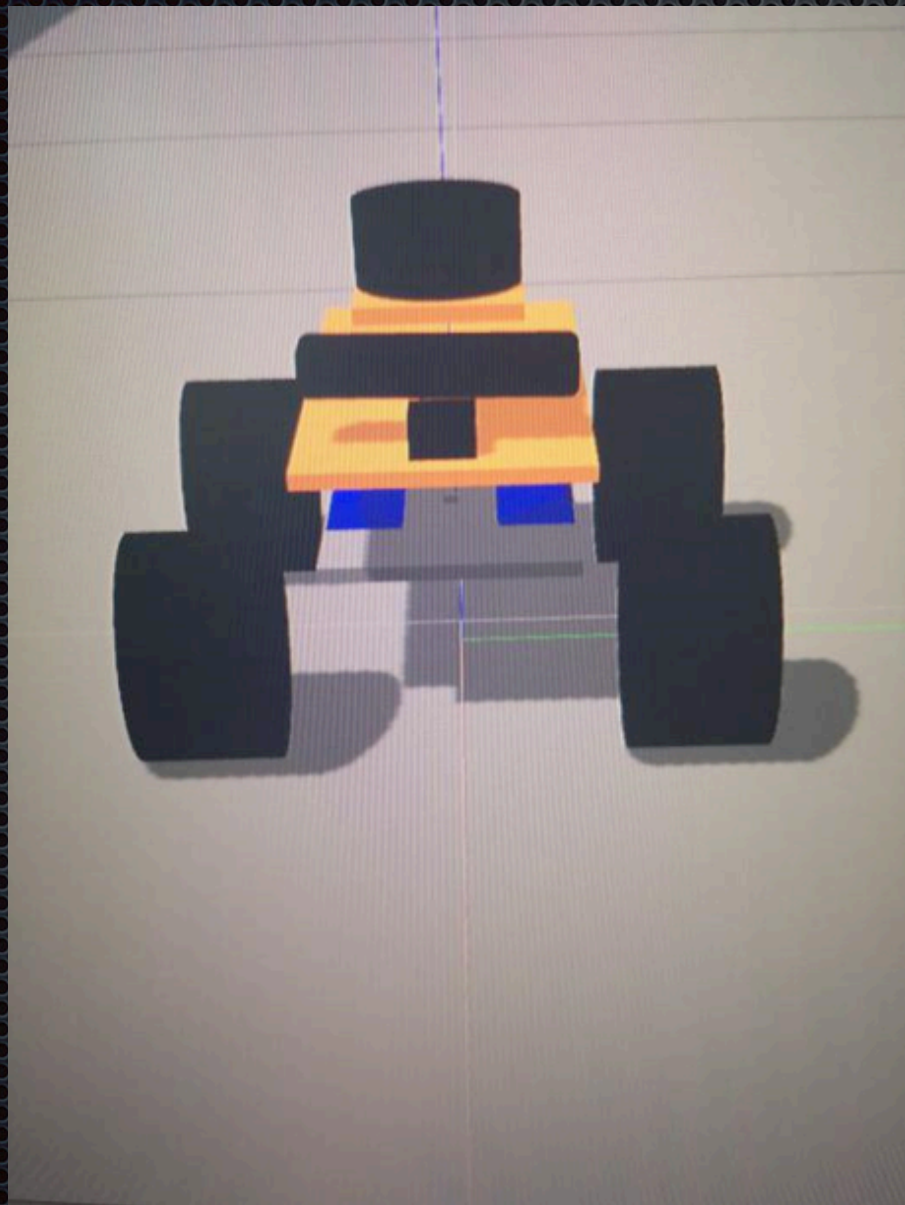
方案简介

- 为保证线上比赛的公平性，智能车室外光电组线上仿真比赛平台统一用Gazebo。
- 赛道模型和无人车三维模型于赛前统一提供。
- 线上比赛需要先把赛道模型导入Gazebo，采用ROS中建地图的方式构建赛道地图，通过自主导航算法实现无人车从起点到终点的运动。
- 仿真平台的传感器可以使用IMU，激光雷达或摄像头。
- 仿真平台自主导航算法不限。

仿真赛道



仿真车模型



仿真车模型自带传感器如下：

- IMU
- 激光雷达
- 深度摄像头

仿真车模型如下方面禁止修改：

- 尺寸大小
- 自带传感器参数
- 无人车质量
- 无人车中各部位的转动惯量矩阵
- 碰撞系数

关于障碍物

- 线上仿真比赛赛道中会有锥桶等障碍物，车模运行时要避开障碍物，否则会有相应处罚，障碍物位置是随机的，会在赛前公布。
- 赛道中的锥桶采用的是Gazebo models中的标准锥桶模型。

比赛违规说明

- 1、车模碰触到锥桶，加罚5s
- 2、车模碰触到赛道围栏，加罚5s
- 3、车模碰触到锥桶或赛道围栏后停止运行，计比赛失败
- 4、车模在赛道中停止运行，计比赛失败
- 5、为了考查参赛队员在传感器数据融合方面的能力，在Gazebo仿真比赛中，禁止使用Gazebo直接发布无人车的精准位置消息（odometry）。

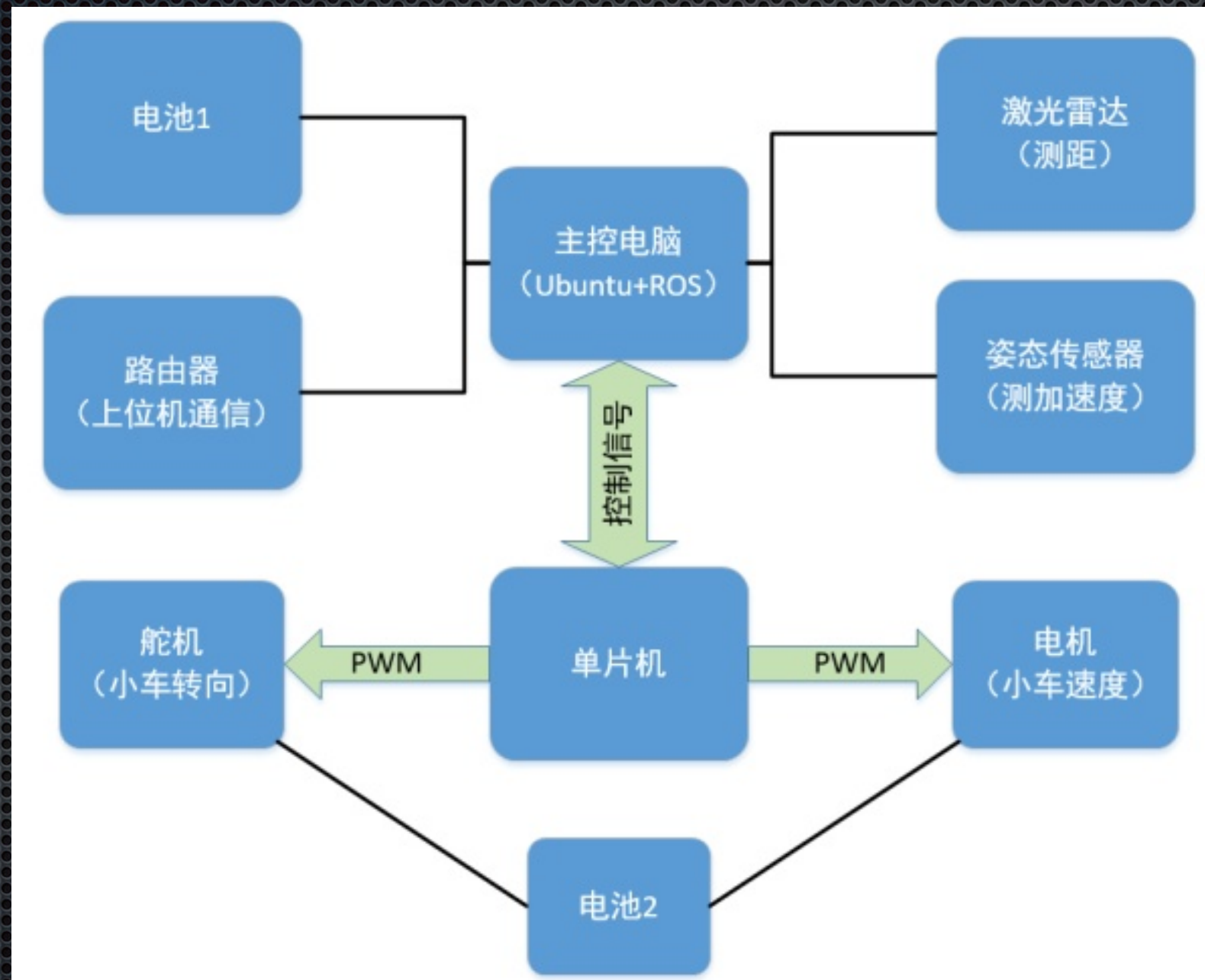
研究的关键技术

- ✦ 1. ROS小车平台的搭建：解决怎么做车的问题
- ✦ 2. 定位：解决车在哪的问题
- ✦ 3. SLAM建图：解决赛道地图信息的问题
- ✦ 4. 全局路径规划：解决最优路径规划的问题
- ✦ 5. 本地路径规划：解决动态躲避障碍物的问题

ROS

- 1. ROS的核心是一个分布式、低耦合的通讯机制；
- 2. ROS提供多种机器人开发工具，可以快速实现数据可视化、机器人仿真等功能；
- 3. ROS开源社区中包含大量机器人应用功能，可以帮助我们快速开发功能原型；
- 4. ROS已经成为一个庞大的生态系统，包含机器人领域的方方面面，同时也得到了越来越多第三方工具的支持，为机器人开发提供了系统化的解决方案。

ROS小车的实现架构



定位

- 小车的定位有很多种方法，常见的方法如下：
 - 1. 里程计定位：通过编码器获取电机/车轮在单位时间内的旋转圈数，计算得到速度，再积分得到位置；
 - 2. 姿态传感器：姿态传感器可以感知小车的线加速度和角加速度，利用积分同样可以得到速度和位置；
 - 3. 视觉定位：通过激光检测小车和周围物体的距离变化，计算位置；

- 在ROS中也常用amcl来实现定位。amcl是一种自适应（或kld采样）的蒙特卡罗定位方法，这是一种概率统计方法，针对已有地图使用粒子滤波器跟踪一个机器人的姿态。



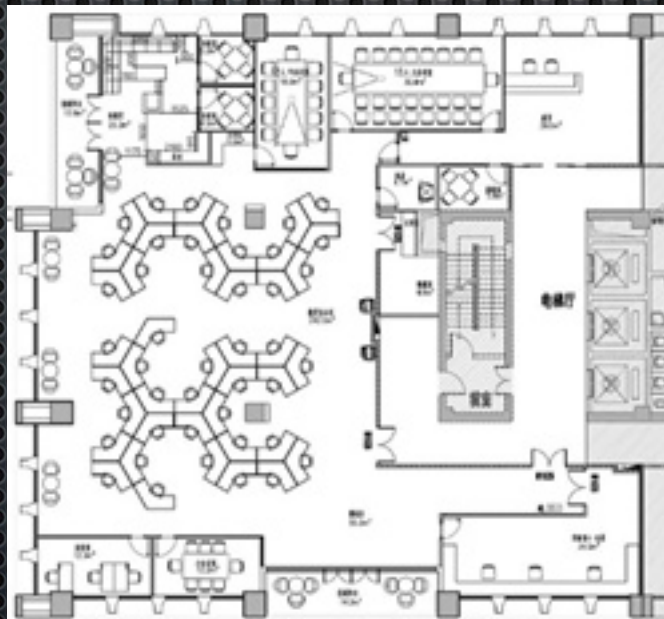
参考链接：

[amcl – ROS Wiki](#)

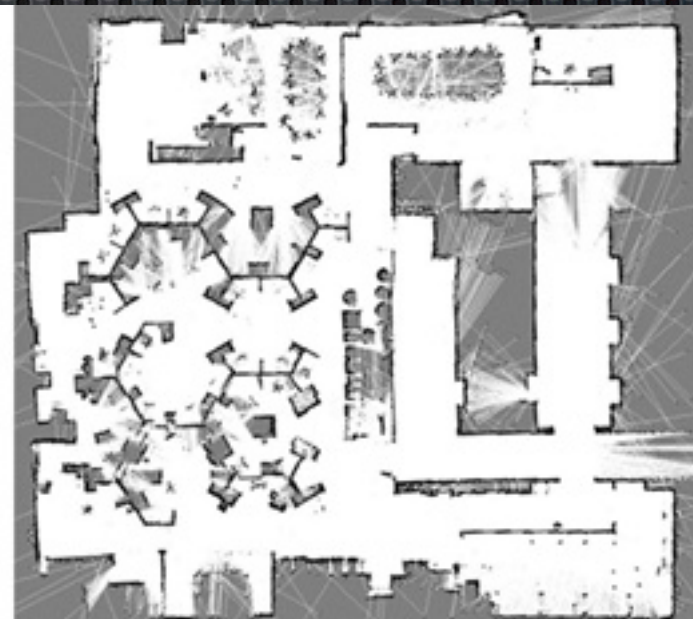
[ROS探索总结（十五）-amcl（导航与定位） - 古月居](#)

SLAM建立环境地图

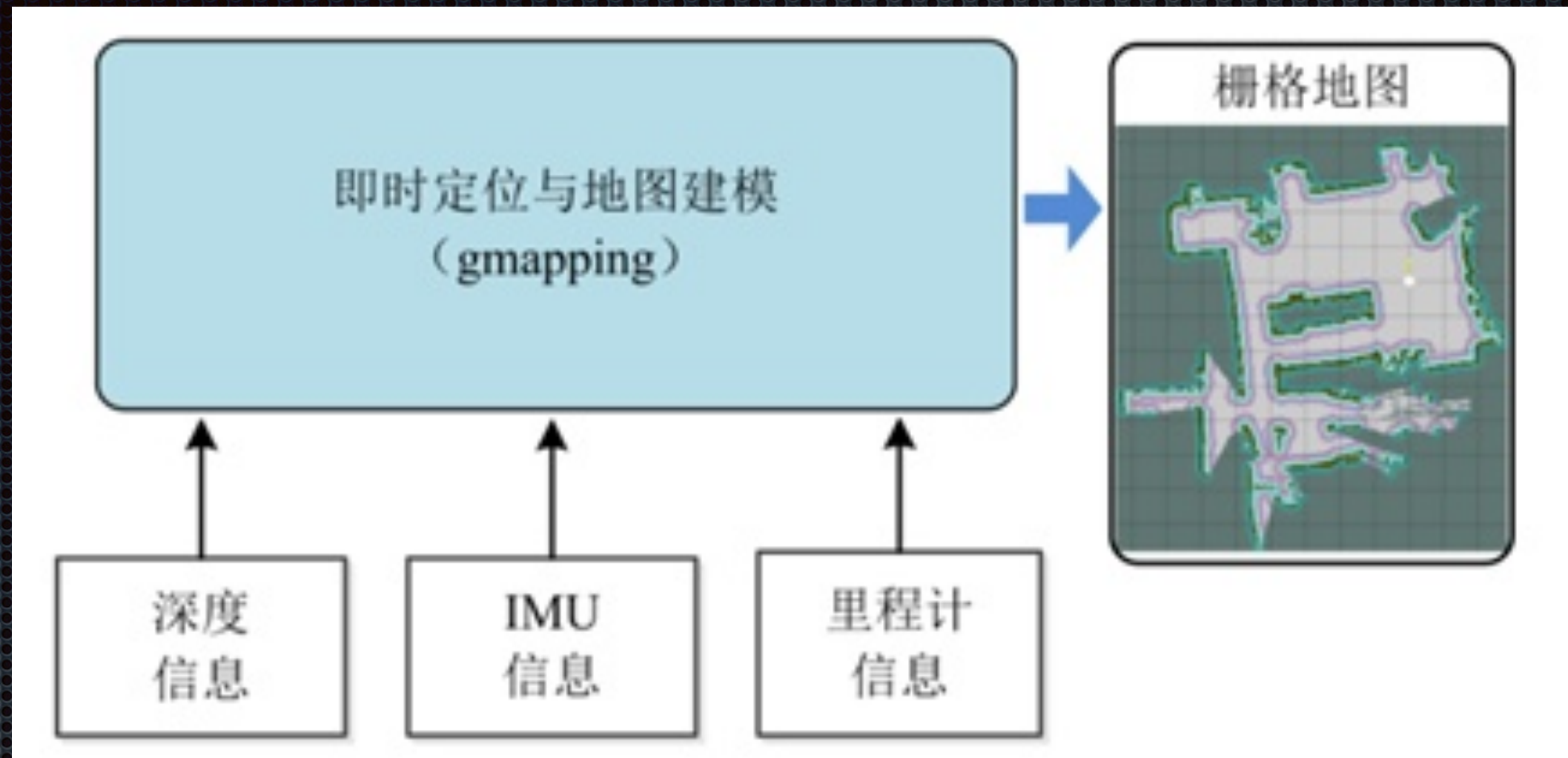
- SLAM问题可以描述为：机器人在未知环境中从一个未知位置开始移动，在移动过程中根据位置估计和地图进行自身定位，同时建造增量式地图，实现机器人的自主定位和导航。
- ROS 开源社区中汇集了多种SLAM 算法，都可以直接使用或进行二次开发，其中最为常用和成熟的是 gmapping 功能包。
- gmapping 包集成了 Rao-Blackwellized 粒子滤波算法，为开发者隐去了复杂的内部实现。



设计图



SLAM产生的高精度地图



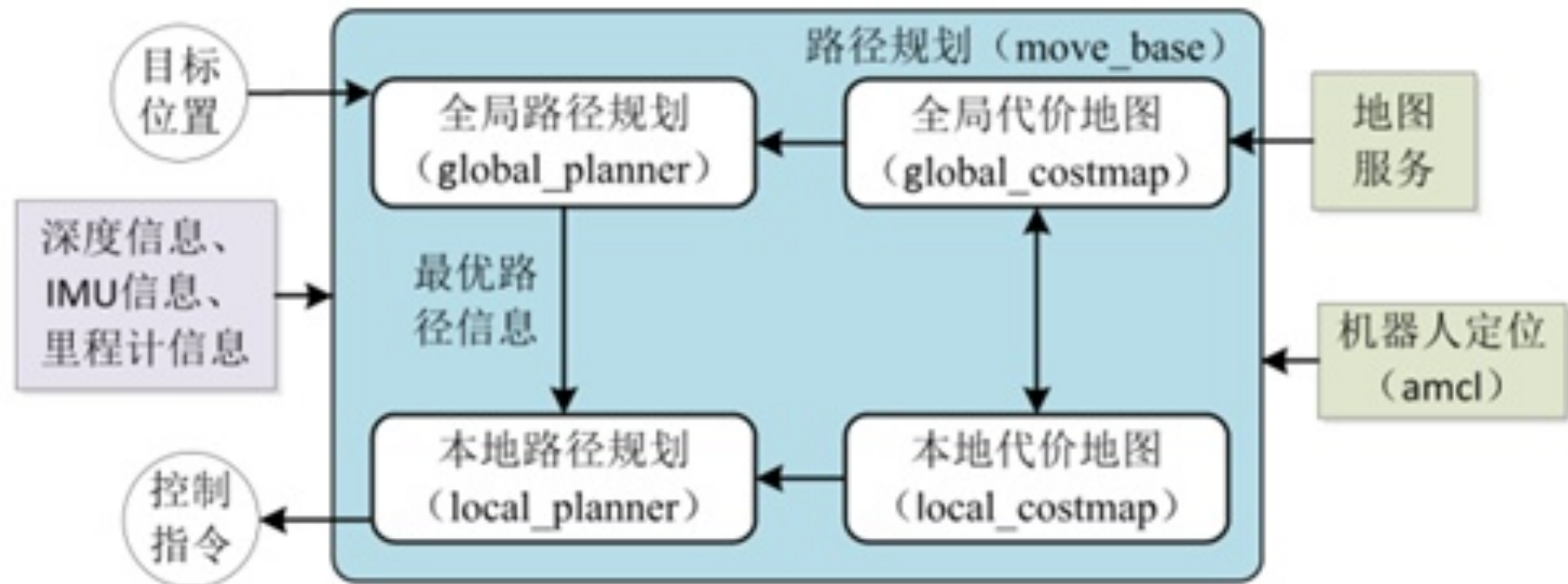
- gmapping功能包需要订阅机器人的深度数据、IMU 信息和里程计信息，同时完成一些必要参数的配置，即可创建并输出基于概率的二维栅格地图。

参考链接：

OpenSLAM.org
[gmapping – ROS Wiki](http://wiki.ros.org/gmapping)

自主导航

- 想实现ROS小车的自主导航，用ROS中路径规划的功能包——move_base就可以



参考

- https://blog.csdn.net/qq_40359328/article/details/95057229?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-1.channel_param&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-1.channel_param
- <https://cloud.tencent.com/developer/article/1423904>
- <http://fast.scripts.mit.edu/racecar/courses/>
- <https://github.com/DJTobias/Cherry-Autonomous-Racecar>
- <https://mit-racecar.github.io/>

