

1. 总结

GMapping 适合中小规模场景建图。

SLAM Toolbox 适合多传感器融合和动态环境下的建图。

Cartographer 适合大规模场景建图。

高精度建图根据场景大小选用 GMapping 或 Cartographer 。

2. GMapping

GMapping 是一种经典的基于激光雷达的 SLAM 算法，主要应用于机器人领域，采用改进的粒子滤波方法进行位姿估计，并通过扫描匹配和粒子滤波逐步构建环境地图

- **适用范围：**中小规模的室内场景，对计算资源需求较低，适合教育和研究用途。
- **优点：**
 - Gmapping 可以实时构建室内地图，在构建小场景地图所需的计算量较小且精度较高。
- **缺点：**
 - 随着场景增大所需的粒子增加，因为每个粒子都携带一幅地图，因此在构建大地图时所需内存和计算量都会增加。因此不适合构建大场景地
 - 没有回环检测，因此在回环闭合时可能会造成地图错位，虽然增加粒子数目可以使地图闭合但是以增加计算量和内存为代价

3. SLAM Toolbox

SLAM Toolbox 是 ROS 中的一个模块化工具箱，支持多种传感器（如激光雷达、RGB-D 相机等），提供实时建图和多传感器融合功能

- **适用范围：**适用于动态环境和工业、商业场景。高精度需求适用同步 SLAM 模式。资源受限及需要快速响应场景适用异步 SLAM 模式
- **优点：**
 - 支持同步（Sync SLAM）和异步（Async SLAM）两种模式，同步模式

通过每次数据更新都重建高精度地图，适合对精度要求高的场景。

- 包含闭环检测、地图发布与保存、参数配置等模块，支持灵活调整地图分辨率、扫描范围等参数
- 支持地图的持续更新与合并，可处理动态环境变化，减少重复扫描。
- 基于 Karto 核心扫描匹配器，结合弹性姿态图定位（EPL）方法，提升动态场景下的定位精度
- **缺点：**
 - 同步 SLAM 需占用大量 CPU 资源，可能影响其他任务；异步 SLAM 虽降低延迟，但地图精度相对较低。
 - 在 2K 以上大范围场景中，地图可能出现明显扭曲，定位精度受限。

4. Cartographer

Cartographer 是基于图优化的方法建图算法，其主要通过闭环检测来消除构图过程中产生的累积误差。Cartographer 专为大场景优化，采用分层地图表示和优化后端

- **适用范围：**适用于大规模环境（如商场、机场、校园等），在动态环境和高精度地图需求方面表现优异。
- **优点：**
 - 累计误差低，能天然的输出协方差矩阵，后端优化的输入项。成本较低的雷达也能跑出不错的效果
 - 没有 imu 和 odom，只有雷达也可以建图，还可以手持建图。
- **缺点：**

内存占用较大，算法体量较大，需要花上很久的时间才能稍微看得懂。