**SLAM 入门教程**

**版本0.1**

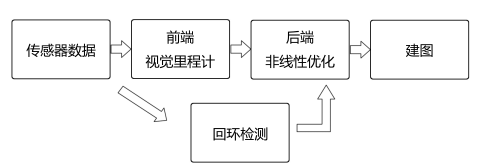
**一 SLAM介绍**

SLAM 是 Simultaneous Localization and Mapping 的缩写,中文译作“同时定位与地图构建”。它是指搭载特定传感器的主体,在没有环境先验信息的情况下,于运动过程中建立环境的模型,同时估计自己的运动 。如果这里的传感器主要为相机,那就称为“视觉 SLAM”。传感器还可以是激光Lidar等等。

SLAM 的目的是解决“定位”与“地图构建”这两个问题。也就是说,一边 要估计传感器自身的位置,一边要建立周围环境的模型。我们希望实时地根据运动和传感器信息来构建环境地图,并且推断机器人的位置。

**二 视觉SLAM流程描述**

经典的视觉SLAM流程图如下图所示:



我们把整个视觉 SLAM 流程分为以下几步:

1. 传感器信息读取。在视觉 SLAM 中主要为相机图像信息的读取和预处理。如果在机器人中,还可能有码盘、惯性传感器等信息的读取和同步。

2. 视觉里程计 (Visual Odometry, VO)。视觉里程计任务是估算相邻图像间相机的运动,以及局部地图的样子。VO 又称为前端(Front End)。

3. 后端优化(Optimization)。后端接受不同时刻视觉里程计测量的相机位姿,以及回环检测的信息,对它们进行优化,得到全局一致的轨迹和地图。由于接在 VO 之后,又称为后端(Back End)。

4. 回环检测(Loop Closing)。回环检测判断机器人是否曾经到达过先前的位置。如果检测到回环,它会把信息提供给后端进行处理。

5. 建图(Mapping)。它根据估计的轨迹,建立与任务要求对应的地图。

**三 SLAM问题的计算原理**

SLAM问题的通用**运动方程**为:



这里 u\_k 是运动传感器的读数(有时也叫输入),w\_k 为噪声。注意到,我们用一个一

般函数 f 来描述这个过程,而不具体指明 f 的作用方式。这使得整个函数可以指代任意

的运动传感器,成为一个通用的方程,而不必限定于某个特殊的传感器上。

 相对对应于运动方程的**观测方程**为:

当机器人在 x\_k 位置上看到某个路标点 y\_j ,产生了一个观测数据 z\_{k,j}。同样,我们用一个抽象的函数 h 来描述这个关系, 这里 v\_{k,j} 是这次观测里的噪声。

对于不同的传感器和机器人的形态, 上述两个方程有不同的参数化表达形式。由此SLAM问题变成了一个状态估计问题: 如何通过带有噪声的运动测量的读数 u,以及传感器的读数 z ,求解定位问题(估计 x)和建图问题(估计 y)。

**四 ROS navigation package 下SLAM的应用**

请先在PC端预安装Ubuntu, ROS, 机器人端预安装ROS, 保持PC端与机器人的通讯畅通。

**步骤一**

根据如下步骤在PC端下载rviz 配置文件

开启一个termminal

cd ~/catkin\_ws/src

git clone https://github.com/LFZ1994/robot\_navigation.git

cd ~/catkin\_ws

catkin\_make

**步骤二**

然后在机器人端执行 slam 的 launch 文件

roslaunch robot\_navigation robot\_slam\_laser.launch

**步骤三**

在 PC 端执行 rviz 图形化监控程序

roslaunch robot\_navigation slam\_rviz.launch

(注意: 如果此处报错, 请尝试运行 source ~/catkin\_ws/devel/setup.sh)

**步骤四**

让机器人到处走走, 将得到地图如下

