

SLAM14讲—11后端2

11.1 位姿图 (Pose Graph)

11.1.1 Pose Graph的意义

- 位姿图的构建：位姿节点之间的边由两个关键帧之间通过特征匹配之后得到的运动估计来给定初始值。一般初始估计完成，我们不再优化那些路标点的位置，而只关心所有的相机位姿之间的联系。机器人在更长的时间和空间中运动时，必须考虑一些解决方式：要么像滑动窗口法那样，丢弃一些历史数据；要么像Pose Graph的做法那样，舍弃对路标点的优化，只保留Pose之间的边，使用Pose Graph。

11.1.2 Pose Graph的优化

Pose Graph的节点表示相机位姿，以 ξ_1, \dots, ξ_n 来表示。而边，则是两个位姿节点之间相对运动的估计，该估计可能来自于特征点法或直接法。

特征点法或直接法，但不管如何，我们估计了，比如说 ξ_i 和 ξ_j 之间的一个运动 $\Delta\xi_{ij}$ 。该运动可以有若干种表达方式，我们取比较自然的一种：

$$\Delta\xi_{ij} = \xi_i^{-1} \circ \xi_j = \ln \left(\exp \left((-\xi_i)^\wedge \right) \exp \left(\xi_j^\wedge \right) \right)^\vee, \quad (11.1)$$

或按李群的写法：

$$\Delta T_{ij} = T_i^{-1} T_j. \quad (11.2)$$

按照图优化的思路来看，实际当中该等式不会精确地成立，因此我们设立最小二乘误差：

$$\min_{\xi} \frac{1}{2} \sum_{i,j \in \mathcal{E}} e_{ij}^T \Sigma_{ij}^{-1} e_{ij}. \quad (11.11)$$

可用G-N、L-M等方法来求解。

11.2 实践：位姿图优化

- g2o_viewer使用方法

```
1 g2o_viewer filename.g2o
```

需要安装gtsam库

```
1 git clone https://bitbucket.org/gtborg/gtsam.git
2
3 //安装tbb库
4 sudo apt-get install libtbb-dev
```

头文件在/usr/local/include中，库文件在/usr/local/lib中。库文件仅由一个libgtsam.so组成

