创建 trajectory的过程：

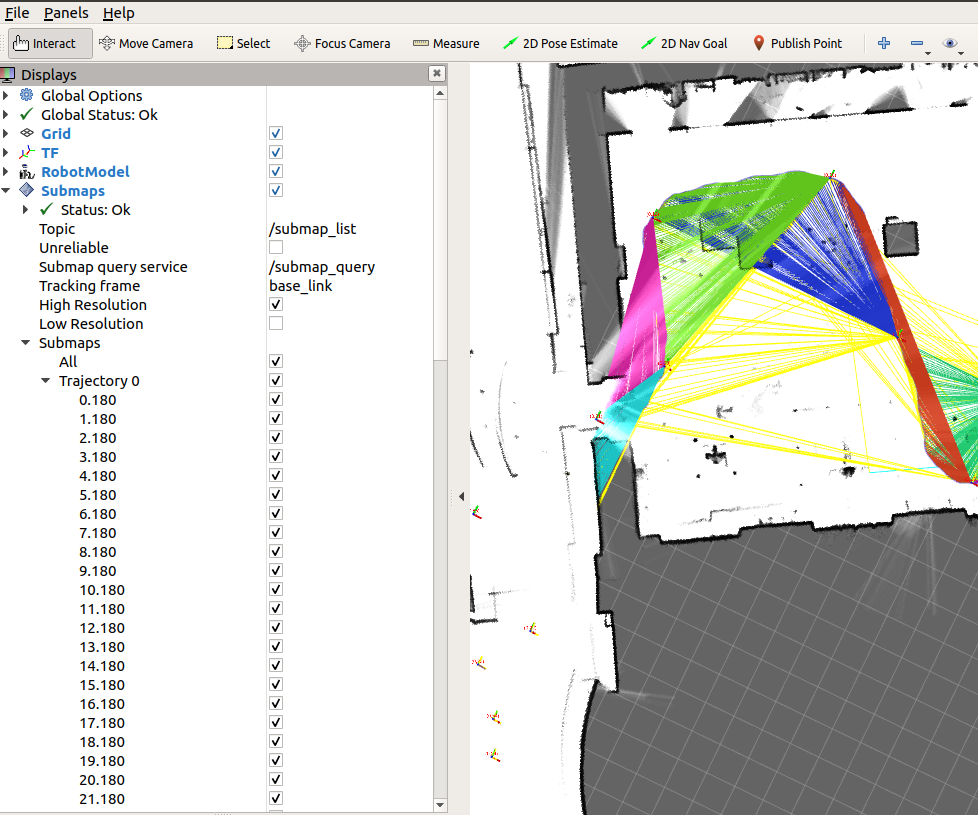
TrajectoryBuilder（用来创建trajectory）

{

创建一条trajectory过程：

1 整体概况

获得传感器数据并保存，从中抽取关键帧（可能是一些相关滤波），其中的一帧关键帧被为trajectory上的一个节点，一条这trajectory由这样一串若干个节点组成。所以TrajectoryBuilder要维护一个节点列表，同时，每一帧时的传感器数据相对于该Submap的局部坐标变换要已知



submapId

nodeId

trajectoryId

直观的说，运行官方给的包，注意左下方这一串数字。有一条trajectory id为0，然后从开始接受传感器数据起这 一列数据开始从0增加，产生一个节点这里就加一，计数满180个后就产生一个submap，然后这个数继续从0开始++构建下一个submap...

所以说一条trajectory是由一串这样的节点组成的。

2源码解读（这部分代码在cartographer里面，不是cartographer\_ros）

//InsertionResult就是用来保存插入Local Slam的一个节点的数据结构

InsertionResult:

{

有三个成员

1. NodeId node\_id;

2.std::shared\_ptr<const TrajectoryNode::Data> constant\_data;

3.std::vector<std::shared\_ptr<const Submap>> insertion\_submaps;

解释：

1 nodeId:包含了trajectoryId和node\_index的信息来指定具体的一个node

2 TrajectoryData:是存储这个节点时的状态，包括时间、传感器数据等信息。比如 //节点在世界坐标系下的位姿transform::Rigid3d global\_pose;

3 submap 一个子图，首先要有一个local\_pose。local\_pose可以看 //做是没有经过全局优化的该submap相对于世界坐标系的位姿。

//就是我们在rviz中看到的机器人在运行途中留下的一系列坐标轴

//（localpose，没有做闭环的优化）

接下来着重研究以下一个submap的构建过程

}