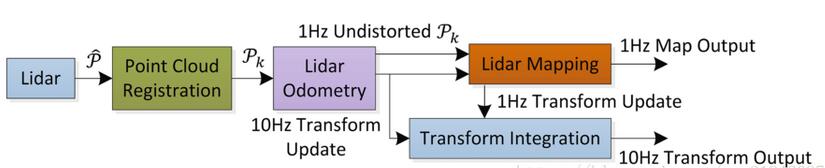
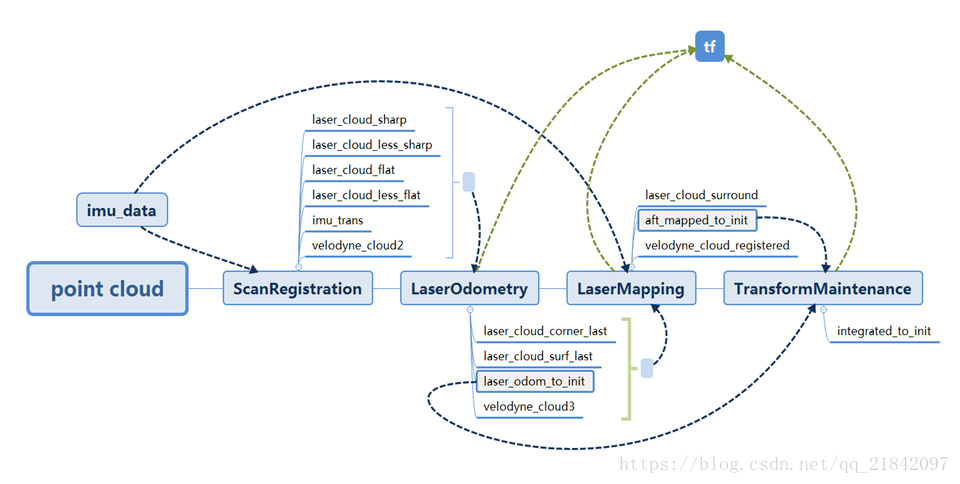
# 本周工作总结：





还是以这两个图为准

详细理解部分：

1回顾之前的工作：点云数据进来后，经过前两个节点的处理可以完成一个完整但粗糙的里程计，可以概略地估计出Lidar所在机器的相对运动。如果不受任何测量噪声的影响，这个运动估计的结果足够精确，没有任何漂移，那我们可以直接利用估计的Lidar位姿和对应时刻的量测值完成建图。但量测噪声是不可避免的，因此Lidar位姿估计偏差一定存在。

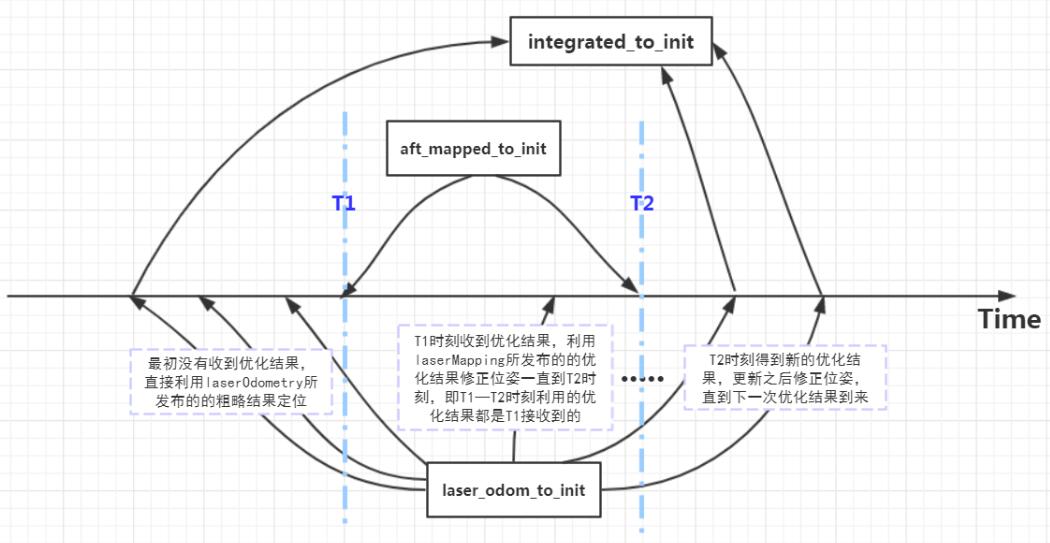
Lidar里程计的结果不准确，拼起来的点也完全不成样子，且它会不断发散，因此误差也会越来越大。我们对特征的提取仅仅只是关注了他们的曲率，且点云中的点是离散的，无法保证上一帧的点在下一帧中仍会被扫到。因此，我们需要依靠别的方式去优化Lidar里程计的位姿估计精度。在SLAM领域，一般会采用与地图匹配的方式来优化这一结果。我们始终认为后一时刻的观测较前一时刻带有更多的误差，换而言之，我们更加信任前一时刻结果。因此我们对已经构建地图的信任程度远高于临帧点云配准后的Lidar运动估计。所以我们可以利用已构建地图对位姿估计结果进行修正。因此LaserMapping这一模块的功能便是**优化Lidar的位姿，在此基础上完成低频的环境建图。**

这部分的难点是会涉及到很多的坐标系变换操作。坐标系变换的操作位置在代码中已经详细注释。

LidarMapping算法的配准跟 LidarOdometry算法的配准过程略有不同（主要体现在对应点确定方式的不同），在LidarOdometry算法中，对应点的确定是为了最快的计算速度（边角点与上一时刻的边角线匹配，平面点与上一时刻的平面匹配），而LidarMapping算法是通过对特征点周围的点云簇进行PCA主成分分析（求点云的协方差矩阵的特征值和特征向量），来找到对应边和对应面。对应点确定后的过程就和LidarOdometry算法中的匹配过程一样了。

2 TransformMaintenance线程的功能是接收laserOdometry节点和laserMapping节点发布的消息，然后对两种不同频率的信号进行处理。

这部分由一个图即可说明：



# 下周工作计划：

# 对LOAM算法进行测试，观察实验效果。