LOAM：Lidar Odometry and Mapping in real-time

一、相关背景

三维建图仍然是一种流行的技术。用激光雷达进行建图是很常见的，因为激光雷达可以提供高频率的范围测量，无论测量的距离有多远，误差都是相对稳定的。在激光雷达的唯一运动是旋转激光束的情况下，点云的配准很简单。

二、存在问题

如果激光雷达本身是移动的，因为测距数据是在不同的时间收到的，运动估计的错误会导致产生的点云的错误匹配。

三、现有解决方案

①使用独立的位置估计（如通过GPS/INS）将激光点配准到一个固定的坐标系中。
②使用里程计系统，如来自车轮编码器或视觉里程计系统来配准激光点。由于里程计集成了随时间变化的小的增量运动，它必然会发生漂移，因此人们对减少漂移给予了很大的关注（例如，使用回环）

四、作者的核心思想

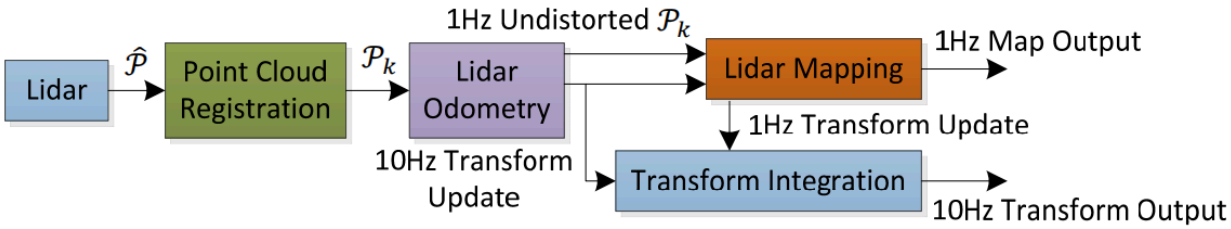


Fig. 3. Block diagram of the lidar odometry and mapping software system.

作者的方法同时实现了低漂移和低计算复杂性，而不需要高精度的测距或惯性测量。获得这种性

能水平的关键思想是将通常复杂的同步定位和建图（SLAM）问题划分为两种算法，该问题寻求同时优化大量的变量。一种算法以高频率但低精度的方式进行定位，以估计激光雷达的速度。另一个算法以低一个数量级的频率运行，用于点云的精细匹配和配准。

①在里程计算法中，特征点的对应关系是通过确保快速计算而找到的。

②在建图算法中，对应关系是通过检查局部点群的几何分布，通过相关的特征值和特征向量来确定的。

本文讨论的问题是用三维激光雷达感知的点云进行自我运动估计，并建立一个穿越环境的地图。我们假设激光雷达是预先校准的。我们还假设激光雷达的角速度和线速度在一段时间内是平滑和连续的，没有突然的变化。

五、核心步骤

符号描述

作为本文的惯例，我们使用右大写的上标来表示坐标系。我们将一个扫频定义为激光雷达完成一次扫描覆盖。我们用右边的订阅号 k ， $k \in Z^+$ 来表示扫频， $\mathcal{P}_k$ 表示在扫频 k 期间感知到的点云。 让我们定义两个坐标系如下。

1）雷达坐标系 $\{L\}$是原点在雷达几何中心的3D坐标系。x轴指向左方，y轴指向上方，z轴指向前方。点云集合 $\mathcal{P}_k$ 中的点 i 在 k 时刻的{ L_k }的坐标表示为 $\boldsymbol{X}_{(k,i)}^{L}$ 。

2）世界坐标系{ W }初始位置是第一帧雷达坐标系 $\{L$ }，也是一个3D坐标系。点云集合 $\mathcal{P}_k$ 中的点 i 在 k 时刻的{ W_k }的坐标表示为 $\boldsymbol{X}_{(k,i)}^{W}$ 。

1、 雷达里程计 LIDAR ODOMETRYLIDAR ODOMETRY

A. 特征点提取 Feature Point Extraction

首先从激光雷达云中提取特征点,P_k 。激光雷达自然会在P_k中产生分布不均的点。

为了计算雷达的运动位姿，我们需要得到的是相邻帧间的姿态变换关系.为了获取到相邻帧的姿态变换，使用全部点云处理是不可靠的，为了减少计算的时间消耗，一般需要使用特征点来代替完整的数据帧。

(1)计算曲率

选择的特征点是在尖锐的边缘和平面上的patch。设**i**是 P_k 中的一个点， i∈P_k，设**S**是激光扫描仪在同一扫描中返回的**i**的连续点的集合。由于激光扫描仪以CW（顺时针）或CCW（逆时针）的顺序产生点的返回，**S**包含其在**i**两侧的一半的点和两点之间的0.25◦间隔。

定义一个术语来评价局部表面的光滑度（也就是点在当前处的曲率）：

$$c = \frac{1}{|\mathcal{S}| \cdot ||\boldsymbol{X}^L_{(k,i)}||} || \sum_{j \in \mathcal{S}, j \neq i} (\boldsymbol{X}^L_{(k,i)} - \boldsymbol{X}^L_{(k,j)}) ||. \qquad (1)$$

这里可以这样理解光滑度：

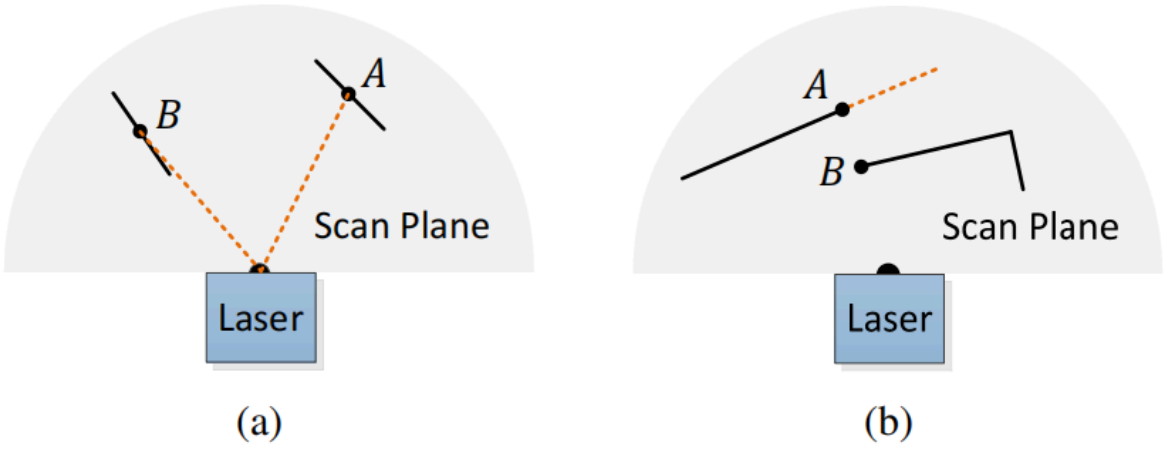
S相当于i激光直线周围的点
c计算的是S中每个点到i的平均距离
如果c小，说明平均距离短，为平面点
否则c大，说明平均距离大，为边缘点

扫描中的点根据 c值进行排序，然后选择 c值最大的特征点，即边缘点，以及 c值最小的特征点，即平面点。为了在环境中均匀分布特征点，我们将一次扫描分成四个相同的子区域。每个子区域最多可以提供2个边缘点和4个平面点。只有当一个点 i的 c值大于或小于一个阈值，并且被选中的点的数量不超过最大值时，它才能被选为一个边缘点或平面点。

(2)选择特征点

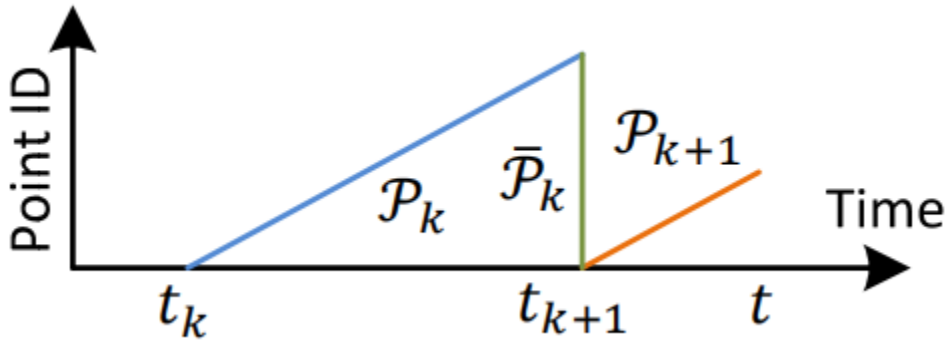
特征点的选择是从最大c值开始的边缘点，以及从最小c值开始的平面点，如果一个点被选中需满足下面的条件:

- 被选中的边缘点或平面点的数量不能超过该子区域的最大值(不能超过设定的size，每个集合平面点4个，边缘点2个,这里选的点相对来说比较少了，但有利于加速计算)
- 它的周围没有一个点已经被选中(使得点可以分布的更加均匀)
- 该点不能位于与激光束大致平行的表面上(这些点通常被认为是不可靠的)，或在被遮挡区域的边界上(如果激光雷达移动到另一个视点，被遮挡的区域会发生变化，成为可观察的区域)。



如该图(a)中的点b与激光几乎平行而不被选择，图(b)中的点a 在被遮挡区域的边界上，故不选择。

B. 寻找特征点的对应关系 Finding Feature Point Correspondence



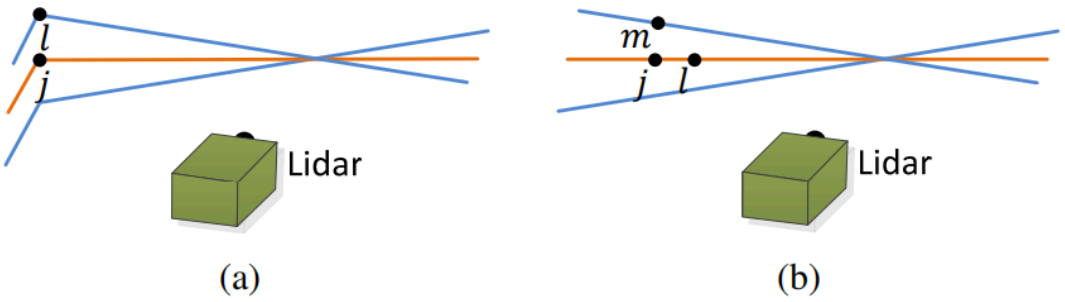
里程计算法估计激光雷达在一次扫描中的运动。在每次扫描结束时，扫描期间感知到的点云P_k。P_k被重新投影到时间戳 t_{k+1} 。如上图所示。我们把重投的点云表示为$\overline{P}$,在下一次扫描k+1期间$\overline{P}$与新收到的点云 P_{k+1}一起用于估计激光雷达的运动。

让 ε_{k+1} 和 H_{k+1} 分别为P_{k+1}边缘点和平面点的集合。我们将从 P_k 中找到边缘线作为 ε_{k+1} 中的点的对应关系，而平面patch作为 H_{k+1}中的点的对应关系。

在这一步我们想要得到的是 P_k和 P_{k+1}之间的变换关系，也就是 ε_k和ε_{k+1} ，以及 H_k和 H_{k+1}之间的关系。

请注意，在扫频k+1开始时， P_{k+1}是一个空集，在扫频过程中随着接收到更多的点而增加。激光雷达里程计在扫描过程中递归地估计6-DOF运动，并随着 P_{k+1} 的增加逐渐包括更多的点。

在每个迭代中，ε_{k+1} 和 H_{k+1} 被重新投射到使用当前估计的变换的扫频开始处。让 $\tilde{\varepsilon_{k+1}}$ 和 $\tilde{H_{k+1}}$ 为重射的点集。对于 $\tilde{\varepsilon_{k+1}}$ 和 $\tilde{H_{k+1}}$ 中的每个点，我们将在 $\overline{P_k}$ 中找到最近的邻点。这里，$\overline{P_k}$ 被存储在一个三维KD树中[24]，以便快速索引。



(a) (b)

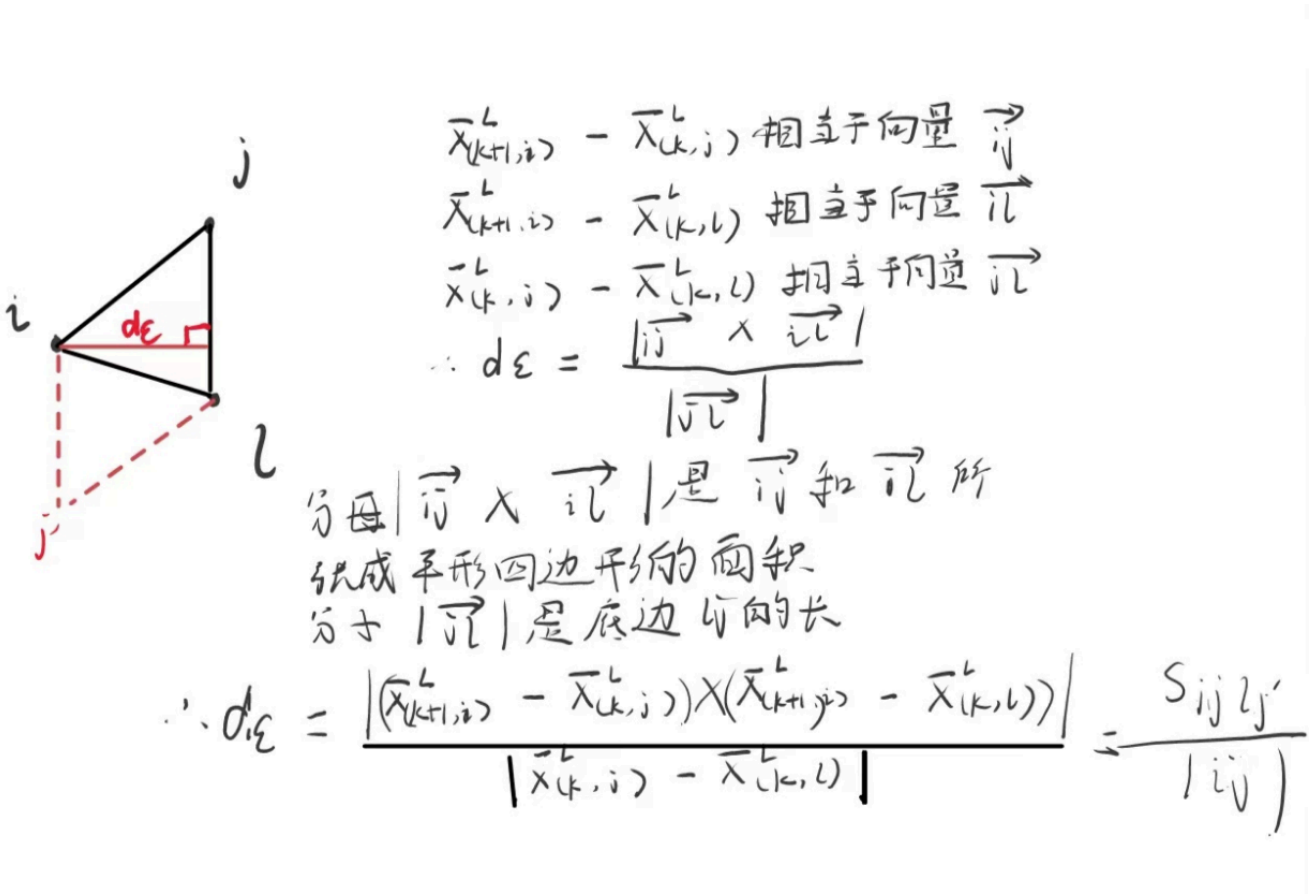
边缘点匹配

如上图(a)，**j**是上一帧中离**i**最近的边缘点，两点成线、需要再找一个边缘点**l**，但**l**不能和**j**在同一个扫描线上的点(如果是这样的话，边缘线就会被退化，在扫描平面上显示为一条直线，而边缘线上的特征点首先就不应该被提取出来。)

计算**i**到匹配直线的距离：

$$d_{\mathcal{E}} = \frac{\left| (\tilde{\boldsymbol{X}}^L_{(k+1,i)} - \bar{\boldsymbol{X}}^L_{(k,j)}) \times (\tilde{\boldsymbol{X}}^L_{(k+1,i)} - \bar{\boldsymbol{X}}^L_{(k,l)}) \right|}{\left| \bar{\boldsymbol{X}}^L_{(k,j)} - \bar{\boldsymbol{X}}^L_{(k,l)} \right|}, \quad (2)$$

该公式解释如下:



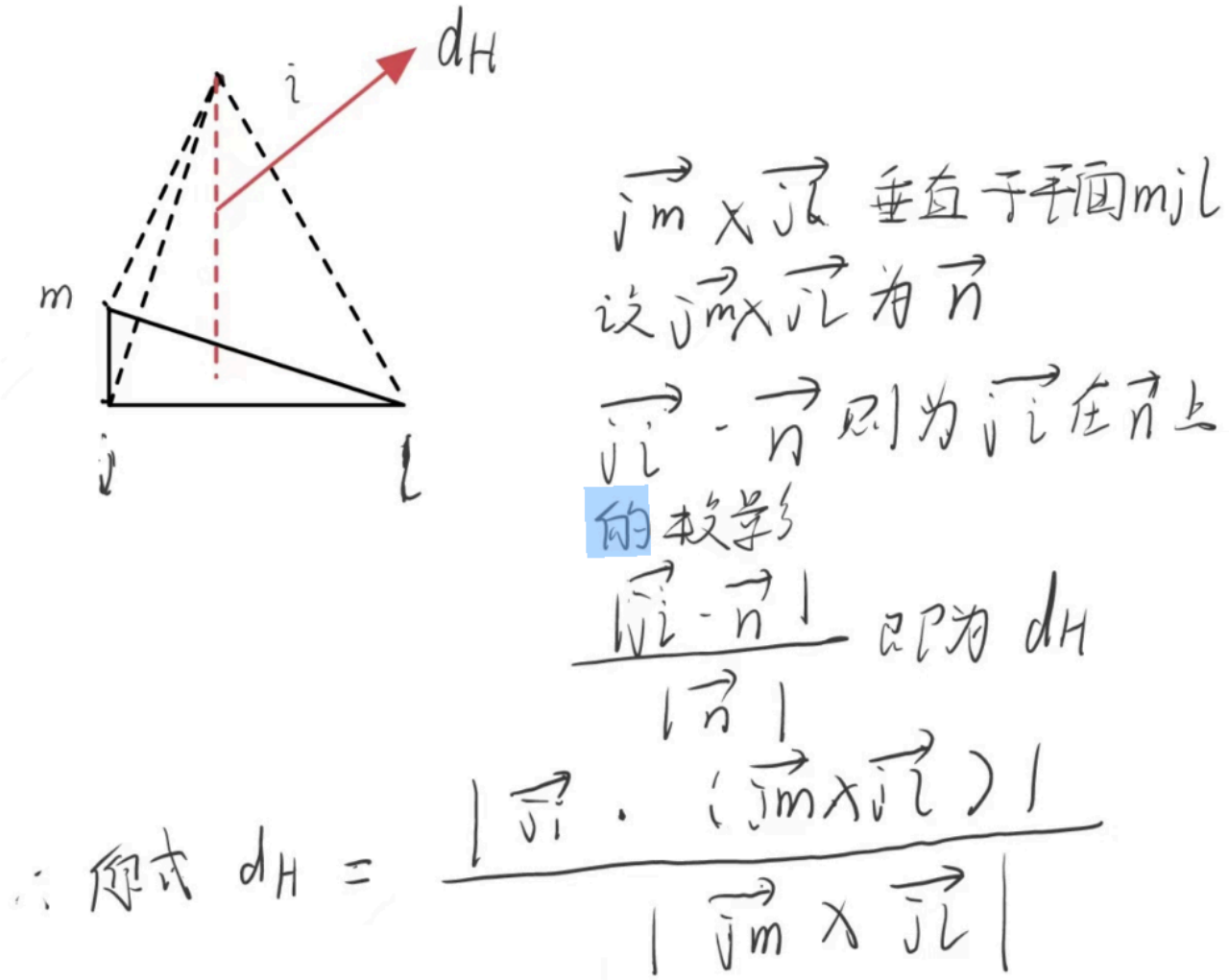
平面点匹配

如上图(b)，平面patch由三个点表示。与上一段类似，我们在 $\overline{P_k}$中找到 **i**的最近邻居**j**。然后，我们找到另外两个点，**l** 和 **m** ， 作为 **i**的最近邻居，一个在**j** 的同一扫描中，另一个在**j**的扫描的两个连续扫描中。这保证了这三个点是不相邻的。

计算**i**到匹配平面的距离：

$$d_{\mathcal{H}} = \frac{\left| (\tilde{\boldsymbol{X}}^L_{(k+1,i)} - \bar{\boldsymbol{X}}^L_{(k,j)}) \cdot ((\bar{\boldsymbol{X}}^L_{(k,j)} - \bar{\boldsymbol{X}}^L_{(k,l)}) \times (\bar{\boldsymbol{X}}^L_{(k,j)} - \bar{\boldsymbol{X}}^L_{(k,m)})) \right|}{\left| (\bar{\boldsymbol{X}}^L_{(k,j)} - \bar{\boldsymbol{X}}^L_{(k,l)}) \times (\bar{\boldsymbol{X}}^L_{(k,j)} - \bar{\boldsymbol{X}}^L_{(k,m)}) \right|}, \qquad (3)$$

该公式解释如下:



C. 运动估计 Motion Estimation

在一次扫描中，激光雷达的运动是以恒定的角度和线性速度为模型的。这使我们能够在一次扫描中对不同时间接收到的点进行线性插值变换。让 t 是当前的时间戳，并回顾 t_{k+1} 是扫频k+1的开始时间。让 $\boldsymbol{T}_{k+1}^{L}$ 是[t_{k+1} , t]之间的激光雷达姿势变换。 $\boldsymbol{T}_{k+1}^{L}$ 包含激光雷达在6-DOF中的刚性运动， $\boldsymbol{T}_{k+1}^{L} = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^{\mathrm{T}}$ ，其中 t_x 、 t_y 和 t_z 分别是沿{ L }的x轴、y轴和z轴的平移， θ_x 、 θ_y 和 θ_z 是旋转角，遵循右手规则。给定一个点 i ， $i \in \mathcal{P}_{k+1}$ ，让 t_i 为其时间戳，让 $\boldsymbol{T}_{(k+1,i)}^{L}$ 为[t_{k+1} , t_i]之间的姿势变换。 $\boldsymbol{T}_{(k+1,i)}^{L}$ 可以通过 $\boldsymbol{T}_{k+1}^{L}$ 的线性插值计算出来。

$$\boldsymbol{T}_{(k+1,i)}^{L} = \frac{t_i - t_{k+1}}{t - t_{k+1}} \boldsymbol{T}_{k+1}^{L}. \tag{4}$$

论文使用的是6-DoF的表示，也就是[x,y,z,roll,pitch,yaw] 。所以，我们需要想办法从6-DoF得到对应的姿态变换矩阵，进行迭代优化求解。设定旋转矩阵 **R**和平移矩阵 **T**,这样的话，就可以构建公式：

$$\boldsymbol{X}^{L}_{(k+1,i)} = \mathbf{R}\tilde{\boldsymbol{X}}^{L}_{(k+1,i)} + \boldsymbol{T}^{L}_{(k+1,i)}(1:3), \tag{5}$$

这样，就可以将**k+1**帧上的点投到**k**帧所在的集合中了。

接下来求解**R**和**T** 了。然而使用的是6-DoF的表示，就需要想办法进行欧拉角到旋转矩阵的变换，使用罗德里格斯公式进行变换：

$$\mathbf{R} = e^{\hat{\omega}\theta} = \mathbf{I} + \hat{\omega}\sin\theta + \hat{\omega}^2(1-\cos\theta). \tag{6}$$

在上述方程中， **θ**是旋转大小：

ω表示旋转方向的单位向量($\hat{\omega}$是**ω**的反对称矩阵)：

$$\boldsymbol{T}^{L}_{(k+1,i)} = \frac{t_i - t_{k+1}}{t - t_{k+1}}\boldsymbol{T}^{L}_{k+1}. \tag{4}$$

然后类似于损失的计算:

$$f_{\mathcal{E}}(\boldsymbol{X}^{L}_{(k+1,i)}, \boldsymbol{T}^{L}_{k+1}) = d_{\mathcal{E}},\ i \in \mathcal{E}_{k+1}. \tag{9}$$

$$f_{\mathcal{H}}(\boldsymbol{X}^{L}_{(k+1,i)}, \boldsymbol{T}^{L}_{k+1}) = d_{\mathcal{H}},\ i \in \mathcal{H}_{k+1}. \tag{10}$$

$$\boldsymbol{f}(\boldsymbol{T}^{L}_{k+1}) = \boldsymbol{d}, \tag{11}$$

然后通过推导

和高斯牛顿法不一样的是，我们加入了信赖区域， μ 为半径， D 为系数矩阵，最小二乘的函数也不相同：

$$min\frac{1}{2}||f(T_{k+1}^L)+J(T_{k+1}^L)^T\Delta T_{k+1}^L|| \quad s.t \quad ||D\Delta T_{k+1}^L<\mu||_2$$

构建拉格朗日函数， λ 是系数因子：

$$L(\Delta T_{k+1}^L,\lambda)=\frac{1}{2}||f(T_{k+1}^L)+J(T_{k+1}^L)^T\Delta T_{k+1}^L||^2+\frac{\lambda}{2}(||D\Delta T_{k+1}^L||^2-\mu)$$

这样的话，化简后求导就可以得到：

$$J(T_{k+1}^L)f(x)+J(T_{k+1}^L)J^T(T_{k+1}^L)\Delta T_{k+1}^L+\lambda D^TD\Delta T_{k+1}^L=0$$

我们化简后得到：

$$(JJ^T+\lambda D^TD)\Delta T=-Jf$$

而 $f(T_{k+1}^L)=d$

故，我们可以得到导数为：

$$\Delta T=-(JJ^T+\lambda D^TD)^-Jd$$

我们可以看到，和论文中略有不同，这个和初始雅可比矩阵 J 以及系数矩阵 D 有关，实际使用中，系数矩阵D，通常是用 J^TJ 来表示的，这样的话，我们就得到了其微分量 ΔT 。

得到

$$\boldsymbol{T}_{k+1}^L\leftarrow\boldsymbol{T}_{k+1}^L-(\mathbf{J}^T\mathbf{J}+\lambda\mathrm{diag}(\mathbf{J}^\mathrm{T}\mathbf{J}))^{-1}\mathbf{J}^\mathrm{T}\boldsymbol{d}. \qquad (12)$$

D. 雷达里程计算法 Lidar Odometry Algorithm

Algorithm 1: Lidar Odometry

input : $\bar{\mathcal{P}}_k$, $\mathcal{P}_{k+1}$, $\boldsymbol{T}_{k+1}^L$ from the last recursion
output : $\bar{\mathcal{P}}_{k+1}$, newly computed $\boldsymbol{T}_{k+1}^L$
begin
  if at the beginning of a sweep **then**
    $\boldsymbol{T}_{k+1}^L \leftarrow \boldsymbol{0}$;
  end
  Detect edge points and planar points in $\mathcal{P}_{k+1}$, put the points in $\mathcal{E}_{k+1}$ and $\mathcal{H}_{k+1}$, respectively;
  for a number of iterations **do**
    for each edge point in $\mathcal{E}_{k+1}$ **do**
      Find an edge line as the correspondence, then compute point to line distance based on (9) and stack the equation to (11);
    end
    for each planar point in $\mathcal{H}_{k+1}$ **do**
      Find a planar patch as the correspondence, then compute point to plane distance based on (10) and stack the equation to (11);
    end
    Compute a bisquare weight for each row of (11);
    Update $\boldsymbol{T}_{k+1}^L$ for a nonlinear iteration based on (12);
    if the nonlinear optimization converges **then**
      Break;
    end
  end
  if at the end of a sweep **then**
    Reproject each point in $\mathcal{P}_{k+1}$ to t_{k+2} and form $\bar{\mathcal{P}}_{k+1}$;
    Return $\boldsymbol{T}_{k+1}^L$ and $\bar{\mathcal{P}}_{k+1}$;
  end
  else
    Return $\boldsymbol{T}_{k+1}^L$;
  end
end

激光雷达的里程计算法如算法1所示。该算法将上一次扫描的点云 $\overline{P}$ 、当前扫描的增长点云 P_{k+1}和上一次递归的姿势变换 T_{k+1}^{L}作为输入。

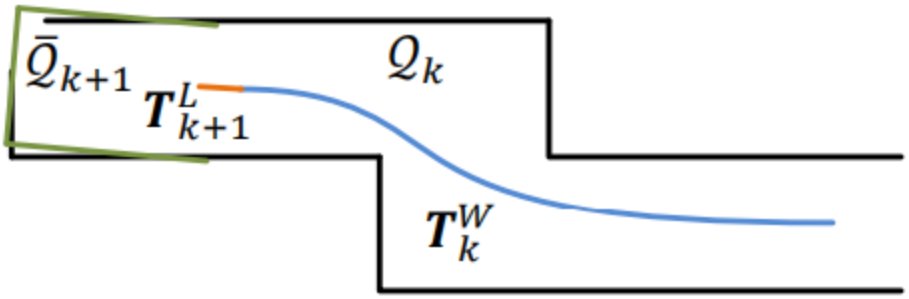
如果开始了新的扫描， T_{k+1}^{L}被设置为零（第4-6行）。然后，算法从 P_{k+1}中提取特征点，在第7行构建 ε_{k+1} 和 H_{k+1}。对于每个特征点，我们在 $\overline{P}$ 中找到其对应关系（第9-19行）。

运动估计适应于稳健的拟合。在第15行，该算法为每个特征点分配了一个二平方权重。与其对应关系有较大距离的特征点被分配较小的权重，而距离大于阈值的特征点被认为是离群值，被分配为零权重。

然后，在第16行，姿势变换被更新一次。如果发现收敛，或达到最大迭代数，非线性优化就会终止。如果算法达到了扫频的终点， P_{k+1}被重新投影到时间戳 t_{k+2} ，使用扫频期间的估计运动。否则，只返回变换 T_{k+1}^{L}用于下一轮递归。

2、雷达建图

当我们获取了若干相邻帧的姿态变换信息后，我们需要做的就是将其和全局地图进行匹配，并将其加入到全局地图之中。



这里设定：

- 第 **k+1**帧之前的扫描点云在全局坐标系下的投影为 Q_k
- 第 **k**次扫描的末位，也就是 **k+1**帧的起始时的姿态变换信息 T_k^W 。
- 利用Odometry的输出 T_{k+1}^{L}，将 T_k^W 从 **k+1**时刻的起始推演到**k+2** 时刻的起始，得到姿态矩阵T_{k+1}^{W}
  。
- 通过 T_{k+1}^{W}，将之前第**k+1** 帧的点云投影到全局坐标系下，记为 $\hat{Q_{k+1}}$。

这样，其实就很明晰了，我们需要做的就是优化求解 T_{k+1}^{W}

这里的已知信息为：{ Q_k,$\hat{Q_{k+1}}$}，不精准的 T_{k+1}^W 。想要优化得到精准的 。T_{k+1}^W

如果使用全部的地图数据，在计算效率上会大打折扣，所以，这里使用的是一个边长为10m的立方体，用以代替全局地图，优化得到最终的姿态变换矩阵 T_{k+1}^W。

选取特征点的方法是一样的，不过具体实现的方式不尽相同。将 Q_k和 $\hat{Q_{k+1}}$中相交的存入到KD-tree中，这里的相交部分，也就是判定是否处于这个cube中。相交的部分属于了两个map之间的重合部分，可以用来作为点云匹配的依据。
在这里，首先选取相邻点集合 **S**，针对平面点和边缘点又有两种处理方法：

- 平面点： **S**只保留平面特征点；
- 边缘点： **S**只保留边缘特征点。

计算**S** 的协方差矩阵，记为 **M**，**M** 的特征值记为 **V**，特征向量记为 **E**。

- 如果**S** 分布在一条线段上，那么 **V**中一个特征值就会明显比其他两个大，**E**中与较大特征值相对应的特征向量代表边缘线的方向。（一大两小，大方向）
- 如果 **S**分布在一块平面上，那么 **V**中一个特征值就会明显比其他两个小，**E** 中与较小特征值相对应的特征向量代表平面片的方向。（一小两大，小方向）

该部分解释:

当数据分布在一条线段上时，方差主要集中在一个方向上，也就是说数据在这个方向上的变化较大。特征值代表了数据变化的大小，而特征向量表示了在该变化较大的方向上保持不变的向量。因此，矩阵S具有一个较大的特征值和与其相对应的特征向量，表示数据在边缘线的方向上变化最明显。而其他两个特征值较小且对应的特征向量表示与边缘线方向垂直的方向，即数据变化较小的方向。

另一方面，当数据分布在一块平面上时，方差分布在两个相互垂直的方向上。其中两个方向上的变化较大，而剩下的一个方向上的变化相对较小。特征值较小的两个表示数据在变化较小的方向上的方差，而特征值较大的一个表示数据在变化较大的方向上的方差。对应的特征向量表示了这些方向上保持不变的向量。因此，与较小特征值相对应的特征向量代表了数据平面内围绕着较小方差方向变化较小的方向，而与较大特征值相对应的特征向量则表示与平面片所在方向垂直的方向。

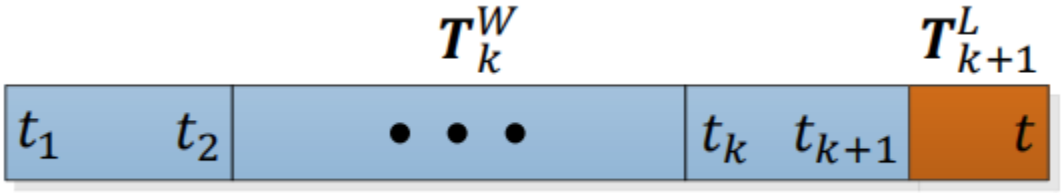
边缘线或平面块的位置通过穿过的几何中心来确定。

特征向量的长度反应了点的分布，也就是说我们根据特征值和特征向量就能计算出直线的方向。平面也是同理，我们可以根据两个较长的特征向量计算平面的法向量，三个向量相交于几何中心，这样平面就确定了。通过这种方法就可以快速的确定对应的边缘线和平面了。

这里需要注意的是除了由于实时性的缘故，找对应的特征点更换了方法，Lidar Mapping其他的

算法步骤和Lidar Odometry 的基本一致。

这里需要注意的是，Lidar Odometry中使用过运动补偿了，这里的点云就都被设置为对应帧的时间戳，就不用再考虑运动补偿的事情了。后续可以通过VoxelFilter来进行降噪，减少点云的数量。



姿势变换的整合在上图中得到说明。蓝色区域代表激光雷达建图线程的姿态输出，T_k^W。，每次扫描产生一次。橙色的区域代表来自激光雷达里程计线程的变换输出，T_{k+1}^W。，频率为10Hz。激光雷达相对于地图的姿态是这两个变换的组合，频率与激光雷达里程计相同。

六、代码阅读

七、实验验证(配置环境遇到问题,还未完成)