

LOAM: Lidar Odometry and Mapping in Real-time

0、摘要

我们提出了一种使用在 6 自由度中移动的 2 轴激光雷达的距离测量进行测距和建图的实时方法。这个问题很难解决，因为距离测量值是在不同时间接收到的，运动估计中的错误会导致生成的点云配准错误。迄今为止，连贯的 3D 地图可以通过离线批处理方法构建，通常使用闭环来校正随时间的漂移。我们的方法无需高精度测距或惯性测量即可实现低漂移和低计算复杂性。获得这种性能水平的关键思想是同时定位和建图的复杂问题的划分，它试图通过两种算法同时优化大量变量。一种算法以高频率但低保真度执行里程计以估计激光雷达的速度。另一种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。这两种算法的组合允许该方法实时建图。该方法已经通过大量实验以及 KITTI 里程计基准进行了评估。结果表明，该方法可以达到最先进的离线批处理方法水平的准确性。

1、 介绍

3D 建图仍然是一种流行的技术 [1]-[3]。使用激光雷达进行测绘很常见，因为激光雷达可以提供高频范围测量，其中无论测量距离如何，误差都相对恒定。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。然而，如果激光雷达本身在许多感兴趣的应用中都在移动，则精确建图需要了解激光雷达在连续激光测距

期间的姿态。解决该问题的一种常见方法是使用独立位置估计（例如通过 GPS/INS）将激光点注册到固定坐标系中。另一组方法使用测距测量，例如来自车轮编码器或视觉测距系统 [4]、[5] 的测距测量来记录激光点。由于里程计随着时间的推移整合了小的增量运动，它必然会漂移，并且很多注意力都集中在减少漂移上（例如使用闭环）。

在这里，我们考虑使用在 6-DOF 中移动的 2 轴激光雷达使用低漂移里程计创建地图的情况。使用激光雷达的一个关键优势是它对场景中的环境照明和光学纹理不敏感。激光雷达的最新发展减小了它们的尺寸和重量。激光雷达可以由人持有 [6]，甚至可以连接到微型飞行器 [7]。由于我们的方法旨在解决与最小化里程计估计中的漂移相关的问题，因此它目前没有闭环。

该方法实现了低漂移和低计算复杂度，无需高精度测距或惯性测量。获得这种性能水平的关键思想是同时定位和建图（SLAM）[8] 的典型复杂问题的划分，它试图通过两种算法同时优化大量变量。一种算法以高频率但低保真度执行里程计以估计激光雷达的速度。另一种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。尽管没有必要，但如果 IMU 可用，则可以提供先验运动以帮助解释高频运动。具体来说，这两种算法都提取位于锐边和平面上的特征点，并将特征点分别与边缘线段和平面块匹配。在里程计算法中，通过确保快速计算找到特征点的对应关系。在建图算法中，

通过关联的特征值和特征向量检查局部点簇的几何分布来确定对应关系。

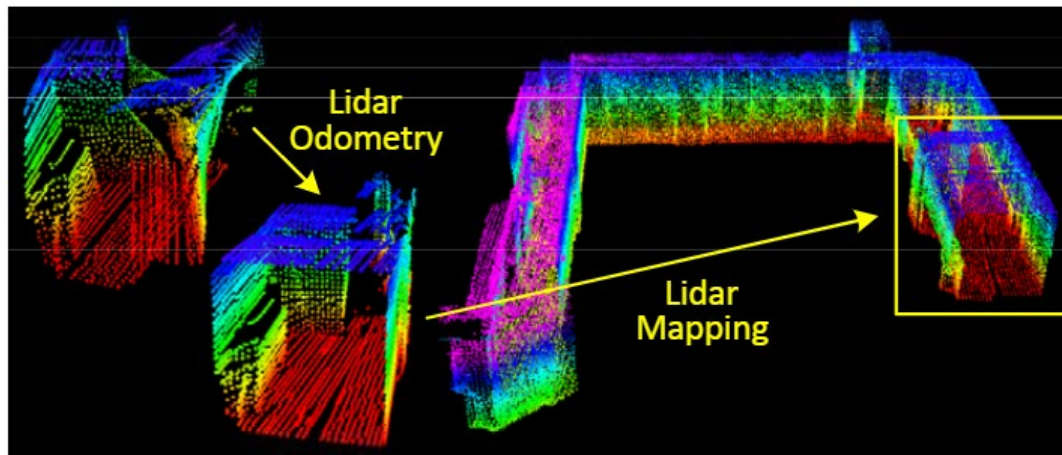


图 1：该方法旨在使用移动的 2 轴激光雷达进行运动估计和建图。由于激光点是在不同时间接收到的，因此由于激光雷达的运动（如左激光雷达点云所示），点云中会出现失真。我们提出的方法通过并行运行的两种算法来分解问题。里程计算法估计激光雷达的速度并校正点云中的失真，然后建图算法匹配并配准点云以创建地图。两种算法的结合确保了实时解决问题的可行性。

通过分解原始问题，首先解决了一个更简单的问题，即在线运动估计。之后，建图作为批量优化（类似于迭代最近点 (ICP) 方法 [9]) 进行，以生成高精度运动估计和地图。并行算法结构保证了实时求解问题的可行性。此外，由于运动估计以更高的频率进行，因此建图有足够的时间来加强准确性。当以较低频率运行时，建图算法能够合并大量特征点并使用足够多的迭代来收敛。

2、相关工作

激光雷达已成为机器人导航中有用的距离传感器[10]。对于定位和建图，大多数应用程序使用 2D 激光雷达[11]。当激光雷达扫描速

率与其外部运动相比较时，扫描中的运动失真可能经常被忽视。在这种情况下，可以使用标准 ICP 方法[12]来匹配不同扫描之间的激光返回。此外，提出了一种两步方法来消除失真[13]：基于 ICP 的速度估计步骤之后是失真补偿步骤，使用计算的速度。类似的技术也用于补偿单轴 3D 激光雷达引入的失真[14]。然而，如果扫描运动相对较慢，则运动失真可能很严重。当使用 2 轴激光雷达时尤其如此，因为一个轴通常比另一个轴慢得多。通常，使用其他传感器来提供速度测量，从而消除失真。例如，激光雷达云可以通过与 IMU 集成的视觉里程计的状态估计来注册[15]。当多个传感器（如 GPS/INS 和车轮编码器）同时可用时，通常通过扩展卡尔曼滤波器[16]或粒子滤波器[1]来解决问题。这些方法可以实时创建地图，以帮助机器人导航中的路径规划和避免碰撞。

如果在没有其他传感器帮助的情况下使用 2 轴激光雷达，则运动估计和失真校正成为一个问题。Barfoot 等人使用的一种方法是从激光强度返回创建视觉图像，并在图像之间匹配视觉上不同的特征[17]，以恢复地面车辆的运动[18]–[21]。[18]、[19]中将车辆运动建模为恒定速度，[20]、[21]中采用高斯过程。我们的方法使用与里程计算法中[18]、[19]相似的线性运动模型，但具有不同类型的特征。方法[18]–[21]涉及强度图像的视觉特征，需要密集点云。我们的方法提取并匹配笛卡尔空间中的几何特征，对云密度的要求较低。

最接近我们的方法是 Bosse 和 Zlot[3]，[6]，[22]。他们使用 2 轴

激光雷达获取点云，通过匹配局部点簇的几何结构进行配准[22]。此外，他们使用多个 2 轴激光雷达来绘制地下矿图[3]。该方法包含一个 IMU，并使用循环闭包来创建大地图。由同一作者提出，Zebedee 是一种由 2D 激光雷达和 IMU 组成的测绘设备，IMU 通过弹簧连接到手杆[6]。通过用手点设备来进行标测。通过批处理优化方法恢复轨迹，该方法处理分段数据集，并在分段之间添加边界约束。在该方法中，IMU 的测量值用于记录激光点，优化用于校正 IMU 偏差。本质上，Bosse 和 Zlot 的方法需要批量处理来开发精确的地图，因此不适用于需要实时地图的应用。相比之下，所提出的方法实时生成的地图在质量上与 Bosse 和 Zlot 的地图相似。区别在于，我们的方法可以为自动驾驶车辆的引导提供运动估计。此外，该方法利用了激光雷达扫描模式和点云分布。实现了特征匹配，分别保证了里程计和建图算法的计算速度和精度。

3、 符号和任务描述

本文解决的问题是利用 3D 激光雷达感知的点云进行自我运动估计，并为穿越的环境构建地图。我们假设激光雷达是预校准的。我们还假设激光雷达的角速度和线速度随时间平滑连续，没有突变。第二个假设将在第 VII-B 节中通过使用 IMU 予以释放。

作为本文中的惯例，我们使用右大写字母来表示坐标系。我们将扫描 (sweep) 定义为激光雷达完成一次扫描覆盖。我们使用右订阅 k , $k \in \mathbb{Z}^+$ 表示扫描， \mathcal{P}_k 表示在扫描 k 期间感知的点云。

· 激光雷达坐标系 $\{L\}$ 是一个三维坐标系，其原点位于激光雷达的几何中心。x 轴指向左侧，y 轴指向上方，z 轴指向前方。 $\{L_k\}$ 中的点 i ， $i \in \mathcal{P}_k$ 的坐标表示为 $X_{(k,i)}^L$ 。

· 世界坐标系 $\{W\}$ 是在初始位置与 $\{L\}$ 重合的 3D 坐标系。 $\{W_k\}$ 中的点 i ， $i \in \mathcal{P}_k$ 的坐标为 $X_{(k,i)}^W$ 。

通过假设和标记，我们的激光雷达里程计和建图问题可以定义为：

问题：给定一系列激光雷达云 \mathcal{P}_k ， $k \in \mathbb{Z}^+$ ，计算激光雷达在每次扫描 k 期间的自我运动，并用 \mathcal{P}_k 构建穿越环境的地图。

4、 系统概述

A、 激光雷达硬件

本文的研究在基于 Hokuyo UTM-30LX 激光扫描仪的定制 3D 激光雷达上进行了验证，但不限于此。通过本文，我们将使用从激光雷达收集的数据来说明该方法。激光扫描仪的（水平）视野为 180° 具有 0.25° 分辨率和 40 行（次）/秒扫描速率。激光扫描仪连接到电机，电机被控制以 $180^\circ/\text{s}$ 角速度旋转介于 -90° 之间和 90° （垂向）。其中激光扫描器的水平方向为零，对于这个特定的单位，扫描是从 -90° 至 90° 或相反方向（持续 1s）。这里，请注意，对于连续旋转的激光雷达，扫描只是一个半球旋转。机载编码器以 0.25 的分辨率测量电机旋转角度，利用该方法，激光点被投影到激光雷达坐标系 $\{L\}$

中。

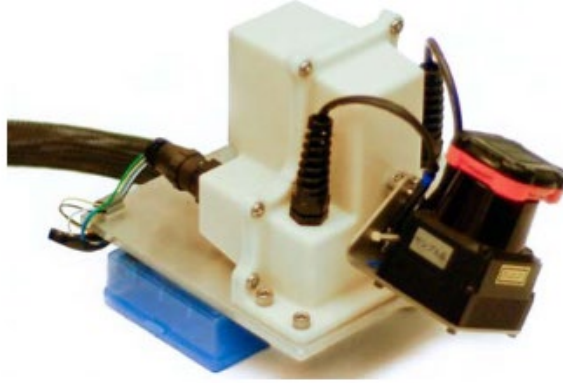


图 2：本研究使用的 3D 激光雷达由一台 Hokuyo 激光扫描仪和一台编码器组成，该扫描仪由电机驱动，用于旋转运动，编码器测量旋转角度。激光扫描仪的视野为 180° 分辨率为 0.25° 。扫描速率为 40 行/秒。控制电机从 -90° 至 90° 其中激光扫描器的水平方向为零。

B、 软件系统概述

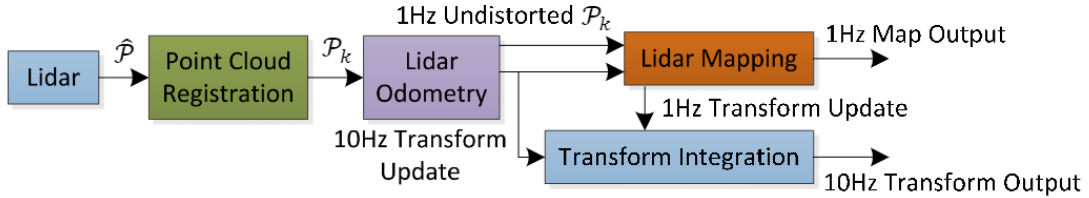


Fig. 3. Block diagram of the lidar odometry and mapping software system.

图 3 显示了软件系统的示意图。设 \hat{P} 为激光扫描中接收的点。在每次扫描 (sweep) 过程中，在 $\{L\}$ 中登记了 \hat{P} 。扫描 k 期间的组合点云形成 \mathcal{P}_k 。然后，用两种算法处理 \mathcal{P}_k 。激光雷达里程计获取点云并计算激光雷达在两次连续扫描之间的运动。估计的运动用于校正 \mathcal{P}_k 中的失真。该算法以 10Hz 左右的频率运行。输出通过激光雷达建图进一步处理，该建图以 1Hz 的频率将未失真的点云匹配并

登记到地图上。最后，将两种算法发布的姿态变换进行集成，以生成大约 10Hz 的变换输出，这与激光雷达相对于地图的姿态有关。第五节和第六节详细介绍了软件图中的框图。

5、 激光里程计

A、 特征点提取

我们首先从激光雷达云 \mathcal{P}_k 中提取特征点。图 2 所示的激光雷达在 \mathcal{P}_k 中生成不均匀分布的点。激光扫描仪的分辨率为 0.25° 。这些点位于扫描平面上。然而，当激光扫描器以 $180^\circ/\text{s}$ 并以 40Hz 产生扫描，垂直于扫描平面方向的分辨率为 $180^\circ/40 = 4.5^\circ$ 。考虑到这一事实，仅使用来自单个扫描的信息从 \mathcal{P}_k 提取特征点，具有共面几何关系。

我们选择锐边和平面面片上的特征点。设 i 是 \mathcal{P}_k 中的一个点， $i \in \mathcal{P}_k$ ，设 S 是激光扫描器在同一扫描中返回的 i 的连续点集。由于激光扫描仪以 CW 或 CCW 顺序生成点返回， s 包含 i 每侧的一半点，两点之间的间隔为 0.25° ，定义一个术语来评估局部表面的平滑度，

$$c = \frac{1}{|S| \cdot \|X_{(k,i)}^L\|} \left\| \sum_{j \in S, j \neq i} (X_{(k,i)}^L - X_{(k,j)}^L) \right\|. \quad (1)$$

基于 c 值对扫描中的点进行排序，然后选择具有最大 c 值的特征点，即边缘点，以及最小 c 值的平面点。为了在环境中均匀分布特

征点，我们将扫描分为四个相同的子区域。每个子区域最多可提供 2 个边缘点和 4 个平面点。只有当点 i 的 c 值大于或小于阈值并且所选点的数量不超过最大值时，点 i 才能被选为边或平面点。

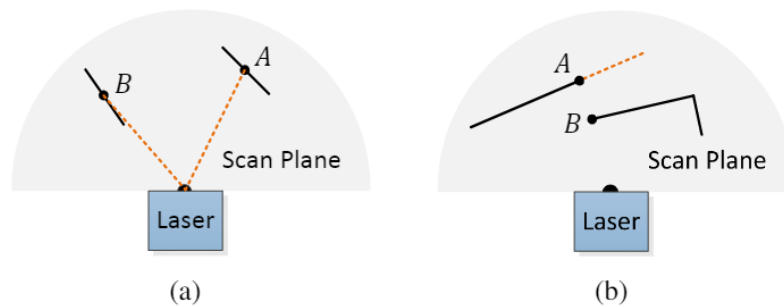


图 4: (a) 实线线段表示局部曲面面片。点 A 位于与激光束成一定角度的曲面片上（橙色虚线段）。点 B 位于大致平行于激光束的表面贴片上。我们将 B 视为不可靠的激光回波，而不选择它作为特征点。(b) 实线段是激光可观察到的物体。点 A 位于被遮挡区域（橙色虚线段）的边界上，可以检测为边缘点。然而，如果从不同的角度观察，被遮挡区域可能会发生变化并变得可观察到。我们不将 A 视为显著边缘点或将其选为特征点。

在选择特征点时，我们希望避免选择其包围点的点，或与激光束大致平行的局部平面上的点（图 4 (a) 中的点 B）。这些点通常被认为是不可靠的。此外，我们希望避免遮挡区域边界上的点[23]。示例如图 4 (b) 所示。点 A 是激光雷达云中的边缘点，因为它的连接表面（虚线段）被另一个物体阻挡。然而，如果激光雷达移动到另一个角度，被遮挡的区域可能会发生变化并变得可观察。为了避免选择上述点，我们再次找到点集 S 。只有当 S 没有形成大致平行于激光束的表面贴片，并且 S 中没有任何点通过激光束方向的间隙与 i 断开，同时距离激光雷达更近时，才能选择点 i （例如图 4 (B) 中的点 B）。

总之，如果选择了一个点，则将特征点选为从最大 c 值开始的边缘点和从最小 c 值开始选择的平面点。

- 选定角点或平面点的数量不能超过子区域的最大值；
 - 特征点周围点不能再次被选择，本来每个子区域的特征点数量就比较少了，进一步防止特征点集中分布；
 - 特征点非平行激光束的平面点，非被遮挡的平面点，非离群点。
- 这些都是不可靠的点云数据，不能在连续两帧点云数据中得到，不能作为特征点；

图 5 显示了从走廊场景中提取的特征点的示例。边缘点和平面点分别以黄色和红色标记。

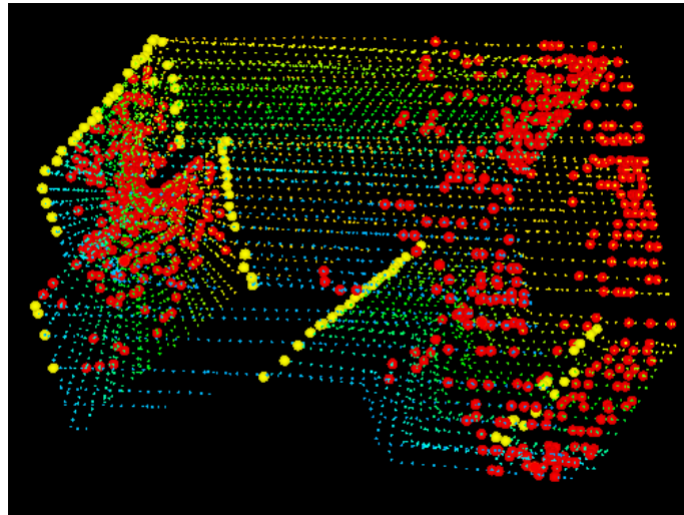


图 5：从走廊中拍摄的激光雷达云中提取的边缘点（黄色）和平面点（红色）的示例。同时，激光雷达以 0.5m/s 的速度向图左侧的墙壁移动，这导致墙壁上的运动变形。

B、 查找对应特征点

里程计算法估计激光雷达在扫描范围内的运动。设 t_k 为扫描 k 的

开始时间，在每次扫描结束时，扫描过程中感知到的点云， \mathcal{P}_k ，重新投影到时间戳 t_{k+1} ，如图 6 所示。我们将重新投影的点云表示为 $\bar{\mathcal{P}}_k$ 。在下一次扫描 $k+1$ 期间，将 $\bar{\mathcal{P}}_k$ 与新接收的点云 \mathcal{P}_{k+1} 一起用于估计激光雷达的运动。

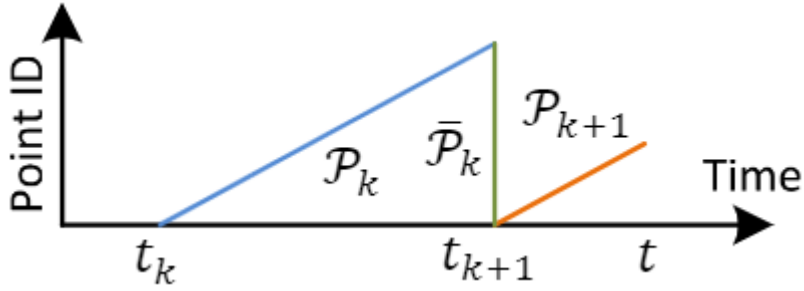


图 6：将点云重新投影到扫描的末端。蓝色线段表示在扫描 k ， \mathcal{P}_k 期间感知到的点云。在扫描 k 结束时， \mathcal{P}_k 被重新投影到时间戳 t_{k+1} 以获得 $\bar{\mathcal{P}}_k$ （绿色线段）。然后，在扫描 $k+1$ 期间，将 $\bar{\mathcal{P}}_k$ 和新感知的点云 \mathcal{P}_{k+1} （橙色线段）一起用于估计激光雷达运动。

让我们假设 $\bar{\mathcal{P}}_k$ 和 \mathcal{P}_{k+1} 目前都可用，并从寻找两个激光雷达云之间的对应关系开始。使用 \mathcal{P}_{k+1} ，我们使用上一节讨论的方法从激光雷达点云中找到边缘点和平面点。设 \mathcal{E}_{k+1} 和 \mathcal{H}_{k+1} 分别是边缘点和平面点的集合。我们将发现来自 $\bar{\mathcal{P}}_k$ 的边缘线作为 \mathcal{E}_{k+1} 中的点的对应关系，而平面面片作为 \mathcal{H}_{k+1} 中点的对应。

注意，在扫描 $k+1$ 开始时， \mathcal{P}_{k+1} 是一个空集，在扫描过程中随着接收到更多点而增加。激光雷达里程计递归地估计扫描期间的 6DOF 运动，并随着 \mathcal{P}_{k+1} 的增加逐渐包括更多的点。在每次迭代中，使用当前估计的变换将 \mathcal{E}_{k+1} 和 \mathcal{H}_{k+1} 重新投影到扫描的开始。设 |

$\bar{\mathcal{E}}_{k+1}$ 和 $\bar{\mathcal{H}}_{k+1}$ 为 **重投影点集**。对于 $\bar{\mathcal{E}}_{k+1}$ 和 $\bar{\mathcal{H}}_{k+1}$ 中的每个点，我们将找到 $\bar{\mathcal{P}}_k$ 中最近的相邻点。这里， $\bar{\mathcal{P}}_k$ 存储在 **3D KD tree**[24]中，用于快速索引。

图 7 (a) 显示了寻找边缘线作为边缘点对应关系的过程。设 i 是 $\bar{\mathcal{E}}_{k+1}$ 中的一个点， $i \in \bar{\mathcal{E}}_{k+1}$ 。边缘线由两点表示。设 j 是 $\bar{\mathcal{P}}_k$ 中 i 的最近邻居， $j \in \bar{\mathcal{P}}_k$ ，并且设 **l 是 j 的两次连续扫描中 i 的最接近邻居**。

(j, l) 形成 i 的对应关系。然后，为了验证 j 和 l 都是边缘点，我们基于 (1) 检查局部表面的平滑度。这里，我们特别要求 **j 和 l 来自不同的扫描**，考虑到单个扫描不能包含来自同一边缘线的多个点。只有一个例外，即边缘线位于扫描平面。

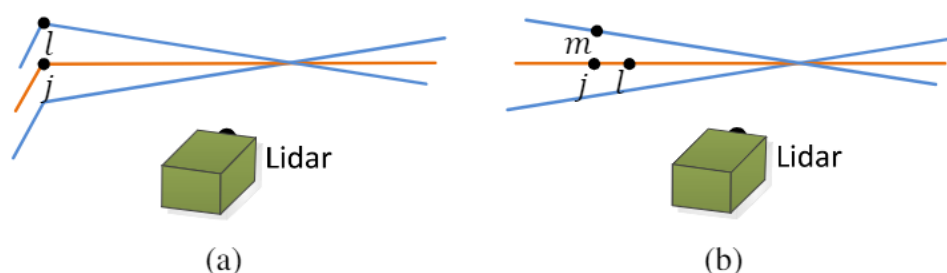


图 7: 找到一条边缘线作为 $\bar{\mathcal{E}}_{k+1}$ (a) 中边缘点的对应关系，并找到一个平面补丁作为

$\bar{\mathcal{H}}_{k+1}$ (b) 中平面点的对应关系。在 (a) 和 (b) 中， j 是距离特征点最近的点，在 $\bar{\mathcal{P}}_k$ 中找到。橙色线表示 j 的相同扫描，蓝色线表示两次连续扫描。为了找到 (a) 中的边缘线对应关系，我们在蓝色线上找到另一个点 l ，对应关系表示为 (j, l) 。为了找到 (b) 中的平面补丁对应关系，我们分别在橙色和蓝色线上找到另外两个点 l 和 m 。对应关系是 (j, l, m) 。

图 7 (b) 显示了将平面面片作为平面点的对应关系的过程。设 i 是一个点，在 $\bar{\mathcal{H}}_{k+1}$ ， $i \in \bar{\mathcal{H}}_{k+1}$ 。平面面片由三个点表示。与上一段

类似，我们在 $\bar{\mathcal{P}}_k$ 中找到 i 的最近邻居，记为 j 。然后，我们找到另两个点 l 和 m 作为 i 的最近邻，一个在 j 的同一扫描中，另一个在与 j 的扫描相邻的两个连续扫描中。这保证了三个点是非共线的。为了验证 j 、 l 和 m 都是平面点，我们再次基于（1）检查局部曲面的平滑度。

通过找到特征点的对应关系，现在我们导出表达式来计算从特征点到其对应关系的距离。我们将在下一节中通过最小化特征点的总距离来恢复激光雷达运动。我们从边缘点开始。对于点 $i \in \bar{\mathcal{E}}_{k+1}$ ，如果 (j, l) 是对应的边缘线， $j, l \in \bar{\mathcal{P}}_k$ ，则点到线的距离可以计算为：

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

$\bar{\mathbf{X}}_{(k+1,i)}^L$ 、 $\bar{\mathbf{X}}_{(k,j)}^L$ 、 $\bar{\mathbf{X}}_{(k,l)}^L$ 别是 $\{\mathbf{L}\}$ 中点 i 、 j 和 l 的坐标。然后，对于点 $i \in \bar{\mathcal{H}}_{k+1}$ ，如果 (j, l, m) 是对应的平面面片， $j, l, m \in \bar{\mathcal{P}}_k$ ，则点到平面的距离为：

$$d_{\mathcal{H}} = \frac{\left| \begin{array}{c} (\tilde{\mathbf{X}}_{(k+1,i)}^L - \bar{\mathbf{X}}_{(k,j)}^L) \\ ((\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L) \times (\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,m)}^L)) \end{array} \right|}{\left| (\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L) \times (\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,m)}^L) \right|}, \quad (3)$$

$\bar{\mathbf{X}}_{(k,m)}^L$ 是点 m 在 $\{\mathbf{L}\}$ 中的坐标。

C、 运动估计

激光雷达运动在扫描过程中以恒定的角速度和线速度建模。这允许我们在扫描中对在不同时间接收的点线性插值姿势变换。设 t 为当前时间戳，并记住 t_{k+1} 是 $k+1$ 次扫描 (sweep) 的开始时间。设 T_{k+1}^L 是 $[t_{k+1}, t]$ 之间的激光雷达姿态变换。 T_{k+1}^L 包含 6 自由度激光雷达的刚性运动， $T_{k+1}^L = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^T$ ，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 是旋转角度，遵循右手法则。给定点 i , $i \in \mathcal{P}_{k+1}$ ，设 t_i 为其时间戳，并设 $T_{(k+1, i)}^L$ 为 $[t_{k+1}, t_i]$ 之间的位姿变换。 $T_{(k+1, i)}^L$ 可以通过 T_{k+1}^L 的线性插值来计算：

$$T_{(k+1, i)}^L = \frac{t_i - t_{k+1}}{t - t_{k+1}} T_{k+1}^L. \quad (4)$$

回想一下， \mathcal{E}_{k+1} 和 \mathcal{H}_{k+1} 是从 \mathcal{P}_{k+1} 中提取的边缘点和平面点的集合，而 $\bar{\mathcal{E}}_{k+1}$ 与 $\bar{\mathcal{H}}_{k+1}$ 则是重新投影到扫描开始 t_{k+1} 的点集合。为了解决激光雷达的运动，我们需要在 \mathcal{E}_{k+1} 和 $\bar{\mathcal{E}}_{k+1}$ ，或 \mathcal{H}_{k+1} 和 $\bar{\mathcal{H}}_{k+1}$ 之间建立几何关系。使用 (4) 中的变换，我们可以得出：

$$X_{(k+1, i)}^L = R \tilde{X}_{(k+1, i)}^L + T_{(k+1, i)}^L(1:3), \quad (5)$$

其中， $X_{(k+1, i)}^L$ 是 \mathcal{E}_{k+1} 或 \mathcal{H}_{k+1} 中的点 i 的坐标， $\tilde{X}_{(k+1, i)}^L$ 为对应的 $\bar{\mathcal{E}}_{k+1}$ 和 $\bar{\mathcal{H}}_{k+1}$ 的点， $T_{(k+1, i)}^L(a:b)$ 是 $T_{(k+1, i)}^L$ 的第 a 到第 b 个条目， R 是由 Rodrigues 公式[25]定义的旋转矩阵，

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

在上式中， θ 是旋转的幅度，

$$\theta = \|\mathbf{T}_{(k+1,i)}^L(4:6)\|, \quad (7)$$

ω 是表示旋转方向的单位向量，

$$\omega = \mathbf{T}_{(k+1,i)}^L(4:6) / \|\mathbf{T}_{(k+1,i)}^L(4:6)\|, \quad (8)$$

$\hat{\omega}$ 是 ω 的斜对称矩阵；

回想一下 (2) 和 (3) 计算 $\bar{\mathcal{E}}_{k+1}$ 和 $\bar{\mathcal{H}}_{k+1}$ 中点之间的距离及其对应关系。结合 (2) 和 (4) —— (8)，我们可以推导出 \mathcal{E}_{k+1} 中的一个边缘点和对应的边缘线之间的几何关系，

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

类似地，结合 (3) 和 (4) —— (8)，我们可以在 \mathcal{H}_{k+1} 中的平面点和对应的平面贴片 (patch) 之间建立另一种几何关系，

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

最后，我们使用 Levenberg-Marquardt 方法 [26] 求解激光雷达运动。对 \mathcal{E}_{k+1} 和 \mathcal{H}_{k+1} 中的每个特征点叠加(9)和(10)，我们得到一个非线性函数，

$$f(T_{k+1}^L) = d, \quad (11)$$

其中 f 的每一行对应一个特征点， d 包含相应的距离。我们计算 f 关于 T_{k+1}^L 的雅可比矩阵，表示为 J ，其中 $J = \partial f / \partial T_{k+1}^L$ 。然后，(11) 可以通过将 d 最小化到零来通过非线性迭代求解，

$$T_{k+1}^L \leftarrow T_{k+1}^L - (J^T J + \lambda \text{diag}(J^T J))^{-1} J^T d. \quad (12)$$

λ 是由 Levenberg-Marquardt 方法确定的因子。

D、 激光雷达里程计算法

激光雷达里程计算法如算法 1 所示。该算法将上次扫描的点云 $\bar{\mathcal{P}}_k$ 、当前扫描的增长点云 \mathcal{P}_{k+1} 和上次递归的位姿变换 T_{k+1}^L 作为输入。如果开始新的扫描， T_{k+1}^L 将设置为零（第 4-6 行）。然后，该算法在第 7 行中从 \mathcal{P}_{k+1} 中提取特征点以构造 \mathcal{E}_{k+1} 和 \mathcal{H}_{k+1} 。对于每个特征点，我们在 $\bar{\mathcal{P}}_k$ 中找到其对应关系（第 9-19 行）。运动估计适用于稳健的拟合 [27]。在第 15 行，该算法为每个特征点分配一个双平方权重。与其对应关系距离较大的特征点分配较小的权重，距离大于阈值的特征点被视为异常值并分配零权重。然后，在第 16 行中，姿势变换被更新一次迭代。如果发现收敛或满足最大迭代次数，则非线性优化终止。如果算法到达扫描结束，则使用扫描期间估计的运动将 \mathcal{P}_{k+1} 重新投影到时间戳 t_{k+2} 。否则，只有转换 T_{k+1}^L 被返回用于下一轮递归。

Algorithm 1: Lidar Odometry

```
1 input :  $\bar{\mathcal{P}}_k, \mathcal{P}_{k+1}, T_{k+1}^L$  from the last recursion
2 output :  $\bar{\mathcal{P}}_{k+1}$ , newly computed  $T_{k+1}^L$ 
3 begin
4   if at the beginning of a sweep then
5      $T_{k+1}^L \leftarrow \mathbf{0}$ ;
6   end
7   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;
8   for a number of iterations do
9     for each edge point in  $\mathcal{E}_{k+1}$  do
10      Find an edge line as the correspondence, then compute
      point to line distance based on (9) and stack the equation
      to (11);
11    end
12    for each planar point in  $\mathcal{H}_{k+1}$  do
13      Find a planar patch as the correspondence, then compute
      point to plane distance based on (10) and stack the
      equation to (11);
14    end
15    Compute a bisquare weight for each row of (11);
16    Update  $T_{k+1}^L$  for a nonlinear iteration based on (13);
17    if the nonlinear optimization converges then
18      Break;
19    end
20  end
21  if at the end of a sweep then
22    Reproject each point in  $\mathcal{P}_{k+1}$  to  $t_{k+2}$  and form  $\bar{\mathcal{P}}_{k+1}$ ;
23    Return  $T_{k+1}^L$  and  $\bar{\mathcal{P}}_{k+1}$ ;
24  end
25  else
26    Return  $T_{k+1}^L$ ;
27  end
28 end
```

6、 激光里程计

建图算法的运行频率低于测距算法，每次扫描仅调用一次。在扫描 $k + 1$ 结束时，激光雷达里程计生成未失真的点云 $\bar{\mathcal{P}}_{k+1}$ ，同时生成一个位姿变换， T_{k+1}^L ，包含扫描期间的激光雷达运动，在 $[t_{k+1}, t_{k+2}]$ 之间。建图算法在世界坐标 $\{W\}$ 中匹配并注册 $\bar{\mathcal{P}}_{k+1}$ ，如图 8 所示。为了解释该过程，让我们将 Q_k 定义为地图上的点云，累积到扫描 k ，并令 T_k^W 是激光雷达在扫描 k 、 t_{k+1} 结束时在地图上的位姿。

利用激光雷达里程计的输出，建图算法将 T_k^W 扩展为从 t_{k+1} 到 t_{k+2} 的一次扫描，以获得 T_{k+1}^W ，并将 \bar{P}_{k+1} 投影到世界坐标 $\{W\}$ 中，表示为 \bar{Q}_{k+1} 。接下来，该算法通过优化激光雷达位姿 T_{k+1}^W 将 \bar{Q}_{k+1} 与 Q_k 匹配。

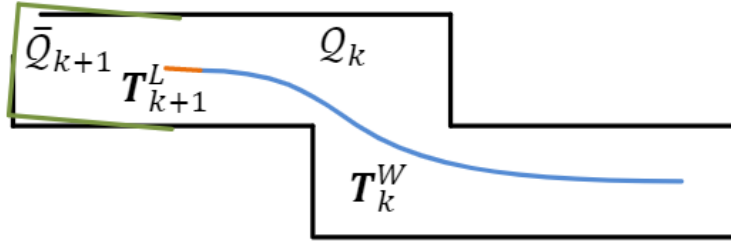


图 8：建图过程图解。蓝色曲线表示由扫描 k 处的建图算法生成的地图上的激光雷达姿态， T_k^W 。橙色曲线表示由里程计算法计算的扫描 $k+1$ 、 T_{k+1}^L 期间的激光雷达运动。使用 T_k^W 和 T_{k+1}^L ，里程计算法发布的无失真点云被投影到地图上，表示为 \bar{Q}_{k+1} （绿色线段），并与地图上的现有点云 Q_k （黑色线段）匹配。

提取特征点的方法与第 5-A 节相同，但使用了 10 倍的特征点。为了找到特征点的对应关系，我们将点云存储在地图 Q_k 的 10m 立方区域中。立方体中与 \bar{Q}_{k+1} 相交的点被提取并存储在 3D KD Tree 中 [24]。我们在特征点周围的某个区域内找到 Q_k 中的点。令 **S'** 为一组周围点。对于边缘点，我们只保留 S' 中边缘线上的点，对于平面点，我们只保留平面块上的点。然后，我们计算 S' 的协方差矩阵，表示为 M，以及 M 的特征值和特征向量，分别表示为 V 和 E。如果 S' 分布在一条边线上，则 V 包含一个明显大于其他两个的特征值，E 中与最大特征值相关联的特征向量表示边线的方向。另一方面，

如果 S' 分布在平面块上，则 V 包含两个较大的特征值，第三个特征值明显较小，并且 E 中与最小特征值相关联的特征向量表示平面块的方向。通过 S' 的几何中心确定边缘线或平面贴片的位置。

为了计算从特征点到对应点的距离，我们在边缘线上选择两个点，在平面块上选择三个点。这允许使用与 (2) 和 (3) 相同的公式计算距离。然后，对每个特征点推导出方程如 (9) 或 (10)，但不同的是 \bar{Q}_{k+1} 中的所有点共享相同的时间戳， t_{k+2} 。非线性优化通过 Levenberg-Marquardt 方法 [26] 的鲁棒拟合 [27] 再次求解，并且 \bar{Q}_{k+1} 在地图上注册。为了均匀分布点，地图点云通过体素网格过滤器 [28] 缩小，体素大小为 5cm 立方体。

姿态变换的集成如图 9 所示。蓝色区域表示激光雷达建图的姿态输出， T_k^W ，每次扫描生成一次。橙色区域表示激光雷达里程计 T_{k+1}^L 的变换输出，频率约为 10Hz。激光雷达相对于地图的姿态是两个变换的组合，频率与激光雷达里程计相同。

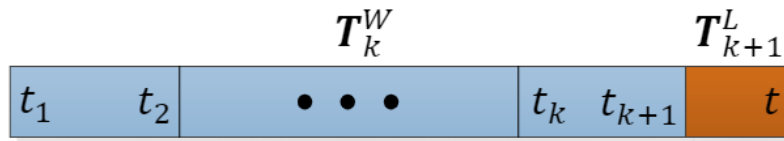


图 9：姿势变换的集成。蓝色区域显示了每扫描一次生成的建图算法， T_k^W 的激光雷达姿态。橙色区域是当前扫描范围内的激光雷达运动， T_{k+1}^L ，由里程计算法计算。激光雷达的运动估计是两种变换的组合，频率与 T_{k+1}^L 相同。