

LOAM:实时激光雷达里程计和建图

Ji Zhang and Sanjiv Singh

摘要: 我们提出了一种利用具有六自由度运动的两轴激光雷达进行测距和建图的实时方法。该问题的难点在于测距是在不同时间获得的,运动估计中造成的误差会引起最终点云匹配时的错误。目前连续的 3D 地图可以通过离线批处理方法来构建,其方法是通过循环闭合来修正随时间的漂移。我们提出的方法不需要高精度的测距以及惯性传感器的测量,就可以实现低漂移和低计算复杂度。获得这种性能级别表现的关键是将同时定位和建图的复杂问题进行划分,即分别通过两种算法同时优化大量的变量。其中一个算法执行高频率但低精度的雷达运动估计;另一个算法以一个低量级的频率进行点云的精确匹配和配准。通过这两种算法的结合,该方法能够用于实时建图。我们通过大量的实验和 KITTI 里程计标准测试对该算法进行了评价。测试结果表明该方法可以达到与最先进的离线批处理方法相同的精度。

一、引言

3D 建图目前仍然是一项流行的技术。利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范围的测量,在这种测量中测量距离的误差相对恒定。在激光雷达只有激光束的旋转运动的情况下,点云的配准过程是简单的。然而在大多数应用场景中,激光雷达自身是移动的,我们需要知道激光雷达在连续测距过程中的位姿信息才能准确地匹配。解决这一问题的一种常用方法是使用独立的位置估计系统(如 GPS/INS)将激光点匹配到一个固定的坐标系统中。另一种方法是使用里程测量法,如通过车轮编码器或视觉里程测量系统来匹配激光点。由于测程法产生随时间推移的小增量运动,产生漂移量,因此有很多人致力于减少漂移(如使用闭环)。

这里我们使用具有六自由度运动的两轴低漂移激光雷达。使用激光雷达的一个主要优点是它对环境光和场景中的光学纹理不敏感。最近激光雷达的发展已经减少了激光雷达的大小和重量,使得其可以由在环境中穿行的人拿着,也可以安装到微型飞行器上。由于我们提出方法的是为了尽量减少里程数估计的漂移,所

以目前不涉及环路的闭合。

该方法实现了低漂移和低计算复杂度，且不需要高精度测距和惯性传感器。获得这种水准表现的关键是将同时定位和建图（SLAM）的复杂问题进行划分，即分别通过两种算法同时优化大量的变量。其中一个算法执行高频率但低精度的雷达运动估计；另一个算法以一个低量级的频率进行点云的精确匹配和配准。提供的 IMU 接口不必要，但可以用来提供运动先验信息以帮助解释高频运动。具体来说，两种算法都提取了尖锐边缘和平面表面的特征点，并将特征点分别与边缘线和平面块匹配。在里程计算法中，通过保证快速计算找出特征点之间的对应关系。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。

通过对原始问题的分解，首先解决了运动估计这一比较简单的问题。然后进行批量优化建图(类似于迭代最近点方法(ICP) [9])，生成高精度的运动估计和地图。并行算法结构保证了可以实时求解。此外，由于运动估计是在一个较高的频率进行，建图有足够的时间来提高准确性。当以较低的频率运行时，建图算法能够合并大量的特征点，并使用足够多的迭代次数进行收敛。

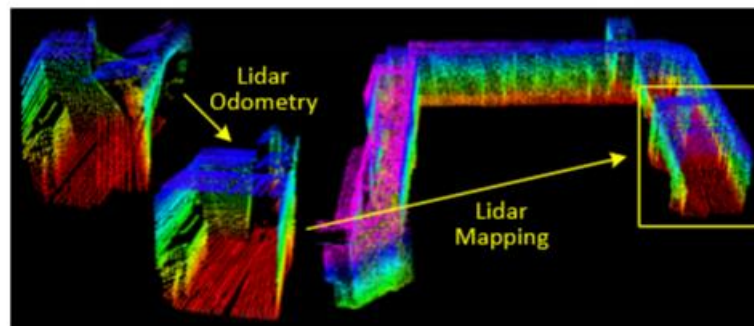


图 1. 该方法旨在利用移动 2 轴激光雷达进行运动估计和建图。由于激光点的接收时间不同，激光雷达的运动导致点云产生畸变(如图左激光雷达云所示)。我们提出的方法是将问题分解为两个并行运行的算法。该算法首先对激光雷达的速度进行估计，并对点云中的畸变进行校正，然后对点云进行匹配和配准以生成地图。两种算法的结合保证了实时求解问题的可行性。

二、相关工作

激光雷达已成为一种机器人导航方面有用的测距传感器。对于定位和建图，大多数应用场景使用的是 2D 激光雷达。当激光雷达的扫描速度比其外部运动速度快时，扫描过程中的运动畸变往往可以忽略不计。在这种情况下，可直接采用标准的 ICP 算法[12]匹配不同扫描返回的激光。另外，又提出了一种消除畸变的

两步方法：在 ICP 速度估计的基础上，利用计算得到的速度进行畸变补偿。类似的技术应用于对单轴三维激光雷达的畸变补偿。然而，如果扫描运动相对较慢，产生的运动畸变可能会很严重。使用二轴激光雷达尤为如此，因为一轴通常比另一轴慢得多。通常，通过其他传感器来提供速度测量就可以消除畸变。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匹配[15]。当多个传感器(如 GPS/INS 和车轮编码器)同时可用时，通常通过扩展的卡尔曼滤波[16]或粒子滤波[1]来解决问题。这些方法可以实时生成地图，辅助机器人在导航中进行路径规划和避撞。

如果使用二轴激光雷达而不借助于其它的传感器，那么运动估计和畸变校正就成为问题。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]来倒推地面车辆的运动[18]-[21]。在文献[18]、[19]中车辆运动被视为匀速运动，在[20]、[21]中又采用了高斯过程。在文献[18]-[21]的方法中，涉及到图像强度的视觉特征，需要密集的点云。而我们的方法提取并匹配了笛卡尔空间中的几何特征，对云密度的要求较低。

和我们的方法最接近的是 Bosse 和 Zlot 在文献[3]，[6]，[22]中的方法。他们利用二轴激光雷达获取点云，而点云是通过匹配局部点簇[22]的几何结构进行匹配的。此外利用了多台两轴激光雷达匹配地面[3]。这种方法结合了 IMU 和闭环映射来创建地图。他们又提出了 Zebedee，这是由一个二维激光雷达和一个通过弹簧连接到手动杆上的 IMU 组成的建图设备。其建图是通过手动点头设备进行的。该方法对分段数据集进行处理，在分段之间添加边界约束。该方法利用 IMU 的测量值对激光点进行配准，并利用优化方法对 IMU 误差进行校正。本质上，Bosse 和 Zlot 的方法需要批处理来开发精确的地图，因此不适合需要实时地图的应用场景。相比之下，该方法可以实时生成与 Bosse 和 Zlot 方法质量类似的地图。区别在于我们的方法可以为自动驾驶车辆的导航提供运动估计。该方法利用了激光雷达的扫描模式和点云分布的优势，在里程计算法和建图算法中分别实现了特征匹配，保证了计算速度和精度。

三、符号表示和任务描述

解决的问题是使用三维激光雷达获得的点云数据进行运动估计并构建遍历

过环境的地图。我们假设雷达预先校准，并且假设激光雷达的角速度和线速度是连续平滑没有突变的。第二个假设在第七节 B 部分中通过使用 IMU 简化实现。

作为惯例，我们在本文中使用大写字母来表示坐标系统。定义雷达的一次扫描作为一次完整覆盖的扫描。用 i 表示扫描次数， P_k 表示第 k 次扫描的点云， $k \in \mathbb{Z}^+$ 。定义两种坐标系如下：

- 激光雷达坐标系 $\{L\}$ 是以激光雷达几何中心为原点的三维坐标系。 X 向左， Y 向上， Z 向前。点 $i \in P_k$ 在局部坐标系 $\{L_k\}$ 中表示为 $X_{(k,i)}^L$ 。
- 世界坐标系 $\{W\}$ 是一个与 $\{L\}$ 在初始位置重合的三维坐标系。点 $i \in P_k$ 在世界坐标系 $\{W_k\}$ 中表示为 $XW(k,i)$ 。

根据假设和符号，我们的激光雷达测程和映射问题可以定义为：

问题： 对于给定一个激光雷达云图 p_k , kz + 序列，计算每次扫描 k 时激光雷达的自运动，用 p_k 构建被遍历环境的地图。

四、系统概述

A. 雷达硬件

本论文的研究在 Hokuyo UTM-30LX 3D 激光扫描仪上进行了验证，但应用不限于此。本文将通过激光雷达采集的数据来说明该方法。该激光雷达视野为 180° ，分辨率为 0.25° ，扫描速度为 40 行/s。激光扫描仪与电机相连，电机在 $-90^\circ \sim 90^\circ$ 之间以 $180^\circ/s$ 的角速度旋转，其水平方向为 0° 。在这个特殊的单元中，扫描从 -90° 转到 90° 或反向旋转(持续 1 秒)。在这里需要注意的是，对于一个连续旋转的激光雷达来说一次扫描只是进行一个半球形的旋转。装载编码器测量的电机旋转角度分辨率为 0.25° ，其中激光点投射到激光雷达坐标为 $\{L\}$ 。

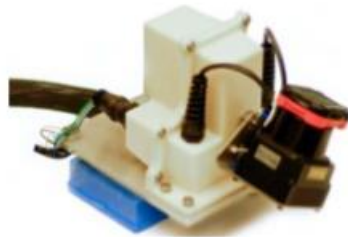


图 2. 本研究使用的三维激光雷达由通过驱动电机进行驱动旋转运动的 Hokuyo 激光扫描仪和一个测量旋转角度的编码器组成。激光扫描仪视野为 180° ，分辨率为 0.25° 。扫描速度为 40 行

/秒。该发动机控制激光扫描仪从 -90° 到 90° 的旋转，其水平方向为零。

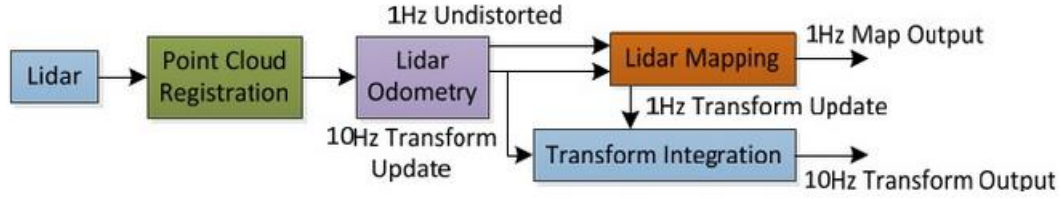


图 3. 激光雷达里程计和建图软件系统框图。

B. 软件系统

软件系统框图如图 3 所示。设 \hat{P} 为激光扫描中接收到的点。在每次扫描中将 \hat{P} 在 $\{L\}$ 中匹配。扫描 k 的激光云组成 P_k 。然后，用两种算法处理 P_k 。激光雷达里程计法获取并计算激光雷达在两次连续扫描之间的运动，用估计的运动来校正 P_k 中的畸变。该算法以 10Hz 左右的频率运行。并将处理后的结果给建图算法做进一步处理。建图算法利用输出，以 1Hz 的频率将未失真的点云匹配到地图上。最后由转换集成算法接收前面两个节点输出的转换后的信息，并将其进行融合处理，以频率为 10Hz 将激光雷达位姿相对于地图的变换输出。第五节和第六节详细介绍了软件图中的模块。

五、激光里程计

A. 特征点提取

我们从激光雷达点云 P_k 的特征点提取开始。图 2 所示的激光雷达在 P_k 中自然产生分布不均匀的点云。从激光扫描仪返回的图像在一次扫描中的分辨率为 0.25° 。这些点云分布在一个扫描平面上。但是，当固定激光雷达的轴以 $180^\circ/\text{s}$ 的角速度旋转，并产生 40Hz 的扫描时，与扫描平面垂直方向的分辨率为 $180^\circ/40 = 4.5^\circ$ 。在此基础上，利用共面几何关系，仅利用单个扫描的信息从 P_k 中提取特征点。

我们从锐变和平面表面提取特征点。设 i 为 P_k 中的一个点， $i \in P_k$ ，设 S 为同一扫描中激光扫描仪由 i 返回的连续点的集合。由于激光扫描仪是按 CW 或 CCW 顺序生成返回点云的，所以 S 包含了 i 左右两边各一半的点，每两个点之间有 0.25° 的间隔。定义一个参数来描述局部表面的平滑度：

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

根据 c 值进行排序，大的对应点为边缘特征点，小的对应点为平面特征点。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。每个子区域可以提供最多 2 边缘点和最多 4 平面点。点 i 可以通过阈值判断是边缘点还是平面点，并且满足点数不超过规定最大值。

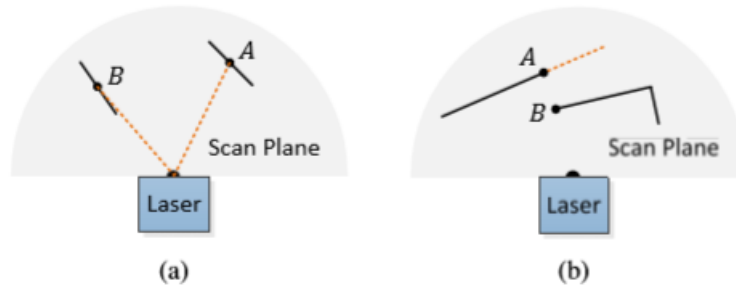


图 4. (a)实体线段表示局部表面。点 A 在与激光束(橙色虚线段)有一定角度的表面上。点 B 在一个大致平行于激光束的表面上。我们认为 B 是一个不可靠的激光返回，不选择它作为一个特征点。(b)固体线段是激光可观察到的物体。点 A 位于被遮挡区域(橙色虚线段)的边界上，可以作为边缘点检测。然而，如果从不同的角度观察，遮挡区域会发生变化，变得可见。

在选择点时希望避免选取已经选过点周围的点或者是在于激光线接近平行的平面上的点（如图 4(a)中的点 B），这些点通常认为是不可靠的，因为它们不能在任何时刻都可以看到。同时我们也希望避免可能遮挡点。如图 4(b)中的点 A 是激光雷达云中的一个边缘点，因为它所连接的表面(虚线段)被另一个物体遮挡所以应该避免。如果激光雷达移动到另一个角度，这个被遮挡的区域可能会改变并变得可见。为了避免上述的点被选择，我们再次找到点集 S 。一个可以选择的点需要满足的条件是， S 不形成一个近似平行于激光束的表面，在 S 中没有点是由于激光束方向上的间隙与 i 断开且同样靠近 i （如图 4(b)中的点 B）。

综上所述，从最大 c 值中依次选取特征点作为边缘点，从最小 c 值依次选取平面点，如果选取了一个点，其需满足：

- 所选择的边缘点或平面点的数量不能超过子区域的最大值；
- 它周围的点都没有被选中；
- 它不能在表面大致平行于激光的面上，或被遮挡的边界上。

从场景中提取特征点的示例如图 5 所示。边缘点和平面点分别用黄色和红色标

记。

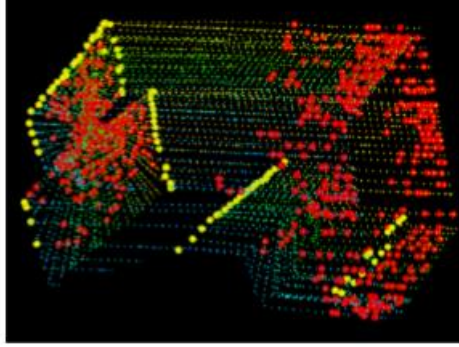


图 5. 一个从在走廊采集的激光雷达点云中提取边缘点(黄色)和平面点(红色) 的例子。此时激光雷达以0.5m/s的速度向图形左侧墙体移动，导致墙体产生运动畸变。

B. 寻找特征对应

里程计算法在扫描范围内估计激光雷达的运动。设 t_k 为扫描 k 的起始时间，每次扫描结束时，扫描过程中感知到的点云 P_k ，会对应到时间戳 t_{k+1} ，如图 6 所示。我们将重新投影的点云定义为 \bar{P}_k 。在下次扫描 $k + 1$ 时， \bar{P}_k 与新接收的点云 P_{k+1} 一起被用来估计激光雷达的运动。

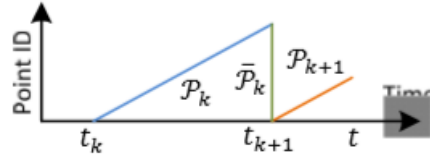


图 6. 将点云重新投射到扫描的末端。蓝色线段表示扫描 k , P_k 时感知到的点云，在扫描 k 结束时， P_k 被重新映射到时间戳 t_{k+1} ，得到 \bar{P}_k (绿色线段)。然后，在扫描 $k + 1$ 的时候， \bar{P}_k 和新感知的点云 P_{k+1} (橙色的线段)一起被用来估计激光雷达的移动。

让我们假设现在 \bar{P}_k 和 P_{k+1} 都是可用的，并开始寻找两个激光雷达云之间的对应。利用 P_{k+1} 和上一节讨论的方法，从中找到边缘点和平面点。设 E_{k+1} 和 H_{k+1} 分别是边点和平面点的集合。我们会发现 \bar{P}_k 中的边线对应于 E_{k+1} 中的点，而平面片对应于 H_{k+1} 中的点。

注意在扫描 $k + 1$ 的开始处 P_{k+1} 是一个空集，在扫描过程中它随着接收到更多的点而增长。激光雷达里程计递归估计扫描过程中的 6 自由度运动，并随着 P_{k+1} 的增加逐渐包括更多的点。在每个迭代中， E_{k+1} 和 H_{k+1} 被利用当前估计的转换重

新映射到扫描起始点的坐标下，记为 \tilde{E}_{k+1} 和 \tilde{H}_{k+1} 。并为每个 \tilde{E}_{k+1} 和 \tilde{H}_{k+1} 找到 \bar{P}_k 中最近的相邻点。 \bar{P}_k 存储在 3D KD-tree[24]中，用于快速索引。

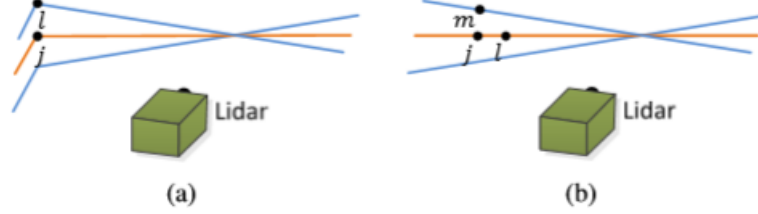


图 7. 在(a) \tilde{E}_{k+1} 中找到一条边线作为一个边点的对应，在(b) \tilde{H}_{k+1} 中找到一个平面片作为一个平面点的对应。在(a)和(b)中，j是最接近特征点的点。橙色的线表示相同的j扫描，蓝色的线表示连续两个扫描。找出(a)中的边线对应关系。我们在蓝色的直线上找到另一个点l，对应关系表示为(j,l)。为了找到(b)中的平面片对应关系，我们分别在橙色和蓝色的直线上找到另外两个点l和m，对应(j,l,m)。

图 7(a)表示找到边点对应边线的过程。令i为 \tilde{E}_{k+1} 中的一点， $i \in \tilde{E}_{k+1}$ 。边线通过两个点表示。令j为 \bar{P}_k 中最接近i的点，令l为j的临近两个连续扫描中i的最近邻点。(j,l)构成了i的对应关系。然后为了验证j和l都是边缘点，我们检查它们的平滑度c。在这里我们要求j和l来自不同的扫描，因为一个线在一个扫描中最多有一个边缘点。除非边线在扫描平面上，但若这样则边缘线退化，在扫描平面上呈现为直线，不应首先提取边缘线上特征点。

图 7(b)所示为平面点的对应关系。令i为 \tilde{H}_{k+1} 中的一点。局部平面片由三个点表示。依然首先找到 \bar{P}_k 中与i的最近点j。另外找到两个点l和m，l是在j的线扫上与i临近的另一点，m是在j相邻扫描线上找的一点，这样就可以保证三个点不在一条线上。为了验证j,l,m都是平面点，我们再次检查基于的局部曲面的光滑性。

利用特征点之间的对应关系，推导出计算特征点到特征点对应距离的表达式。对于点 $i \in \tilde{E}_{k+1}$ ，如果(j,l)为对应边界线， $j, l \in \bar{P}_k$ ，点到线的距离表示为：

$$d\epsilon = \frac{|(\bar{X}_{(k+1,i)}^L - \bar{X}_{(k,j)}^L) \times (\bar{X}_{(k+1,i)}^L - \bar{X}_{(k,l)}^L)|}{|(\bar{X}_{(k,j)}^L - \bar{X}_{(k,l)}^L)|} \quad (2)$$

其中 $\bar{X}_{(k+1,i)}^L$ ， $\bar{X}_{(k,i)}^L$ 和 $\bar{X}_{(k,j)}^L$ 是点i,j,l在 $\{L\}$ 中的坐标。然后对于 $i \in \tilde{H}_{k+1}$ ，如果(j,l,m)是相应的平面片， $j, l, m \in \bar{P}_k$ ，点到平面的距离为：

$$dH = \frac{\left| \frac{(\bar{X}_{(k+1,i)}^L - \bar{X}_{(k,j)}^L) \times ((\bar{X}_{(k,j)}^L - \bar{X}_{(k,l)}^L) \times (\bar{X}_{(k,j)}^L - \bar{X}_{(k,m)}^L))}{((\bar{X}_{(k,j)}^L - \bar{X}_{(k,l)}^L) \times (\bar{X}_{(k,j)}^L - \bar{X}_{(k,m)}^L))} \right|}{\left| (\bar{X}_{(k,j)}^L - \bar{X}_{(k,l)}^L) \times (\bar{X}_{(k,j)}^L - \bar{X}_{(k,m)}^L) \right|} \quad (3)$$

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

C. 运动估计

激光雷达运动在扫描过程中采用恒定的角速度和线速度进行建模。这允许我们在扫描中对不同时间接收到的点进行线性插值位姿变换。设 t 为当前时间戳， t_{k+1} 是扫描 $k+1$ 的起始时间。令 T_{k+1}^L 为 $[t_{k+1}, t]$ 之间的激光雷达位姿变换。 T_{k+1}^L 包含激光雷达的刚性运动 6 自由度， $T_{k+1}^L = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^T$ ，其中 t_x, t_y, t_z 沿着 $\{L\}$ 的 X 、 Y 、 Z 轴， $\theta_x, \theta_y, \theta_z$ 为旋转角度，遵循右手法则。对于给定点 i ，对于 $i \in P_k$ ，令 t_i 为其时间戳， $T_{(k+1,i)}^L$ 为在 $[t_{k+1}, t_i]$ 间的位姿变换。 $T_{(k+1,i)}^L$ 满足线性性：

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

回想一下， E_{k+1} 和 H_{k+1} 是 P_{k+1} 提取的边缘点和平面点的集合， \tilde{E}_{k+1} 和 \tilde{H}_{k+1} 是映射到扫描起始点 t_{k+1} 的集合。为了解决激光雷达的运动，我们需要在 E_{k+1} 和 \tilde{E}_{k+1} ， H_{k+1} 和 \tilde{H}_{k+1} 间建立一个几何关系。利用(4)中的变换，我们可以推出

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

其中 $X_{(k+1,i)}^L$ 是 E_{k+1} 或 H_{k+1} 中点 i 的坐标， $\tilde{X}_{(k+1,i)}^L$ 是 \tilde{E}_{k+1} 或 \tilde{H}_{k+1} 中点 i 的坐标。

$T_{(k+1,i)}^L(a:b)$ 表示 $T_{(k+1,i)}^L$ 中的第 a 项到第 b 项， R 是一个由 Rodrigues 定义的旋转矩阵[25]：

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

在上面的方程中， θ 是旋转的大小：

$$\theta = \left\| T_{(k+1,i)}^L(4:6) \right\| \quad (7)$$

ω 是表示旋转方向的单位向量：

$$\omega = T_{(k+1,i)}^L(4:6) / \left\| T_{(k+1,i)}^L(4:6) \right\| \quad (8)$$

$\hat{\omega}$ 是 ω 的斜对称矩阵。

回顾(2)和(3)计算点 \tilde{E}_{k+1} 和 \tilde{H}_{k+1} 之间的距离。结合(2)和(4)-(8)，可以得到 E_{k+1} 中一个边点与对应边线的几何关系：

$$f_{\varepsilon}(X_{(k+1,i)}^L, T_{k+1}^L) = d_{\varepsilon}, i \in E_{k+1} \quad (9)$$

同样的，结合(3)和(4)-(8)，我们可以在 H_{k+1} 中的平面点与对应的平面片之间建立另一个几何关系：

$$f_H(X_{(k+1,i)}^L, T_{k+1}^L) = d_H, i \in H_{k+1} \quad (10)$$

最后，我们用 LevenbergMarquardt 方法[26]求解激光雷达的运动。对 E_{k+1} 和 H_{k+1} 中的每个特征点叠加(9)和(10)，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 包含对应的距离。计算 f 关于 T_{k+1}^L 的雅克比矩阵，

记为 $J = \partial f / \partial T_{k+1}^L$ 。则(11)可通过非线性迭代求解使 d 趋于零：

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

λ 是一个 Levenberg-Marquardt 方法决定的参数。

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      $T_{k+1}^L \leftarrow 0$ ;
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 (12);
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

```

D. 激光雷达测距算法

激光雷达测程算法如算法 1 所示。该算法以上次扫描的点云 \bar{P}_k 、当前扫描的正在增长的点云 P_{k+1} 以及最后一次递归的位姿变换 T_{k+1}^L 为输入。如果开始新的扫描，则 T_{k+1}^L 设置为 0(第 4-6 行)。然后，算法从 P_{k+1} 中提取特征点，构造第 7 行中的 E_{k+1} 和 H_{k+1} 。对于每个特征点，我们在 \bar{P}_k 中找到其对应关系(9-19 行)。运动估计用于鲁棒拟合[27]。在第 15 行中，算法为每个特征点分配双平方权重。与它们的对应关系具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视为离群点并被分配零权重。然后在第 16 行中，姿势变换将更新一次。如果找到收敛或满足最大迭代次数，则非线性优化将终止。如果算法到达扫描结束处，则使用扫描期间的估计运动将 P_{k+1} 重新投影到时间戳 t_{k+2} 。否则，仅返回变换 T_{k+1}^L 进行下一轮递归。

六、激光雷达建图

建图算法的运行频率比测程算法低，每次扫描只调用一次。扫描 $k+1$ 结束后激光雷达测程生成一个无畸变的点云 \bar{P}_{k+1} ，并同时生成一个包含了 $[t_{k+1}, t_{k+2}]$ 间激光雷达运动的位姿 T_{k+1}^L 。建图算法在世界坐标系中匹配并匹配 T_{k+1}^L ，如图 8。为了解释这个过程，先定义 Q_k 为累积扫描到 k 次时地图上的点云， T_k^W 为第 k 次扫描结束时激光雷达在地图上的位姿。通过测程算法的输出，一次扫描后将 T_k^W 扩展到 T_{k+1}^W ，同时将 \bar{P}_{k+1} 投影到世界坐标系 $\{W\}$ 中记为 \bar{Q}_{k+1} 。然后通过对 T_{k+1}^W 的优化匹配 \bar{Q}_{k+1} 和 Q_k 。

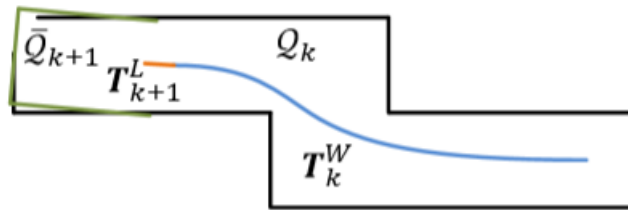


图 8. 建图过程的说明。蓝色曲线表示扫描 k 时激光雷达在地图上的位姿 T_k^W ，由映射算法通过扫描 k 生成。橙色曲线表示扫描 $k+1$ 时激光雷达的运动 T_{k+1}^L ，通过里程计算法计算得到。在 T_k^W 和 T_{k+1}^L 中，将里程计算法发布的不失真点云投射到地图上，标记为 \bar{Q}_{k+1} (绿色线段)，并与地图上现

有的云 Q_k (黑色线段)进行匹配。

建图算法特征点的提取方法与前面第五节 A 部分相同，但使用了十倍的特征点并将点云存储到一个十立方米的图上，记为 Q_k 。与 \bar{Q}_{k+1} 相交的点提取并存储到一个 3D KD-tree。我们发现 Q_k 中的点在特征点中的特定区域内。记周围的点为 S' 。对于边缘点，只保留其中在边线上的点，对于平面点，只保留其中平面上的点。然后记 M 为 S' 的协方差矩阵， M 的特征值和特征向量分别记为 V 和 E 。如果 S' 分布在边线上， V 包含一个明显大于其他两个的特征值，那个明显大的特征值对应的特征向量表示边缘线的方向。如果 S' 分布在平面上， V 包含两个大的特征值以及一个明显小的特征值，小的那个对应的特征值表示平面片的方向。边缘线和平面片的位置是通过 S' 的几何中心确定的。

为了计算距离，选择两个点，一个在边缘线，一个在平面上，可使用与 (2) (3) 相同的计算公式。然后，对每个特征点分别推导出(9)、(10)式，但不同的是 \bar{Q}_{k+1} 中的所有点都有相同的时间戳 t_{k+2} 。通过 Levenberg-Marquardt 方法进行鲁棒拟合，再次求解非线性优化问题，并在地图上匹配 \bar{Q}_{k+1} 。为了均匀分布这些点，地图云被一个大小 5 立方厘米的立体像素网格过滤器缩小。

位姿变换的积分如图 9 所示。蓝色区域表示激光雷达映射的位姿输出， T_k^W ，每次扫描生成一次。橙色区域表示来自激光雷达测程技术的变换输出， T_{k+1}^L ，频率为10Hz。激光雷达相对于地图的姿态是这两种变换的组合，频率与激光雷达里程计相同。

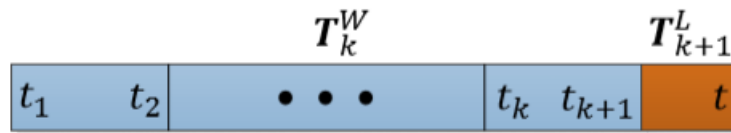


图 9. 姿态变换的整合。蓝色区域显示了来自映射算法 T_k^W 的激光雷达姿态，该算法每次扫描生成一次。橙色区域是激光雷达在当前扫描范围内的运动 T_{k+1}^L ，由测程算法计算。激光雷达的运动估计是这两种变换的组合，频率与 T_{k+1}^L 相同。

七、实验

在实验中，处理激光雷达数据的算法运行在一台 2.5GHz 四核 6Gib 内存的笔

记本电脑上，运行在 Linux 下的机器人操作系统(ROS)[29]上。该方法总共消耗两个核心，里程计和建图程序在两个独立的核心上运行。我们的软件代码和数据集是公开的。

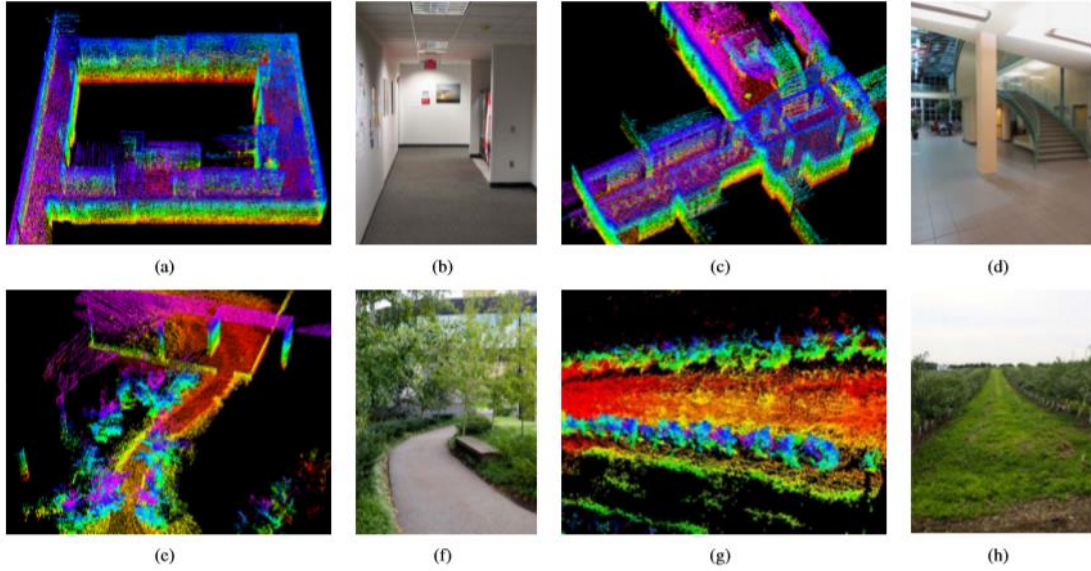


图 10. (a)-(b)一条狭长的走廊，(c)-(d)一个大的大厅，(e)-(f)一条植被覆盖的道路，(g)-(h)两排树之间的果园。在室内测试中，激光雷达被放在一个推车上，在室外测试中，它被安装在地面车辆上。所有的测试的移动速度都是0.5m/s。

A. 室内外实验

该方法已经在室内和室外环境中进行了测试。在室内测试时，激光雷达与电池和笔记本一起放在推车上，由一个人推着推车走。图 10（a）和图 10（c）展示了在两个代表性的室内环境（狭窄而又长的走廊和宽敞的大厅）中构建的地图。图 10（b）和图 10（d）展示了从相同场景拍摄的两张照片。在室外测试中，激光雷达安装在地面车辆的前部。图 10（e）和图 10（g）展示了从植被道路和树木生成的地图，并且在图 10（f）和图 10（h）的两行之间分别在果园中呈现的照片。在所有测试中，激光雷达以0.5m/s的速度移动。

为了评估地图的局部精度，我们从相同的环境中收集了第二组的激光雷达点云。在数据选择过程中，激光雷达保持静止，并放置在每个环境中的几个不同位置。利用点平面 ICP 方法对两个点云进行了匹配和比较。匹配完成后，将点云与第二点云中相应的平片视为匹配误差。误差分布密度如图 11 所示。室内匹配误

差小于室外匹配误差。这个结果是合理的，因为自然环境中的特征匹配不如人工环境中的精确。

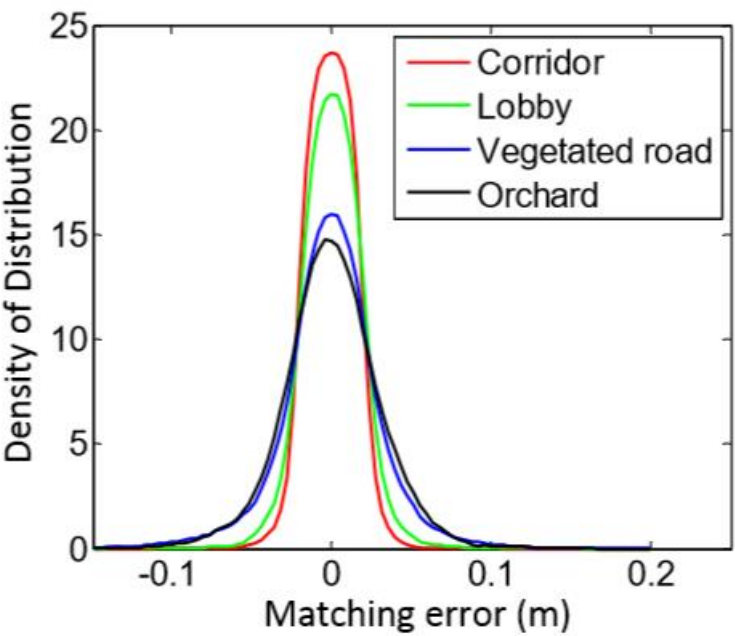


图 11. 廊道(红色)、大堂(绿色)、植被道路(蓝色)、果园(黑色)的匹配错误，对应图 10 中的四个场景。

此外，我们通过测试来测量运动估计值的累积漂移。我们选择包含闭环的室内实验走廊。这使我们可以在同一位置开始和结束。运动估计会在开始位置和结束位置之间生成缺口，该缺口指示漂移量。对于户外实验，我们选择果园环境。搭载激光雷达的地面车辆配备了用于地面真实位置采集的高精度 GPS / INS。将测得的漂移量与行驶距离进行比较，作为相对精度，列于表一。具体来说，测试 1 使用与图 10(a)和图 10(g)相同的数据集。一般来说，室内测试的相对准确度在 1%左右，室外测试的相对准确度在 2.5%左右。

TABLE I
RELATIVE ERRORS FOR MOTION ESTIMATION DRIFT.

Environment	Test 1		Test 2	
	Distance	Error	Distance	Error
Corridor	58m	0.9%	46m	1.1%
Orchard	52m	2.3%	67m	2.8%

B. IMU 辅助

我们将 Xsens MTi-10 IMU 连接到激光雷达，以处理快速的速度变化。点云经过两种方式进行预处理，然后再发送到所提出的方法：1) 从 IMU 定向，旋转一次扫描接收的点云，以使其与该扫描中激光雷达的初始方向对齐；2) 通过加速测量，激光雷达在扫描过程中以恒定速度移动，可以部分消除运动畸变。然后由激光雷达测距和制图程序处理点云。

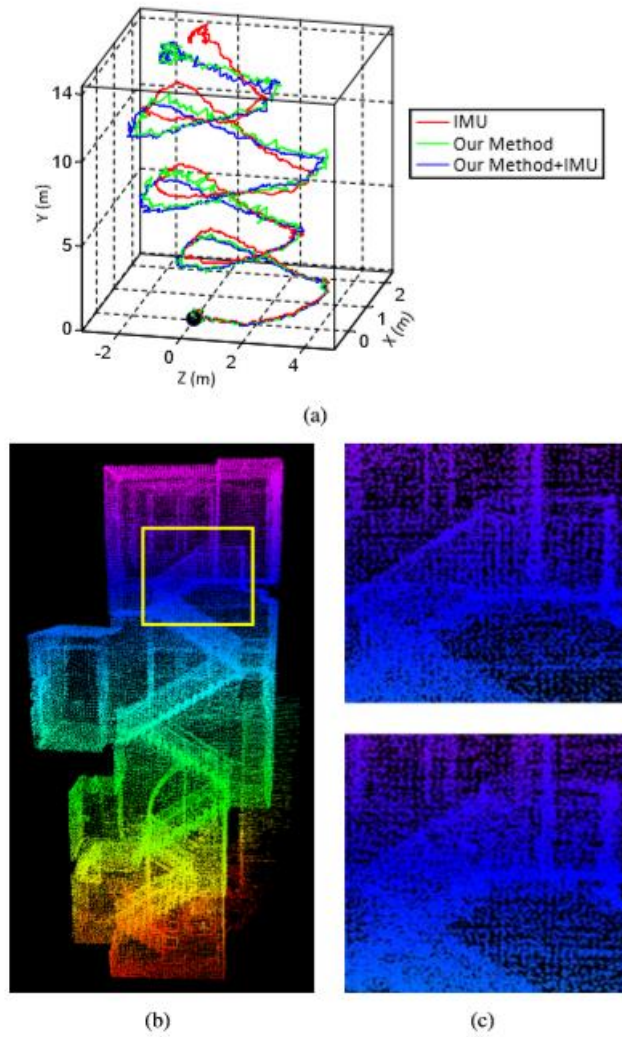


图 12. 比较有无 IMU 辅助的结果。一个人拿着激光雷达走在楼梯上。黑点是起点。(a)中，红色的曲线是利用 IMU 的方向计算得到的，我们的方法估计出平移量，绿色的曲线只依靠于我们的方法的优化，蓝色的曲线使用 IMU 数据进行预处理，然后使用我们的方法。(b)为蓝色曲线对应的地图。在(c)中，上、下两图分别对应蓝色和绿色曲线，分别使用(b)中黄色矩形标记的区域

IMU 的方向是通过积分陀螺的角速度和加速度计在卡尔曼滤波[1]中的读数来获得的。图 12(a)为样本结果。一个人拿着激光雷达走在楼梯上。在计算红色曲线时，我们使用 IMU 提供的方向，我们的方法只估计平移运动。在 5 分钟的数据收集过程中，方向漂移超过 25° 。绿色曲线只靠我们的方法中的优化，即假设没有 IMU 可用。蓝色曲线使用 IMU 数据进行预处理，然后使用所提出的方法。我们观察到绿色和蓝色曲线之间的细微差别。图 12(b)为蓝色曲线对应的地图。在图 12(c)中，我们比较了图 12(b)中黄色矩形中两个封闭的地图视图。上面的数字和下面的数字分别对应着蓝色和绿色的曲线。仔细比较发现，上图的边缘更清晰。

表 II 比较了使用和不使用 IMU 的运动估计中的相对误差。激光雷达由一个以 0.5m/s 速度行走，并以 0.5m 左右的幅度上下移动激光雷达的人握住。地面真实位置由卷尺手动测量。在所有四个测试中，在 IMU 的帮助下使用所提出的方法获得了最高的精度，而使用 IMU 的方向的只能得到最低的精度。结果表明，IMU 可以有效地消除非线性运动，从而解决了线性运动问题。

TABLE II
MOTION ESTIMATION ERRORS WITH/WITHOUT USING IMU.

Environment	Distance	Error		
		IMU	Ours	Ours+IMU
Corridor	32m	16.7%	2.1%	0.9%
Lobby	27m	11.7%	1.7%	1.3%
Vegetated road	43m	13.7%	4.4%	2.6%
Orchard	51m	11.4%	3.7%	2.1%

C. KITTI 数据集进行测试

我们还使用了 KITTI 里程表基准测试[30], [31]的数据集评估我们的方法。使用安装在结构化道路上的乘用车顶部的传感器（图 13 (a)）对数据集进行充分匹配。该车辆配备了 360° Velodyne 激光雷达，彩色/单色立体摄像头和用于地面实况的高精度 GPS / INS。激光雷达数据记录在 10Hz ，并使用我们的方法进行里程估计。由于篇幅所限，我们无法在此展示结果。但是我们鼓励读者在基准测试网站上查看我们的结果。

数据集主要涵盖三种类型的环境：周边有建筑的城市，周边有植被的乡村小

路，道路宽阔，周边环境相对干净的高速公路。图 13（b）展示了一个激光雷达云样本和城市环境中相应的视觉图像。数据集中包含 39.2 公里总行驶里程。上传车辆轨迹后，由基准服务器自动计算精度和排名。我们的方法在所有通过基准评估的方法中不考虑传感方式排名第一，包括最先进的立体视觉测程[32]，[33]。平均位置误差为使用轨迹段生成在三维坐标下长度等于 100 米、200 米、…、800 米的 0.88%。

八、总结与展望

利用旋转激光扫描仪的点云进行运动估计和建图比较困难，因为这涉及到恢复激光雷达云的运动和校正运动畸变。该方法通过两种并行算法进行分割和求解：激光雷达里程计算法进行粗处理以估计较高频率下的速度，而激光雷达建图算法进行精细处理以创建较低频率下的地图。这两种算法的协作可以实现精确的运动估计和实时的建图。该方法还可以利用激光雷达的扫描模式和点云分布的优势。为了保证测程算法计算速度快，保证映射算法的准确性，我们提出了一种基于特征匹配的测程算法。该方法已在室内和室外以及 KITTI 测程基准上进行了测试。

由于目前的方法不能识别环路闭合，我们未来的工作涉及到开发一种通过闭合环路来修正运动估计漂移的方法。此外，我们将我们的方法的输出与一个 IMU 进行卡尔曼滤波，以进一步减少运动估计的漂移。

参考文献

- [1] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics. Cambridge, MA, The MIT Press, 2005.
- [2] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, “iSAM2: Incremental smoothing and mapping using the bayes tree,” The International Journal of Robotics Research, vol. 31, pp. 217 – 236, 2012.
- [3] R. Zlot and M. Bosse, “Efficient large-scale 3D mobile mapping and surface reconstruction of an underground mine,” in The 7th International Conference on Field and Service Robots, Matsushima, Japan, July 2012.
- [4] K. Konolige, M. Agrawal, and J. Sol, “Large-scale visual odometry for rough terrain,” Robotics Research, vol. 66, p. 201212, 2011.

- [5] D. Nister, O. Naroditsky, and J. Bergen, “Visual odometry for ground vehicle applications,” *Journal of Field Robotics*, vol. 23, no. 1, pp. 3 – 20, 2006.
- [6] M. Bosse, R. Zlot, and P. Flick, “Zebedee: Design of a spring-mounted 3-D range sensor with application to mobile mapping,” *vol. 28, no. 5*, pp. 1104 – 1119, 2012.
- [7] S. Shen and N. Michael, “State estimation for indoor and outdoor operation with a micro-aerial vehicle,” in *International Symposium on Experimental Robotics (ISER)*, Quebec City, Canada, June 2012.
- [8] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part i the essential algorithms,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99 – 110, 2006.
- [9] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm,” in *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, Quebec City, Canada, June 2001.
- [10] A. Nuchter, K. Lingemann, J. Hertzberg, and H. Surmann, “6D SLAM 3D mapping outdoor environments,” *Journal of Field Robotics*, vol. 24, no. 8-9, pp. 699 – 722, 2007.
- [11] S. Kohlbrecher, O. von Stryk, J. Meyer, and U. Klingauf, “A flexible and scalable SLAM system with full 3D motion estimation,” in *IEEE International Symposium on Safety, Security, and Rescue Robotics*, Kyoto, Japan, September 2011.
- [12] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, “Comparing ICP variants on real-world data sets,” *Autonomous Robots*, vol. 34, no. 3, pp. 133 – 148, 2013.
- [13] S. Hong, H. Ko, and J. Kim, “VICP: Velocity updating iterative closest point algorithm,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, May 2010.
- [14] F. Moosmann and C. Stiller, “Velodyne SLAM,” in *IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, June 2011.
- [15] S. Scherer, J. Rehder, S. Achar, H. Cover, A. Chambers, S. Nuske, and S. Singh, “River mapping from a flying robot: state estimation, river detection, and obstacle mapping,” *Autonomous Robots*, vol. 32, no. 5, pp. 1 – 26, May 2012.

- [16] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “FastSLAM: A factored solution to the simultaneous localization and mapping problem,” in The AAAI Conference on Artificial Intelligence, Edmonton, Canada, July 2002.
- [17] H. Bay, A. Ess, T. Tuytelaars, and L. Gool, “SURF: Speeded up robust features,” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008.
- [18] H. Dong and T. Barfoot, “Lighting-invariant visual odometry using lidar intensity imagery and pose interpolation,” in The 7th International Conference on Field and Service Robots, Matsushima, Japan, July 2012.
- [19] S. Anderson and T. Barfoot, “RANSAC for motion-distorted 3D visual sensors,” in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, Nov. 2013.
- [20] C. H. Tong and T. Barfoot, “Gaussian process gauss-newton for 3D laser-based visual odometry,” in IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, May 2013.
- [21] S. Anderson and T. Barfoot, “Towards relative continuous-time SLAM,” in IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, May 2013.
- [22] M. Bosse and R. Zlot, “Continuous 3D scan-matching with a spinning 2D laser,” in IEEE International Conference on Robotics and Automation, Kobe, Japan, May 2009.
- [23] Y. Li and E. Olson, “Structure tensors for general purpose LIDAR feature extraction,” in IEEE International Conference on Robotics and Automation, Shanghai, China, May 9-13 2011.
- [24] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computation Geometry: Algorithms and Applications*, 3rd Edition. Springer, 2008.
- [25] R. Murray and S. Sastry, *A mathematical introduction to robotic manipulation*. CRC Press, 1994.
- [26] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. New York, Cambridge University Press, 2004.
- [27] R. Andersen, “Modern methods for robust regression.” Sage University Paper Series on Quantitative Applications in the Social Sciences, 2008.

- [28] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” in IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, May 9-13 2011.
- [29] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, “ROS: An open-source robot operating system,” in Workshop on Open Source Software (Collocated with ICRA 2009), Kobe, Japan, May 2009.
- [30] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The kitti vision benchmark suite,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 3354 – 3361.
- [31] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” International Journal of Robotics Research, no. 32, pp. 1229 – 1235, 2013.
- [32] H. Badino and T. Kanade, “A head-wearable short-baseline stereo system for the simultaneous estimation of structure and motion,” in IAPR Conference on Machine Vision Application, Nara, Japan, 2011.
- [33] A. Y. H. Badino and T. Kanade, “Visual odometry by multi-frame feature integration,” in Workshop on Computer Vision for Autonomous Driving (Collocated with ICCV 2013), Sydney, Australia, 2013.