（2014 投出，2016 发表）

- Abstract

提出一个低漂移的度量建图方案，使用 6 自由度的 3D 激光扫描仪做里程计测量。问题难点：测距信号返回时间点不同；运动估计误差导致点云匹配错误（尤其是没有 GPS 等外界参照时）。

如今，离线批处理方法(batch methods)建立连贯 3D 地图的方法常用回环检测修正随时间积累的漂移误差（后面部分提到 LOAM 无回环检测）（计算资源消耗更少）。

本文方案同时实现了低运动估计漂移误差和低计算复杂度。实现其的关键在于把 SLAM 变量优化问题拆分为两个算法，二者联合可达成实时性。第一个高频率里程计，精确度低，用于估计运动速度；且可选用 IMU 提供运动先验信息，减轻扭曲复杂(gross)的和高频的运动造成的影响。第二个算法频率比第一个低几个数量级，为的是点云匹配高精度。

做了室内实验、室外实验、benchmark 测试(KITTI 上)。结果在离线批处理方法中和先进水平相当。

---

1 Introduction

3D 建图是热门技术。“激光源运动的激光测距”和“结果点云匹配”互相关联。若仅有已知的激光束指向变化(属于 internal kinematics)，而基座固定，则匹配非常简单。然而，通常应用中传感器基座是运动的，则匹配还与外部运动有关，此时需要知道每个测距时刻传感器的定位和指向。激光测距每秒可运行数十万次，因此需要**高速的位姿估计**。一种常用解决方案是借助独立的位姿估计(如准确的 GPS/INS 系统)，用一个固定的坐标系，把距离数据整合到同一片点云中。外界方案不可用时，通常使用某种里程计估计(odometry estimation)方法来得到点的位置，例如轮速计、陀螺仪(gyros)、视觉图像特征的组合。

考虑情景：使用 6 自由度机械扫描激光测距设备(可选用 low-grade 惯性测量进行增强)执行低漂移里程计测量，从而建图。只用激光的好处：对周围环境光照和场景光学纹理不敏感。新的激光扫描器尺寸和重量较小，可装配于移动机器人(飞行、步行、轮式)和人身上。考虑到实时性需求，**不加入回环检测**。虽然回环检测能进一步减小漂移，但很多实用案例中(如楼层和建筑物建图)中回环检测都是不必要的。

- Fig. 1　激光雷达的运动会造成返回信号接收时点出现差异变化，造成失真(distortion)。LOAM 将问题解耦为两个并行的算法：一是里程计算法，估计激光雷达的运动速度，修正点云失真；二是建图算法，匹配并混合点云。二者联合使得实时性成为可能。

LOAM 可同时实现低 6 自由度运动估计漂移和低计算复杂度。关键在于把 SLAM 实时优化大量变量的问题拆分成两个算法：第一个是里程计算法，估计激光扫描器运动速度，高频率，低精确度，可选用 IMU 提供运动先验知识来应对复杂高频运动；第二个是建图算法，匹配并混合点云，频率比第一个低几个数量级，以达到更好精度。两个算法都提取边缘和平面表面的特征点，并将这些特征点分别与边缘 -线段(edge-line segments)和平面表面片(planar surface patches)匹配。里程计算法中，特征点匹配优先保证计算速度更快；而建图算法中，特征点匹配优先保证高准确度。

LOAM 中，先解决较简单的在线速度估计问题，然后使用批优化(batch optimization)处理建图问题以得到高精度的运动估计和地图。二者并行的结构使问题能够实时得到解。运动估计频率更高，使建图有充足时间做更高准确度，合并更多特征点，做更多次迭代以达到收敛。

主要贡献：

(1).双层优化的在线自我运动估计和建图软件系统；

(2).设计几何特征检测和匹配方法，适应需求：里程计算法的特征匹配粗糙且快速(为了高频率)，建图算法的特征匹配精确且缓慢(为了低漂移)；

(3).在很多数据集和不同种类的环境中做了测试；

(4).尽可能展现细节，以便读者复现。

文章结构：

2.相关工作；3.研究问题详述；4.所用硬件和软件的描述；5.里程计算法详述；6.建图算法详述；7.实验结果和讨论；8.总结。

---

2 Related work

激光雷达常在机器人导航中用作测距。

无其他传感器协助时，一个 2-axis 的激光雷达的运动估计和失真修正变为一个问题。

基于视觉的状态估计同样有运动分布(motion distribution)的问题。

从特征的角度，……

LOAM 产生的地图与 Bosse 和 Zlot 的结果定性地相似。不同之处在于 LOAM 可提供用于自动驾驶交通工具制导的运动估计。

本文是会议论文版本的扩充，做了更多实验，展示更多细节。

---

3 Notations and task description

本论文重点关注的问题是：使用 3D 激光雷达得到的点云来做自我运动估计，并且建立经过的场景的地图。假设 1：激光雷达的内在运动参数已经精确地标定了(使得激光点的 3D 投影成为可能)。假设 2：激光雷达的角速度和线速度随时间连续平滑变化，无突变(有 IMU 时可放松该假设，见 7.2 和 7.3)。

本文惯例记号：右斜大写字母上标表示坐标系。右斜下标正整数k表示第k次 sweep，P_k表示第k次扫 sweep 感知到的点云。其中一次 sweep 定义为：激光雷达完成一次扫描覆盖。

两个坐标系：

1.雷达坐标系$\{L\}$，是 3D 的。原点在激光雷达的几何中心。与惯例相同，使用右手系，x 坐标轴指左，y 指上，z 指前。$X^L_{(k,i)}$表示雷达坐标系下第 k 次 sweep 得到的第 i 个点。$T^L_k(t)$表示对雷达坐标系下在时间点 t 收到的点的投影变换(时间起点是第 k 次 sweep 的开始)。

2.世界坐标系$\{W\}$，是 3D 的。与$\{L\}$初始位姿重合。$X^W_{(k,i)}$表示世界坐标系下第 k 次 sweep 得到的第 i 个点。$T^W_k(t)$表示对世界坐标系下在时间点 t 收到的点的投影变换(时间起点是第 k 次 sweep 的开始)。

使用定义的符号重述**核心问题**：给定一个激光雷达点云$P_k\left(k \in Z^+\right)$序列，计算激光雷达在世界中的自我运动$T^W_k(t)$(定位)，并使用$P_k$建立经历的环境的地图(建图)。

---

4 System overview

4.1 激光雷达(Lidar)硬件

本论文实物验证使用四传感器系统：一个 back-and-forth spin 激光雷达，一个 continuously-spinning 激光雷达，一个 Velodyne HDL-32 激光雷达，以及 KITTI benchmark 使用的传感器系统。

其中第一个雷达用作描述系统的例子，在论文前面部分就介绍。其余在实验部分引入。

Fig. 2 的激光雷达：以一个 Hokuyo UTM-30LX 激光扫描器为基础。扫描器与一个在水平方向为 0 度的(-90°,90°)范围内以 180 度每秒的角速度旋转的马达相连。此处，一个 sweep 即一次从-90°到 90°的旋转，或者一次其反向的旋转，持续时间 1 秒。而对于 continuously-spinning 激光雷达而言，一个 sweep 即一次半球(semi-spherical)或完整球(full-spherical)旋转。机上有编码器，以 0.25 度的分辨率测量马达的旋转角，激光点可用它反投影到雷达坐标系中。

- Fig. 2(用一个单线激光雷达和电机组合构成多线激光雷达) 实验中用的 3D 激光雷达实例。该传感器由一个马达驱动旋转的 Hokuyo 激光扫描器和一个测量旋转角的编码器组成。视野宽 180 度，分辨率 0.25 度。扫描速度 40 线/秒。设水平方向为 0 度，则旋转范围是(-90°,90°)。

4.2 软件系统概述

如图 3 是软件系统框图。用$\hat{P}$表示一次激光扫描接收到的所有点。每个 sweep 过程中，$\hat{P}$都会被加入$\{L\}$，在第 k 个 sweep 中组合出点云P_k。随后，P_k送入两个算法处理。

1.激光雷达里程计用它算出连续两个相邻 sweep 间激光雷达的运动，估计结果用于修正P_k中运动导致的失真。该算法运行频率约 10Hz。

2.里程计的输出被激光雷达建图算法进一步处理。无失真的点云以 1Hz 的频率匹配并加入地图。地图输出频率是 1Hz。(建图输出)

3.两个算法提交(里程计 10Hz，建图 1Hz)的位姿变换被整合，产生 10Hz 的变换结果输出，代表着激光雷达关于地图的位姿。(定位输出)

Section 5 和 6 会详细介绍软件系统各个模块。

- Fig. 3(激光雷达里程计和建图软件系统框图)

---

5 里程计 Lidar odometry

5.1 特征点提取 Feature point extraction

从雷达点云P_k提取特征点。很多 3D 激光雷达自然产生的点云中的点都是不均匀(unevenly)分布的。例如图 2 的，一次扫描中分辨率是 0.25 度，点分布在一个扫描平面上(一个过转轴的面)。然而，由于激光扫描器以 180 度每秒的角速度(绕一个水平轴)旋转，产生扫描的频率是 40Hz(每秒 40 次扫描)，故在扫描平面的垂直方向(竖直方向)分辨率是 180°/40=4.5°。考虑到此，从P_k提取特征点时只使用单次扫描的信息，这些信息点有几何共面(共扫描面)的关系。

我们选出那些位于锋锐边缘和平面表面片上的特征点。用 i 表示P_k中的一个点，用 S 表示激光雷达在一次扫描中产生的与 i 连续的(共扫描面的)点的集合。激光雷达以顺时针(CW)或逆时针(CCW)顺序返回点。S 中位于 i 两侧的点各占一半，每两个相邻点间隔 0.25°。

定义一个衡量局部平面平缓程度的参数(约为连续点集里一个点和其它点距离的平均值):

$$c=\frac{1}{|S|\cdot\left\|X^L_{(k,i)}\right\|}\left\|\sum_{j\in S,j\neq i}\left(X^L_{(k,i)}-X^L_{(k,j)}\right)\right\| \tag{1}$$

该参数已用与激光雷达中心的距离归一化，以移除尺度效应，使远点和近点都可使用该参数。

同一个扫描面的点根据 c 值排序，选出 c 最大的点(边缘点)和 c 最小的点(平面点)作为特征点。为了使特征点均匀地(evenly)分布在环境中，**将一次扫描分成四个相同的子区域。每个子区域能提供最多 2 个边缘点和 4 个平面点**。一个点**只有在 c 值大于一个阈值(5×10^{-3})时才可被选为边缘点，只有在 c 值小于该阈值(5×10^{-3})时才可被选为平面点**。同时，**一个子区域选出点的数目不超过一个最大值**。

选点时，希望避免选到的点：1.周围点已经被选的点；2.所处局部平面表面和激光束几乎平行的点(Fig. 4a)。这些点通常被认为是不可靠的。同时，还希望避免选到遮挡区域边缘上的点。例如 Fig. 4b 中，点 C 在点云中被视为边缘点，因为它一侧和它相连的平面被前方更近的物体挡住了。而激光雷达移动到另一视点时，遮挡区域可能发生变化，变得可见。为了避免选到上面提到的这些点，我们重新评估 i 的相邻点集 S。一个点 i 只有在满足以下条件时才会被选到：1.S 组成的平面片的法线和激光束夹角在 10 度以外(排除平行于激光束的平面上的点)；2.S 中没有“与 i 间的间隙沿激光束方向”，同时“比 i 更靠近激光雷达”的点(排除被遮挡点)。

- Fig. 4(二维示意图)

a.实线代表局部平面片。点 A 所在平面片和激光束(橙色虚线)有一定的夹角。点 B 所在平面片和激光束几乎平行。点 B 被视为一个不可靠的激光返回点，不会被选为特征点。

b.实线是激光可见物体。点 D 是遮挡物的边缘。点 C 是一片被遮挡区域(橙色虚线)的边界，会被检测为边缘点。然而，从不同角度看时，遮挡区域可能变化，变得可见。点 C 不会被当作一个显著边缘点(salient edge point)，也不会被选为特征点。

总之，特征点的选择按照：从 c 最大值开始选边缘点，从 c 最小值开始选平面点。对于选出的点，满足:

1.选出的边缘点和平面点数目不可超过子区域的最大规定数目;

2.周围点没有被选中的;

3.不能位于一个法线和激光束夹角在 10 度以外的平面片上，不能在被遮挡区域的边界上。

- Fig. 5 例子：从一个走廊获得的点云提取出的边缘(黄)和平面(红)特征点。同时，激光雷达以 0.5m/s 的速度向图上左侧的墙移动，造成墙的运动失真。

5.2 特征点匹配 Finding feature point correspondence

里程计算法估计激光雷达在一个 sweep 内的运动。设t_k是第 k 次 sweep 的开始时间点。在第(k-1)次 sweep 中感知到的点云P_{k-1}会在这次 sweep 结束时被投影到时间戳t_k，如图 6(投影变换详见 5.3)。用$\bar{P}_{k-1}$表示投影所得点云。下次 sweep，即第 k 次中，用$\bar{P}_{k-1}$和新接收到的点云P_k一起估计激光雷达的运动。

- Fig. 6　在一个 sweep 的末尾投影点云。蓝色线段表示第(k-1)次 sweep 中感知到的点云P_{k-1}。在第(k-1)次 sweep 的末尾，P_{k-1}被投影到时间戳t_k(纵坐标是点的 ID，横坐标是时间，投影即把前一个 sweep 中不同时间点得到的不同 ID 的点水平移动到本次 sweep 的开始时间点，纵坐标不变，横坐标统一)，以得到$\bar{P}_{k-1}$(绿色线段)。接着，第 k 次 sweep 中，$\bar{P}_{k-1}$和新感知到的点云P_k(橙色线段)一同被用于估计激光雷达运动。

假设$\bar{P}_{k-1}$和P_k都可用，则可开始在两片点云之间找对应点。在P_k中，用 5.1 讨论的方法从点云中找边缘点和平面点。用ε_k和H_k分别表示边缘点集和平面点集。我们将从$\bar{P}_{k-1}$中找出与ε_k中的点对应的边缘线，以及与H_k中的点对应的平面片。

注意，第 k 次 sweep 开始时，P_k是空集，随后它在 sweep 中随着更多点被接收而变大。激光雷达里程计递归地估计 sweep 过程中的六自由度运动，并且随P_k增长，将更多点包含在内。ε_k和H_k被投影到下个 sweep 开始时，分别用$\tilde{\varepsilon}_k$和$\tilde{H}_k$表示投影所得点集。对$\tilde{\varepsilon}_k$和$\tilde{H}_k$中的每个点，我们要找到$\bar{P}_{k-1}$中最近的邻点。为快速索引，$\bar{P}_{k-1}$以$\{L_k\}$存储在一张三维 KD 树中。

图 7a 表示找出一个边缘点对应的边缘线的过程。设 i 是$\tilde{\varepsilon}_k$中的一个点。边缘线用两个点表示。设 j 是 i 在$\bar{P}_{k-1}$中最邻近的点，l 是 i 在 j 所在扫描的前后两次扫描中最邻近的点。(j,l)即组成 i 的对应。接着，为验证 j 和 l 都是边缘点，我们用公式(1)检查局部表面的平滑度，需要两个点都满足$c>5\times10^{-3}$。此处，我们尤其需要 j 和 l 来自不同的扫描，因为一次扫描不能包含来自同一条边缘线的一个以上的点(唯一一种例外情况是边缘线在扫描平面上，然而此时边缘线会退化，在扫描平面上表现为一条直线，**此边缘线上的点从一开始就不应该被选为特征点**)。

图 7b 表示找出一个平面点对应的平面片的过程。设 i 是$\tilde{H}_k$中的一个点。平面片用三个点表示。与上一段相似，找出 i 在$\bar{P}_{k-1}$中最邻近的点，用 j 表示。接着，我们找出 i 的另外两个最邻近点 l 和 m，其中一个和 j 在同一次扫描中但不是 j，另一个在 j 所在扫描的前后两次扫描中。由此可保证三个点不共线。为验证 j，l，m 都是平面点，我们检查局部表面平滑度，需要$c<5\times10^{-3}$。

- Fig.7　a 是找到$\tilde{\varepsilon}_k$中边缘点对应边缘线；b 是找到$\tilde{H}_k$中平面点对应平面片。a 和 b 中都有：j 是$\bar{P}_{k-1}$中找出的离特征点 i 最近的点。橙色线表示 j 所在的扫描，蓝色线表示前一次和后一次扫描。a 中，为了找出对应边缘线，在蓝色线上找另一点 l，对应边缘线表示为(j,l)；b 中，为了找到对应平面片，在橙色线和蓝色线上分别再找一点 l 和 m，对应平面片表示为(j,l,m)。

找出特征点云间的对应后，我们给出计算一个特征点和它的对应间的距离的表达式。下一节(5.3)中，我们将用最小化特征点总体距离的方法复原激光雷达运动。我们从边缘点开始：对于$i\in\tilde{\varepsilon}_k$，若(j,l)是它对应的边缘线，则点到线距离可由下式算出(i 到它对应的边缘线两端的距离向量的外积，即这两条边平移构成的平行四边形的面积，等于边缘线和一条边平移构成的平行四边形的面积，除以边缘线的长度，得到平行四边形边缘线对应的高)：

$$d_\varepsilon=\frac{\left|\left(\tilde{X}^L_{(k,i)}-\bar{X}^L_{(k-1,j)}\right)\times\left(\tilde{X}^L_{(k,i)}-\bar{X}^L_{(k-1,l)}\right)\right|}{\left|\bar{X}^L_{(k-1,j)}-\bar{X}^L_{(k-1,l)}\right|} \tag{2}$$

其中$\tilde{X}^L_{(k,i)}$，$\bar{X}^L_{(k-1,j)}$，$\bar{X}^L_{(k-1,l)}$分别是点 i，j，l 在$\{L_k\}$中的坐标。接着，对于$i\in\tilde{H}_k$，若(j,l,m)是它对应的平面片，则点到面距离(分子求四棱柱体积，方法是侧棱和底面的内积，再除以分母，即除以平面片上三点构成的底面的面积，得到四棱柱的高)：

$$d_H=\frac{\left|\left(\tilde{X}^L_{(k,i)}-\bar{X}^L_{(k-1,j)}\right)\cdot\left(\left(\bar{X}^L_{(k-1,j)}-\bar{X}^L_{(k-1,l)}\right)\times\left(\bar{X}^L_{(k-1,j)}-\bar{X}^L_{(k-1,m)}\right)\right)\right|}{\left|\left(\bar{X}^L_{(k-1,j)}-\bar{X}^L_{(k-1,l)}\right)\times\left(\bar{X}^L_{(k-1,j)}-\bar{X}^L_{(k-1,m)}\right)\right|} \tag{3}$$

5.3 运动估计 Motion estimation

激光雷达在一个 sweep 中的运动被建模为常数角度和线性速度的。这使得对一个 sweep 内不同时间接收到的点，我们都可以用线性插值得到其位姿变换。设现在时间戳是 t，t_k是现在第 k 次扫描的开始时间。设$T_k^L(t)$是$[t_k,t]$间的激光雷达位姿变换。$T_k^L(t)$由激光雷达的六自由度运动构成，$T_k^L(t)=[\tau_k^L(t),\theta_k^L(t)]^T$，其中$\tau_k^L(t)=[t_x,t_y,t_z]^T$是$\{L_k\}$中的平移，$\theta_k^L(t)=[\theta_x,\theta_y,\theta_z]^T$是$\{L_k\}$中的旋转。给定$\theta_k^L(t)$后，对应的旋转矩阵可由 Rodrigues(罗德里格斯)公式给定：

$$\boldsymbol{R}_k^L(t)=e^{\hat{\theta}_k^L(t)}=\boldsymbol{I}+\frac{\hat{\theta}_k^L(t)}{\|\theta_k^L(t)\|}\sin\|\theta_k^L(t)\|+\left(\frac{\hat{\theta}_k^L(t)}{\|\theta_k^L(t)\|}\right)^2(1-\|\cos\theta_k^L(t)\|) \tag{4}$$

其中$\hat{\theta}_k^L(t)$是$\theta_k^L(t)$的反对称矩阵(skew-symmetric Matrix)。

给定一个P_k中的点 i，用$t_{(k,i)}$表示它的时间戳，用$T_{(k,i)}^L$表示$[t_k,t_{(k,i)}]$间的位姿变换。$T_{(k,i)}^L$可由对$T_k^L(t)$的线性插值计算得到：

$$T_{(k,i)}^L=\frac{t_{(k,i)}-t_k}{t-t_k}T_k^L(t) \tag{5}$$

此处应注意变换$T_k^L(t)$是一个随时间变化的变量，插值使用的是现在时间 t 的变换。前面提到用ε_k和H_k分别表示从P_k提取的边缘点集和平面点集。下式将ε_k和H_k投影到 sweep 的开始时刻，即得到$\widetilde{\varepsilon}_k$和$\widetilde{H}_k$：

$$\widetilde{X}_{(k,i)}^L=R_{(k,i)}^L X_{(k,i)}^L+\tau_{(k,i)}^L \tag{6}$$

其中$X_{(k,i)}^L$是ε_k或H_k中的一个点坐标，$\widetilde{X}_{(k,i)}^L$是其在$\widetilde{\varepsilon}_k$或$\widetilde{H}_k$中的对应点坐标。$R_{(k,i)}^L$和$\tau_{(k,i)}^L$分别是$T_{(k,i)}^L$对应的旋转矩阵和平移向量。

前面提到式(2)和(3)分别计算$\widetilde{\varepsilon}_k$和$\widetilde{H}_k$中的点与其对应点间的距离。组合(2)和(6)可以得到ε_k中一个边缘点与其对应的边缘线间的一个几何关系：

$$f_\varepsilon(X_{(k,i)}^L,T_k^L(t))=d_\varepsilon,i\in\varepsilon_k \tag{7}$$

相似地，组合(3)和(6)可以建立H_k中一个平面点与其对应平面片间的一个几何关系：

$$f_H(X_{(k,i)}^L,T_k^L(t))=d_H,i\in H_k \tag{8}$$

最终，我们用 Levenberg-Marquardt(LM)方法解出激光雷达的运动。把ε_k和H_k中每个特征点对应的(7)和(8)堆叠起来，得到一个非线性方程：

$$\boldsymbol{f}(T_k^L(t))=\boldsymbol{d} \tag{9}$$

其中 **f** 的每一行对应一个特征点，**d** 包含点们对应的距离。计算 **f** 关于$T_k^L(t)$的雅可比矩阵 **J**，即$\boldsymbol{J}=\partial f/\partial T_k^L(t)$。随后，通过将 **d** 最小化到零，可由非线性迭代解(9)：

$$T_k^L(t)\leftarrow T_k^L(t)-(\boldsymbol{J}^T\boldsymbol{J}+\lambda\,\mathrm{diag}(\boldsymbol{J}^T\boldsymbol{J}))^{-1}\boldsymbol{J}^T\boldsymbol{d} \tag{10}$$

其中λ是用 LM 方法确定的一个因子。

5.4 激光雷达里程计算法 Lidar odometry algorithm

激光雷达里程计算法如算法 1 所示。算法输入是上个 sweep 的点云$\bar{P}_{k-1}$，现在 sweep 中不断增长的点云P_k，以及作为初始猜测的来自上次递归的位姿变换$T_k^L(t)$。当一个新 sweep 开始时，$T_k^L(t)$置零以重新初始化(第 4-6 行)。随后，从P_k提取特征点以构建ε_k或H_k(第 7 行)。对每个特征点，找到它在$\bar{P}_{k-1}$中的对应(第 9-19 行)。运动估计被调整为一个鲁棒拟合的架构(robust fitting framework)(Andersen 2008)。在第 15 行，每个特征点根据下式被赋予一个双平方权重(bisquare weight)。距离自己的对应更远的特征点的权重更小；距离大于一个阈值的特征点被视作 outlier(离群点)并被赋予零权重。

$$w=\begin{cases}(1-\alpha^2)^2,-1<\alpha<1\\0,otherwise\end{cases},\quad \alpha=\frac{r}{6.9459\,\sigma\sqrt{1-h}} \tag{11}$$

上式中，r 是最小二乘问题中对应的残差，σ是残差相对于中位值(median)的绝对偏差(absolute

deviation)，h 是杠杆值或矩阵$\boldsymbol{J}\left(\boldsymbol{J}^T\boldsymbol{J}\right)^{-1}\boldsymbol{J}^T$对角线上的对应元素，其中 **J** 是(10)中的雅可比矩阵。随后，在第 16 行，位姿变换进行一个迭代的更新。非线性优化在收敛时或达到最大迭代次数时终止。当一个 sweep 结束时，用 sweep 中估计的运动把P_k投影到时间戳t_{k+1}，形成$\bar{P}_k$，给下一个 sweep 和$\bar{P}_k$的匹配做好准备；若未到 sweep 末尾，则只返回变换$T_k^L(t)$，用于下轮递归。

算法 1：激光雷达里程计	
1	**输入**：$\bar{P}_{k-1}$，P_k，来自上次递归的$T_k^L(t)$
2	**输出**：$\bar{P}_k$，新算出的$T_k^L(t)$
3	**begin**
4	----|**if** 是一个 sweep 的开始 **then**
5	----|----|$T_k^L(t)\leftarrow 0$
6	----|**end**
7	----|----|P_k中找出边缘点和平面点，分别放入ε_k或H_k
8	----|**for** 某个迭代次数 **do**
9	----|----|**for** ε_k中每个边缘点 **do**
10	----|----|----|找出一个边缘线作为对应，然后用(7)计算点到线距离，并堆叠到(9)
11	----|----|**end**
12	----|----|**for** H_k中每个平面点 **do**
13	----|----|----|找出一个平面片作为对应，然后用(8)计算点到面距离，并堆叠到(9)
14	----|----|**end**
15	----|----|给(9)每一行算出一个 bisquare(实现是这样吗?)权重
16	----|----|根据(10)对$T_k^L(t)$做一个非线性迭代的更新
17	----|----|**if** 非线性优化收敛 **then**
18	----|----|----|**Break**
19	----|----|**end**
20	----|**end**
21	----|**if** 是一个 sweep 的末尾 **then**
22	----|----|把P_k中每个点投影到t_{k+1}，形成$\bar{P}_k$
23	----|----|返回$T_k^L(t)$和$\bar{P}_k$
24	----|**end**
25	----|**else**
26	----|----|返回$T_k^L(t)$
27	----|**end**
28	**end**

6　建图 Lidar mapping

建图算法以一个低于里程计算法的频率运转，每个 sweep 只调用一次。在第 k 个 sweep 末尾，激光雷达里程计产生一个无失真点云$\bar{P}_k$，并即时地产生位姿变换$T_k^L(t_{k+1})$，包含了激光雷达在这次 sweep 中，即

$[t_k, t_{k+1}]$间的运动。建图算法对$\bar{P}_k$做匹配并将其插入世界坐标系$\{W\}$，如图 8 所示。为解释该过程，定义Q_{k-1}为地图上到第 k-1 个 sweep 为止积累的点云，并用$T_{k-1}^{W}(t_k)$表示第 k-1 个 sweep 末尾时，即t_k时激光雷达在地图上的位姿。利用里程计的输出，建图算法把$T_{k-1}^{W}(t_k)$延伸一个 sweep，即从t_k到t_{k+1}，得到$T_k^{W}(t_{k+1})$，并把$\bar{P}_k$变换到世界坐标系$\{W\}$中，记作$\bar{Q}_k$。然后通过优化雷达位姿$T_k^{W}(t_{k+1})$来匹配$\bar{Q}_k$和Q_{k-1}。

- Fig.8　建图过程示意图。蓝色曲线表示激光雷达在地图上的位姿$T_{k-1}^{W}(t_k)$，由建图算法在第 k-1 个 sweep 时生成。橙色曲线表示激光雷达在整个第 k 次 sweep 中的运动$T_k^{W}(t_{k+1})$，由里程计算法算出。利用$T_{k-1}^{W}(t_k)$和$T_k^{W}(t_{k+1})$，里程计算法给出的无失真点云被投影到地图中，记作$\bar{Q}_k$(绿色线段)，并和地图上已有的点云Q_{k-1}(黑色线段)匹配。

用和 5.1 相同的方法提取特征点，但所用特征点的数目变为 10 倍。为找到特征点的对应，把点云在地图Q_{k-1}上以 10 米边长的立方体区域存储。提取和$\bar{Q}_k$相交的立方体中的点，存储在$\{W\}$里的一个三维 KD 树中。找出Q_{k-1}中在特征点周围一定范围内(10cm×10cm×10cm)的点，用S'表示周围点的集合。对于一个边缘点，只保留S'中边缘线上的点；对一个平面点，只保留平面片上的点。用 c 值区分边缘点和平面点，使用的阈值和 5.1 相同(5×10^{-3})。随后，计算S'的协方差矩阵，记作$\boldsymbol{M}$，$\boldsymbol{M}$的特征值和特征向量分别记作$\boldsymbol{V}$和$\boldsymbol{E}$。这些值决定了点簇的位姿，进而决定点到线和点到面距离。具体而言，若S'分布在一条边缘线上，则$\boldsymbol{V}$包含一个显著大于其他两个特征值的特征值，并且$\boldsymbol{E}$中最大特征值对应的特征向量表示了边缘线的方向；另一方面，若S'分布在一个平面片上，则$\boldsymbol{V}$包含两个大特征值，第三个显著地小，并且$\boldsymbol{E}$中最小特征值对应的特征向量指示了平面片的方向。算出边缘线或平面片的位置，使得线或面穿过S'的质心(centroid)。

为计算一个特征点到它的对应(边缘线或平面片)的距离，每条特征线上选 2 个点，每个平面片上选 3 个点。此时可用式(2)和(3)计算距离。然后，每个特征点可得到一个(7)或(8)等式，但有所不同，因为$\bar{Q}_k$中的所有点时间戳相同，都是t_{k+1}。仍然使用适应 robust fitting 的 LM 方法解非线性优化问题，随后$\bar{Q}_k$被插入地图。

为使点分布均匀(evenly)，每当一个新扫描(scan)合并到地图中时，使用体素栅格滤波器(voxel-grid filters)缩小精简(downsize)地图整体点云。体素栅格滤波器将每个体素中的点平均化(average)，在体素中产生一个平均点(averaged point)。边缘点和平面点使用的体素尺寸不同：边缘点体素尺寸 5cm×5cm×5cm；平面点体素尺寸 10cm×10cm×10cm。地图被截断到传感器周围 500m×500m× 500m 区域内以限制内存使用。

位姿变换的整合(integration)如图 9 所示。蓝色区域表示每个 sweep 产生一次的激光雷达建图输出位姿$T_{k-1}^{W}(t_k)$。橙色区域表示激光雷达里程计以 10Hz 左右频率输出的变换$T_k^{L}(t)$。地图中激光雷达位姿是二者的组合，频率与激光雷达里程计相同。

- Fig.9　位姿变换整合。蓝色区域表示来自建图算法的激光雷达位姿$T_{k-1}^{W}(t_k)$，每个 sweep 产生一次。橙色区域是现在 sweep 中激光雷达的运动$T_k^{L}(t)$，由里程计算法算出。激光雷达的运动估计是二者的组合，频率和$T_k^{L}(t)$相同。

7　Experiments

实验过程中，处理激光雷达数据的算法在一个有 2.5GHz 四核处理器和 6Gib 内存的笔记本电脑上运行，基于 Linux 系统中的 ROS(robot operating system)。方案总共使用 2 个线程。里程计和建图程序在两个彼此独立的线程上运行。代码/数据集：wiki.ros.org/loam_back_and_forth，wiki.ros.org/loam_continuous

---

7.1 准确度测试 Accuracy tests

方案已用图 2 所示激光雷达进行了室内和室外环境测试。室内测试期间，激光雷达和一块电池以及一台笔记本电脑共同被置于一辆手推车上。一个人步行推着手推车移动。图 10a,c 是两个有代表性的室内环境中建立的地图。图 10b,d 是两个场景中照的照片。室外实验中，激光雷达被安装在一辆地面载具的前端。图 10e,g 是在一条有植被的路和一个两行树间的果园生成的地图，照片如图 10f,h。所有测试中，激光雷达以 0.5m/s 的速度移动。

- Fig.10　实际场景中生成的地图和采集的照片。

为了评估地图的局部精确度，我们从相同场景收集了第二组激光雷达点云。每个场景中，激光雷达在数据选择期间保持静止不动，并被放置在数个不同的位置。两组点云被匹配并用点到面 ICP 方法(point to plane ICP method)比较。匹配完成后，计算一个点云和它在另一个点云中对应平面片的距离，作为匹配误差。图 11 展示了误差的密度分布，它表明室内场景匹配误差小于室外场景。这个结果是合理的，因为自然环境中的特征匹配准确度比人造环境低。

- Fig.11　图 10 四个场景的匹配误差：走廊(红色)，大厅(绿色)，有植被的路(蓝色)，果园(黑色)。

进一步，我们想理解激光雷达里程计和建图对最终准确度的影响和贡献方式。为此，我们使用图 1 中的数据集并展示每个算法的输出。轨迹长 32 米。图 12a 直接使用里程计输出来拼合激光点；图 12b 是建图优化后的最终输出。正如我们在开头提到的，里程计的角色是估计速度并消除点云中的运动失真。里程计的低准确性无法保证精确建图。另一方面，建图做了深入仔细的扫描匹配以保证地图的精确。表格 1 展示了两个程序在精确度测试中的计算时间。建图移除漂移所用时间是里程计的 6.4 倍。注意此处建图每被调用一次，里程计被调用 10 次，导致两个线程的计算负担实际相当。

- Fig.12　比较：a 里程计输出和 b 最终建图输出。使用图 1 数据集。里程计的角色是估计速度和消除点云运动失真，其算法的准确度低。建图深入仔细地执行扫描匹配以保证地图精确度。
- Table 1

此外，我们做了测试来测量运动估计的累积漂移。我们选择有闭环的走廊进行室内实验。这使起点和终点在同一位置。运动估计会在起始位置和结束位置间产生一个间断(gap)，它显示了漂移的大小。**室外实验**则选择果园环境。**携带激光雷达的地面载具装备了高精度 GPS/INS 以取得真实参考数据**。测得的漂移和移动距离作比较，以得到相对准确度，如表 2。具体而言，测试 1 使用了和图 10a,g 相同的数据集。总体上，室内测试相对准确度(误差)在 1%左右，室外测试在 2.5%左右。

- Table 2 运动估计漂移相对误差。

7.2 加入 IMU 辅助 Tests with IMU assistance

我们把一台 Xsens MTi-10 IMU 附在激光雷达上，以应对速度的快速变化。点云在送到前述方案前经过了两种方法的预处理：(1)利用 IMU 提供的方向，一个 sweep 中接收到的点云可旋转到与这个 sweep 中激光雷达的初始方向对齐；(2)利用加速度测量，运动失真可被部分地消除，等效于激光雷达在该 sweep 中以恒定速度移动。此处，IMU 方向由整合陀螺仪角速度和一个卡尔曼滤波器中的加速度计的读数得到。IMU 预处理之后，要解出(1)来自 IMU 的方向漂移，假设在一个 sweep 内是线性的；(2)线性的速度。由此，“激光雷达在一个 sweep 内的运动是线性的”的假设可以满足。点云随后送入里程计和建图程序。

图 13a 展示了一个简单的结果。一人持激光雷达在楼梯上行走。计算图中红色曲线时，使用 IMU 提供的方向，我们的前述方案仅估计平移。5 分钟数据收集期间，方向漂移了 25 度。绿色曲线假设无 IMU 可用，只依靠我们前述方案的优化。蓝色曲线预处理 IMU 数据，其后是我们的方案。绿色和蓝色曲线间的差异很小。图 13b 展示了对应蓝色曲线的地图。图 13c 中，我们比较两幅图 13b 地图中黄色方框内部分的近景，上图和下图分别对应蓝色和绿色曲线。仔细对比可发现，上图中边缘比下图更锋锐(sharper)。

- Fig.13　有/无 IMU 协助的结果比较。一人持激光雷达在楼梯上行走。黑点是起点。a 中，红色曲线由 IMU 得到的方向和我们方案估计的平移算出，绿色曲线只依赖我们方案的优化，蓝色曲线预处理 IMU 数据并在其后使用我们的方案。b 是蓝色曲线对应的地图。c 中，上图和下图分别对应蓝色和绿色曲线在图 b 中被黄色方框标记的区域，上图边缘更锋锐(清晰)，表明地图更精确。

表 3 比较了有无 IMU 情况下运动估计的相对误差。激光雷达被一个以 0.5m/s 速度步行的人手持，同时被此人以 0.5m 左右的幅度上下移动。真实参考由一个卷尺(tape ruler)测量得到。所有四个测试中，精确度最高的都是有 IMU 支持的我们的方案，最低的是只使用 IMU 提供方向的方法。结果表明，IMU 可以有效地消除非线性运动，我们的方案在 IMU 支持下只需处理线性运动。

- Table 3 有/无 IMU 时的运动估计误差

7.3 使用微型直升机数据集 Tests with micro-helicopter datasets

我们进一步使用一个八旋翼微型飞行器收集的数据评估我们的方案。如图 14 所示，该直升机装配了一个 2 轴激光雷达，设计和图 2 一样，但激光扫描器持续旋转。对这样一个激光雷达单元，一个 sweep 定

义为绕慢轴的一次半球旋转，持续一秒。直升机上还装配了一个 Microstrain 3DM-GX3-45 IMU。激光雷达和 IMU 数据都会交给里程计和建图程序处理。

- Fig.14　a 研究中用到的一个八旋翼直升机。一个 2 轴激光雷达装配在其前端，图 b 是其放大图。激光雷达基于一个 Hokuyo 激光扫描器，和图 2 中的设计相同，但此处激光扫描器持续旋转。

从两个数据集得到的结果分别如图 15 和图 16。两个测试中，直升机都由人工操纵以 1m/s 速度飞行。图 15 中，直升机从桥的一侧出发，穿过桥下，并再次经过桥下返回。图 16 中，直升机从地面起飞出发，降落在地面上结束。图 15a 和 16a 中，展示了飞行轨迹；图 15b 和 16b 中，展示了放大的对应图 15c 和 16c 黄色方框内区域的地图。我们未能获取直升机位姿和地图的真实参考。对于类似两个测试的小环境来说，我们的方案会持续地在测试开始时建立的地图上重定位，因此计算回环检测误差是无意义的(就是说该方案无回环检测？)。作为替代，我们只能使用视觉通过放大的视图来检验地图的准确性。

- Fig.15　从一座小桥得到的结果。a 显示直升机轨迹。黑点是起始位置。b,c 显示我们的方案建立的地图，其中 b 是 c 中黄色方框内区域的放大图。数据采集期间，直升机由人操控飞行。直升机从一侧出发，穿过桥下，转向，然后再次穿过桥下。
- Fig.16　从一座房屋前的一片区域得到的结果。a 显示直升机轨迹。黑点是起始位置。b,c 显示我们的方案建立的地图，其中 b 是 c 中黄色方框内区域的放大图。数据采集期间，直升机由人操控飞行。直升机从地面起飞出发，最后降落在地面上。

7.4 使用 Velodyne(威力登)激光雷达 Tests with Velodyne lidar

实验中所用装载了 Velodyne HDL-32E 激光雷达的两个载具如图 17。图 17a 是一辆多用途车，在人行道和越野地带行驶。图 17b 是一客运车辆，在大街上行驶。两辆载具的激光雷达都安装在车顶高处，以防车身可能造成的遮挡。

- Fig.17　两辆用于数据记录的搭载 Velodyne HDL-32E 的载具。

Velodyne HDL-32E 是一个单轴激光扫描器。它即时地向三维环境投射 32 道激光束。我们将每个激光束组成的平面视作一个扫描平面。一个 sweep 定义为激光扫描器的一次整圆旋转(full-circle rotation)。激光雷达获取扫描(scan)的默认频率是 10Hz。我们设置激光雷达里程计以 10Hz 运行以处理单独的扫描。建图把一秒的扫描堆积起来做批优化。两个程序的计算用时如表 4。

- Table 4　Velodyne HDL-32E 测试计算用时。(比前面用 Hokuyo 自组装的激光雷达用时更长)

图 18 展示了对大学校区的建图结果。数据由图 17a 中载具在 1.0km 长的行驶中记录。其间驾驶速度保持在 2-3m/s。图 18a 展示了最终地图。图 18b 在一张卫星图像上展示了估计的轨迹(红色曲线)和采集到的激光点(蓝色)。通过将轨迹和人行道匹配(载具并非在大街上行驶)，将激光点和建筑物墙匹配，我们确定水平位置漂移≤1m。通过对比两侧建图出的建筑物，我们确定垂直漂移≤1.5m。从而，全程总体位置误差在≤0.2%范围内。

图 19 显示了另一测试的结果。我们在大街上驾驶图 17b 所示载具行进 3.6km。除等待信号灯外，载具速度基本在 11-18m/s 之间。图 19 图片组织方式和图 18 相同。通过和卫星图片的对比，我们确定水平漂移≤2m。但我们未能评估垂直准确度。

- Fig.18　对大学校区的建图结果。
- Fig.19　对街道的建图结果。

7.5 使用 KITTI 数据集 Tests with KITTI datasets

最后，我们用 KITTI 里程计基准(benchmark)测试我们的方案。这个数据集是用装载在一辆公路行驶场景中的客运车辆顶部的传感器记录得到的。如图 20 所示，载具上装有彩色立体摄像机、单色立体摄像机、一台 Velodyne HDL-64E 激光扫描器，以及一个提供真实参考的高精度 GPS/INS。激光数据以 10Hz 记录，被用于运动估计方法。为达到最大可能精确度，扫描所得数据处理方法和 7.4 有略微不同。此处并不积累 10 个来自里程计的扫描(scan)，而是使建图运行频率和里程计相同，处理每个单独的扫描。这使得系统运行速度只有实时运行的 10%，每秒处理一个扫描。

- Fig.20　a 一辆 KITTI 基准用于数据记录的载具。我们的方案使用其中 Velodyne 激光雷达提供的数据。b 传感器放大图。图片来自 http://www.cvlibs.net/datasets/kitti/

这个数据集包含 11 个提供了 GPS/INS 真实参考的测试。其中最大行驶速度达到 85km/h(23.6m/s)。数据主要包含三种环境：城市(urban)，周围有建筑物；乡村(country)，在场景中有植被的小路上；高速公路(highway)，道路宽阔并且行驶速度很快。图 21 展示了三种环境的简单结果。在上方行，我们展示估计的载具轨迹，与 GPS/INS 真实参考对比。在中间和下方行，展示了每个数据集的地图和一张对应照片。地图以海拔高度编码上色。对 11 个数据集的完整测试结果如表 5。图 21 中的三个测试从左至右依次是表中的数据集 0，3 和 1。此处，精确度是基于三维坐标系，对在 100,200,…,800 米处的轨迹段的相对位置误差做平均算出的。

- Fig.21　使用 KITTI 基准数据集的样例结果。图中三个数据集从左至右分别取自城市、乡村、高速公路场景，对应表 5 中数字 0、3、1。在 a,b,c 中，比较了估计的轨迹和 GPS/INS 真实参考。黑点是起始位置。d,e,f 是对应 a,b,c 的地图，根据海拔高度编码上色。g,h,i 是描绘各环境的照片。
- Table 5　KITTI 基准数据集参数和结果。误差测算使用了轨迹在 100,200,…,800m 长度处的片段，基于三维坐标系，结果是“占片段长度的平均百分比”。

不考虑感知模态，我们结果准确度在 KITTI 里程计基准(www.cvlibs.net/datasets/kitti/eval_odometry.php)中排名#2(20220217 已经变#3 了)，其中和旅行距离相比的平均位置误差为 0.88%。结果优于前沿的基于视觉的方案(立体视觉里程计方案)，其中位置误差好 14%，方向误差好 24%。事实上排名#1 的方案也部分地使用了我们的方案：他们使用视觉里程计进行运动估计，使用我们提出的方案进行运动改进(refinement)。我们认为基于激光的状态估计优于基于视觉的方法，因为激光雷达对远点的测量能力。激光雷达的长度误差相对测量距离来说在相当程度上是常数，并且离载具很远的点确保了扫描匹配期间的方向精确性。

---

8 Discussion

建立一个复杂系统时，一个要考虑的问题是：系统中哪些组件是确保性能表现良好的关键？我们的第一个回答是双层数据处理结构。这使得我们可以把状态估计问题拆分成两个简单得多的问题。里程计只关注传感器的速度和运动失真的移除。里程计的速度估计并不精确(如图 12a)，但足以解包?(de-wrap)点云。此后，建图程序在精确扫描匹配时只需考虑固定的位姿变换(rigid body transform)。

我们对几何特征检测和匹配的实现是通向在线和实时处理的路径。具体到激光雷达里程计，匹配(matching)不需要非常精确，而对点云解包?(de-wrapping)而言更重要的是高频率。我们的实现以处理速度为最优先。然而，若计算资源充足，如 GPU 加速可用，则这种实现就不那么必要。

我们确实可以选择里程计和建图的频率比值。设为 10 是我们的偏好(建图正好比里程计慢一个数量级)。这意味着建图一次堆积 10 个来自里程计的扫描输出做批优化。把比值设的更高通常会造成更多漂移。同时，每当建图结束处理，会对运动估计产生一个跳变(jump)。这个比例可以使运动估计保持平滑。另一方面，把比值设的更低会导致消耗更多资源，且通常不必要。这个比值还平衡了两个 CPU 线程的计算负载。改变比值将造成一个线程的计算负载比另一个更大。

---

9 Conclusion and future work

使用来自一个旋转的激光扫描器的点云进行运动估计和建图是困难的，因为这个问题包含运动的复原和激光雷达点云运动失真的修正。我们提出的方案把问题拆分成两个，用两个并行运行的算法解决。两个算法的协作使得精确的运动估计和建图可以实时实现。方案已经过涵盖各种各样环境的大量实验测试使用了作者收集的数据以及 KITTI 里程计基准数据集。

(画的饼，20220217 似乎还没有实现)由于当前方案无法识别回环(无回环检测)，我们未来的工作包括开发一个利用回环修正运动估计漂移的方案。