DOI:10.19651/j.cnki.emt.2518240

基于深度学习的三维点云与 IMU 融合里程计*

张 乔 黄 瑞 张 裕 陈筱彦
(上海应用技术大学智能技术学部 上海 201418)

摘 要:里程计是即时定位与建图技术的重要组成部分,但是现有的里程计算法多数使用点云数据或图片数据等单一数据,没有充分利用多数据融合来提升轨迹估计的精度,同时在复杂环境和特征缺失的场景中轨迹估计的精度不足。针对此问题,本文提出了一种融合激光雷达数据和惯性测量单元数据的深度网络 3DPointLIO。首先结合特征金字塔网络和权重注意力机制来降低场景中动态信息的影响,提高点云特征的鲁棒性。然后在 IMU 数据处理网络中,通过卷积网络和门控循环单元相结合的方式来降低原始 IMU 数据中的噪声影响,并使用双向长短期记忆网络来提取降噪后的 IMU 数据的时序特征。最后通过多层线性层构成的位姿估计网络进行平移和旋转的估计。在开源数据集 KITTI 上进行验证,实验结果表明,该里程计算法相比于基准模型在旋转的估计上提升了 0.76°,平移的估计上提升了 2.17%。与其他常见的里程计算法相比,旋转和平移的估计也表现出较好的效果,特别是在旋转的估计上具有更高的精度。

关键词:即时定位与建图;激光惯导融合里程计;激光雷达;深度学习
中图分类号:TN242;TN958.98 **文献标识码**:A **国家标准学科分类代码**:510.99

Deep learning-based 3D point cloud and IMU fusion odometry

Zhang Qiao Huang Rui Zhang Yu Chen Xiaoyan
(School of Computer Science and Information Engineering, Shanghai Institute of Technology, Shanghai 201418, China)

Abstract: Odometry is a crucial component of Simultaneous Localization and Mapping (SLAM) technology. However, most existing odometry algorithms rely on single data sources such as point cloud data or image data, failing to fully leverage multi-data fusion to improve trajectory estimation accuracy. Additionally, these algorithms often exhibit insufficient accuracy in complex environments and feature-deficient scenarios. To address these issues, this paper proposes a deep network called 3DPointLIO, which fuses LiDAR data and Inertial Measurement Unit (IMU) data. Firstly, a feature pyramid network combined with a weight attention mechanism is introduced to reduce the impact of dynamic information in the environment and enhance the robustness of point cloud features. Secondly, in the IMU data processing network, a convolutional network is integrated with Gated Recurrent Units (GRUs) to mitigate noise in raw IMU data, and a Bidirectional Long Short-Term Memory (BiLSTM) network is employed to extract temporal features from the denoised IMU data. Finally, a pose estimation network composed of multiple linear layers is used to estimate translation and rotation. The proposed algorithm is validated on the open-source KITTI dataset. Experimental results demonstrate that, compared to the baseline model, the proposed odometry algorithm improves rotation estimation by 0.76° and translation estimation by 2.17%. Furthermore, it outperforms other common odometry algorithms in both rotation and translation estimation, particularly achieving higher accuracy in rotation estimation.

Keywords: simultaneous localization and mapping; LiDAR-IMU fusion odometry; LiDAR; deep learning

0 引 言

同时定位与建图技术(simultaneous localization and mapping, SLAM)是自主移动机器人和自动驾驶的关键技术之一,是一项能够提供精确状态估计和地图构建的关键技术,主要用于移动机器人在移动中时,利用各种类型的传感器采集到的环境数据,实现对当前所处位置以及姿态的估计,同时建立高精度的环境地图。而里程估计(odometry

收稿日期:2025-02-03
* 基金项目:上海市自然科学基金(21ZR1462600)项目资助

estimation),即通过时间和空间估计机器人在三维空间中的位置和姿态,是 SLAM 技术中的关键一环,其使用的感知信息包括摄像头、激光雷达(Lidar)和惯性测量单元(inertial measurement unit, IMU)等传感器采集到的数据。因此对里程计的研究有助于更深入的理解机器人运动的估计方法,还能够推动自动驾驶和移动机器人技术的进一步发展。

传统里程计算法按照所使用的传感器的不同,可以将其分单传感器和多传感器融合两大类。在单传感器方面,激光雷达里程计和视觉里程计是最为常见和广泛应用的技术。提取点云的点、线、面特征匹配是最常见的传统激光里程计的点云配准算法[1],通过迭代的方式最小化两组点云之间的误差,直到收敛。这种类型的算法对初始估计较为敏感,非常容易受到动态信息的干扰,对于复杂场景适应性不足。后来 LOAM[2]、Lego-LOAM[3] 等工作被提出,通过提取点云的边缘特征和平面特征来优化里程计的效果,但是仍然非常容易受到动态信息的影响,且在处理大规模场景时性能急剧下降。传统的激光里程计算法由于点云数据本身存在的噪声,以及动态物体和遮挡情况的存在,造成算法很容易产生错误的点云配对[4],除此之外,点云的稀疏性和非结构化性质使得传统激光里程计算法更难以有更好的效果。因此研究人员开始将激光点云和 IMU 结合起来,LIO-SAM[5]、Fast-LIO[6] 等工作利用了高频率的 IMU 数据来增强激光雷达低频率的位姿估计,但是 IMU 数据本身的不确定性非常容易影响其位姿估计的精度。

由于传统里程计算法非常容易受到数据本身的噪声和环境特征缺失的影响,而深度学习的技术在图像领域表现令人瞩目,因此研究人员开始将深度学习技术应用到里程计算法中,Nicolai 等[7]、Velas 等[8] 提出使用卷积网络作为特征提取网络,点云投影作为输入的激光里程计模型,但简单的网络结构导致点云特征提取不够鲁棒,位姿估计不够理想。后来 Li 等[9] 提出 LO-Net,Cho 等[10] 提出 DeepLO,利用残差网络来扩展特征提取网络,虽然位姿估计效果有所提升,但是对于长距离以及复杂场景下的位姿估计存在较大的漂移误差。因此为了提高激光里程计算法的位姿估计精度,研究人员开始将 IMU 特征和点云特征进行融合。DeepLIO[11]、TransFusionOdom[12] 以及李隆等[13] 提出的 MLVIO-Net 等模型通过融合点云和 IMU 数据来获取较好的位姿估计效果,但是在融合两者时忽略了点云特征的鲁棒性和 IMU 数据本身的噪声影响,同时使得网络结构更为复杂。为了关注点云特征本身的鲁棒性,Liu 等[14] 和 Ali 等[15] 分别提出了 TransLO 和 DELO,分别利用分层的特征提取和复杂的交叉注意力机制取得了较好的位姿估计效果,但是同样忽略了 IMU 数据的作用。而李一凡等[16] 则利用点云分割来降低动态点的影响,从提高点云的鲁棒性,但是容易产生误分割从而造成误匹配。

在视觉里程计领域,深度学习得到了更为深入的应用,Min 等[17] 提出的 Voldor 和 Yang 等[18] 提出的 D3VO,以及纪泽源等[19] 提出的 VINS-GNN 和张震宇等[20] 提出的 VFE-VO,都利用神经网络来增强视觉特征的鲁棒性,但是他们都忽略了动态点的影响,因此黄友锐等[21] 和史涛等[22] 分别提出了利用 YOLO 系列的算法来分割出动态点,从而提高位姿估计的效果。

尽管深度学习在视觉里程计领域取得了显著的进展,但是在激光里程计领域,深度学习的应用仍然是一个具有挑战性的问题。上述的研究虽然在里程计算法方面取得了一定的效果,但在位姿估计精度上仍有较大的提升空间,尤其是在特征稀疏或环境复杂的场景中表现欠佳,且忽略了 IMU 数据噪声带来的负面影响。针对这些问题,本文提出了一种特征金字塔网络,结合位置编码和注意力机制对点云数据进行鲁棒的特征提取。设计了一种 IMU 数据处理网络,通过卷积操作和门控循环单元(gated recurrent unit, GRU)的结合来降低 IMU 数据的噪声,增强 IMU 特征的表征能力,并使用多层的双向长短期记忆网络 Bi-LSTM[23] 来提取 IMU 数据特征。通过全连接层构成位姿估计网络。最终基于上述网络构建了一个 3DPointLIO 里程计网络,在权威 KITTI 自动驾驶数据集上的实验表明,3DPointLIO 通过简单的网络结构,在旋转和平移的估计上取得了优于基准模型 DeepLIO 的效果,且相比于其它常见算法,在旋转的估计上取得了更高的精度。

1　网络模型设计

1.1　网络模型

如图 1 所示,本文所提出的三维点云和 IMU 融合里程计的网络模型由以下及部分组成:基于特征金字塔的点云特征提取网络、IMU 数据的降噪网络、基于 Bi-LSTM 的 IMU 特征提取网络和位姿估计网络。里程计模型的输入为相邻两帧点云以及两帧点云之间的 IMU 数据,输出为旋转四元数 $q=[a,b,c,d]$ 和平移向量 $\omega=[\omega_x,\omega_y,\omega_z]$。具体来说就是将降采样后的三维点云坐标作为 $P_{t,0}$ 和 $P_{t+1,0}$,对点云进行预处理后的向量作为两帧点云的初始特征 $F_{t,0}$ 和 $F_{t+1,0}$。然后利用特征金字塔网络分别提取两帧点云从稠密到稀疏的特征信息 $F_{t,l}$ 和 $F_{t+1,l}(1\leqslant l\leqslant 4)$,$l$ 表示特征金字塔网络的层数。将两帧点云经过网络处理后得到的全局特征拼接得到特征 F_{cat}。同时,按照两帧点云的时间戳获取两帧点云之间时刻的 IMU 数据,并将其输入到构建的基于 CNN-GRU 的降噪网络,分别对角速度测量值 ω 和加速度测量值 a 进行修正,然后将修正后的 IMU 数据送到搭建的 Bi-LSTM 网络分别处理角速度和加速度值,得到 IMU 的时序特征 $F_{t,IMU}$。将点云的全局特征 F_{cat} 和 IMU 的时序特征 $F_{t,IMU}$ 进行加权融合后经由两个位姿估计网络分别得到两帧点云之间的位姿变换信息:旋转四元数 q 和平移向量 ω。

1.2　点云数据预处理

由于室外的自动驾驶场景中，每一帧的点云数据的都比较庞大，在 KITTI 数据集中，每一帧点云拥有十几万个点，如果直接对原始点云进行处理，那么会耗费庞大的计算资源和内存资源。因此本文在将一帧点云 P 作为输入之前，分别去除掉点的 x 轴方向和 y 轴方向 30 m 之外的点，得到点云 P'，然后对 P' 进行随机采样，采样 8 192 个点作为输入点云，也记为 P。之所以采样随机采样进行降采样，是因为在测试中，使用体素降采样和使用最远均匀降采样会增加处理数据所需要的时间，而随机采样不仅不会降低点云特征提取的效果，同时拥有最快的采样速度，所以最终选择随机采样来对点云进行降采样。

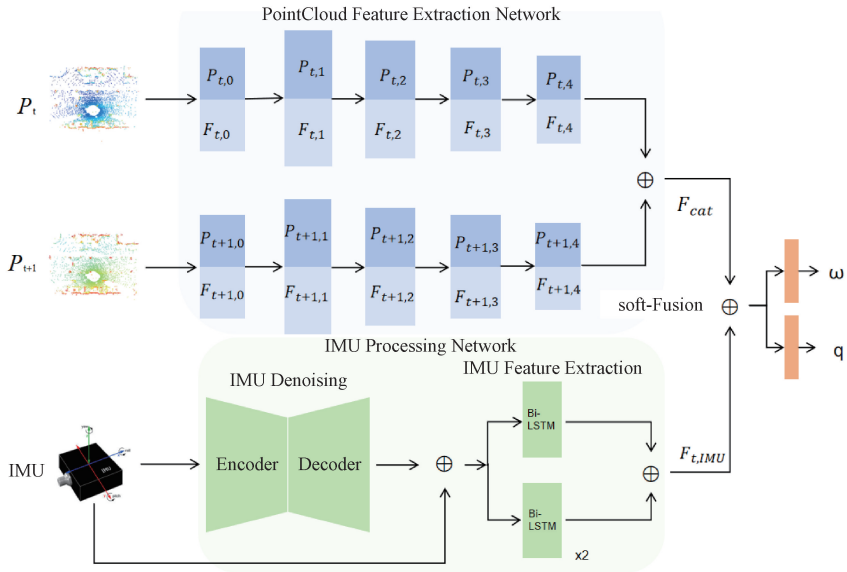


图 1　3DPointLIO 网络结构

Fig. 1　Network structure of 3DPointLIO

对点云进行距离划分和随机采样后，得到最终使用的点云 Points，每帧点云包含有 8 192 个点。具体表达式如下：

$$P' = \{P[i] \mid -30 < x_i < 30, -30 < y_i < 30\} \quad (1)$$

$$Points = random(P') \quad (2)$$

然后对降采样后的点云 Points 进行去中心化和归一化处理。对点云进行去中心化和归一化处理有助于降低传感器在不同位置采集数据时所带来的大尺度差异，让算法更专注于点云的几何特征，并且提高了算法的通用性和适应性。去中心化和归一化处理数学表达式如下：

$$P_c[i] = Points[i] - \frac{1}{N}\sum_{i=1}^{N} Points[i] \quad (3)$$

$$P_n[i] = \frac{P_c[i]}{d + \in} \quad (4)$$

式中：P_c 为得到的去中心化后的点云，P_n 为归一化后的点云，d 为点云中每个点到原点的欧氏距离的最大值，将所有按照最大距离进行归一化。$\in$ 为一个小常数，防止出现除零错误，通常设置为 1×10^{-8}。

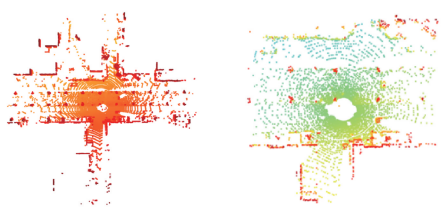
上述经过去中心化和归一化处理后的一帧点云记为 $P_{t,0}$，然后分别计算点云中每个点相对该帧点云中高度最低的点的相对高度 z_{rel}，然后将该相对高度和点的坐标以及该帧的点云质心坐标 $[c_x, c_y, c_z]$ 拼接，得到每个点的初始特征 $F_{t,0}$。表达式如下：

$$z_{rel}[i] = z_i - \min(z_1, z_2, \cdots, z_N) \quad (5)$$

$$F_{t,0} = [x_i, y_i, z_{rel,i}, c_x, c_y, c_z] (0 \leqslant i \leqslant 8\,192) \quad (6)$$

通过将点的相对高度以及点云的质心作为点的初始特征，能够利用高度差来减少地面点云和动态物体带来的部分影响，因为相对高度可以一定程度上将地面和非地面点云区分开，同时区分出由于移动造成相对高度发生变化的移动物体。而利用质心则一定程度上可以减小全局误差，帮助校正帧间位移误差，从而提高位姿估计的精度。

图 2 展示了点云降采样之前和降采样之后的情形，可以看到，降采样之前的点云更为密集，范围大且数量多，对于计算的要求非常大，降采样之后可以看到点云变得稀疏一些，但是一些明显的特征仍然保留了下来，降低了计算负担。



(a) 降采样前　(b) 降采样后

(a) Before downsampling　(b) After downsampling

图 2　点云降采样前后对比

Fig. 2　Comparision befor and after Point Cloud Downsampling

1.3　点云特征提取网络

虽然在点云数据预处理阶段对大规模的点云进行降采样得到了 8 192 个点,但是如果对 8 192 个点进行逐点特征提取,那么仍然会造成计算资源和时间的大量损耗,且直接通过逐点特征来获取点云的全局特征,动态点或点云噪声对特征的鲁棒性影响较大,因此采用特征金字塔的形式来逐层提取点云的特征,通过多分辨率的特征聚合的方法,来降低动态点和噪点的影响。通过最远点均匀采样(FPS)来获取每一层用于特征提取的点。

如图 3 所示,该特征金字塔网络包含两个分支,分别用于提取相邻的两帧点云的特征,二者参数共享。以其中一个网络分支为例,对一帧点云数据进行了预处理后,得到了点云数据 P_t 以及每个点的初始特征 $F'_{t,0}$,然后使用一个全连接层对每个点的初始特征 进行点编码,得到特征金字塔网络的输入特征 $F_{t,1}$,表达式如下:

$$F_{t,1} = GELU(BatchNorm(Conv1d(F'_{t,0}))) \tag{7}$$

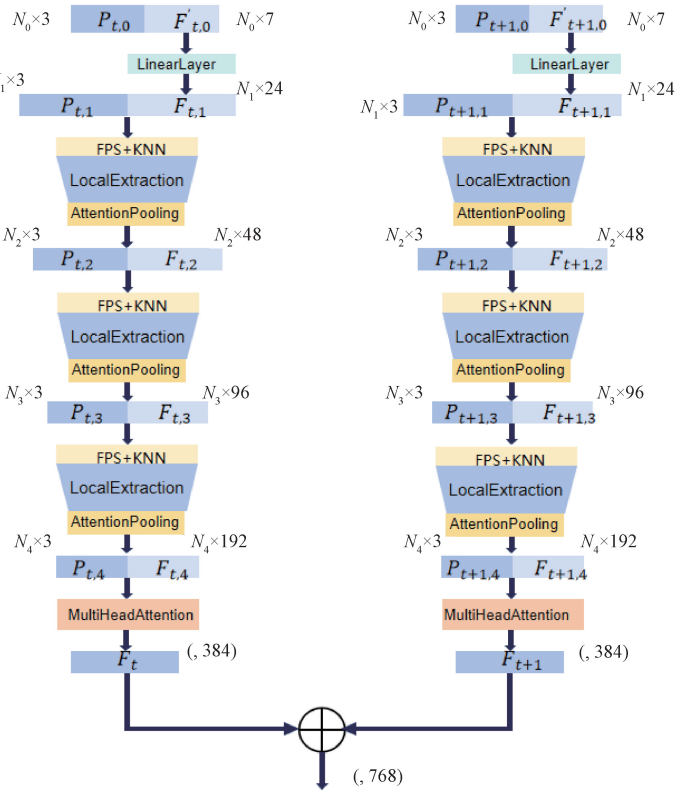


图 3　点云特征金字塔网络结构

Fig. 3　Point cloud feature pyramid network structure

使用 FPS 算法对输入点云进行降采样,并使用 K 近邻算法(KNN)找到降采样后每个点在降采样前点云中的 k 个邻域点。然后使用局部特征提取模块和基于权重的注意力池化来对邻域信息进行编码和聚合。在局部特征提取模块中利用三角函数来对 k 个邻域点进行位置编码,将编码后的特征 p 和点初始特征 $F_{t,1}$ 进行特征加权后,利用两个线性变换层 f_l 进行特征扩展,得到每个邻域点的特征 F_i。表达式如下:

$$p = \sin\left(\frac{\beta \cdot P_{i,xyz}}{\alpha^{\frac{i}{d}}}\right) \oplus \cos\left(\frac{\beta \cdot P_{i,xyz}}{\alpha^{\frac{i}{d}}}\right), 0 \leqslant i \leqslant k \tag{8}$$

$$F_i = f_l(f_l((F_{t,1} + p) \cdot p)), 0 \leqslant i \leqslant k \tag{9}$$

式中:α,β 是控制位置编码的超参数,i 表示邻域 k 的索引。

得到每个点的邻域点的特征 F_i 后,利用基于权重的注意力池化操作,得到降采样后每个点的特征 $F_{t,j}$。表达式如下:

$$a = Conv1d(x_f)(x_f \in R^{(B\times G\times D\times K)}) \tag{10}$$

$$w = soft\max(a, \dim = -1) \tag{11}$$

$$F_{t,j} = \sum_{k=1}^{K} x_f(:,:,k) \cdot w(:,k) \tag{12}$$

式中:x_f 表示 F_i 展平后的向量,B 表示批次大小,G 表示每层的采样点的数量,D 表示每个点的特征维度,K 表示每个点的邻域点数目,a,w 表示注意力权重。

对点云重复上述操作,得到一个 4 层的特征金字塔网络,在最后一层特征聚合之后,利用多头注意力机制对点云特征再次进行处理,得到最终的帧点云全局特征 F_t。对相邻点云 P_{t+1} 也是用相同的网络对其进行处理,最终得到一个孪生特征金字塔网络。在逐层的特征提取过程,随着点云数量的减少,动态信息的影响也逐渐降低,最终得到

一个较为鲁棒的全局特征，为后续的位姿估计提供了一个较好的支持。

1.4　IMU 数据处理网络

由于本文使用的 KITTI 数据集中，所采用的 IMU 数据的频率是 100 Hz，而激光雷达的频率是 10 Hz，因此首先需要将 IMU 数据和点云数据按时间进行同步。即取一帧点云数据的起始和结束时间作为两个时间戳，然后将这段时间内的数个 IMU 数据作为一帧数据与该帧点云数据相对应。

因为原始的 IMU 测量数据由于惯性测量单元的长时间执行，存在较大的累积误差，这对于提取 IMU 测量数据的特征造成了非常大的影响，为了解决这个问题，本文设计了一种由 CNN-GRU 构成的 IMU 去噪网络，来降低 IMU 测量数据中噪声的影响。如图 1 所示，该去噪网络由一个 Encoder 模块和一个 Decoder 模块组成。其中 Encoder 的组成如图 4 所示，将一帧的 IMU 数据包含的 L(L=10)个加速度测量值 a_x，a_y，a_z 和角速度测量值 ω_x，ω_y，ω_z 分别送入到具有相同结构的特征提取通道中，得到加速度测量数据的高纬特征 $F_{i,a}(i \leqslant L)$ 和角速度测量数据的高纬特征 $F_{i,\omega}$。具体表达式如下：

$$[F_{i,a}^{j}, F_{i,\omega}^{j}] = GRU(CNN(F_{i,a}^{j-1}, F_{i,\omega}^{j-1})) \tag{13}$$

式中：$i(0 \leqslant i \leqslant L)$ 表示一帧 IMU 数据的索引，$j(1 \leqslant j \leqslant 3)$ 表示上述公式计算次数。经过 3 次处理后，将加速度和角速度测量数据从 3 维映射到 128 维。

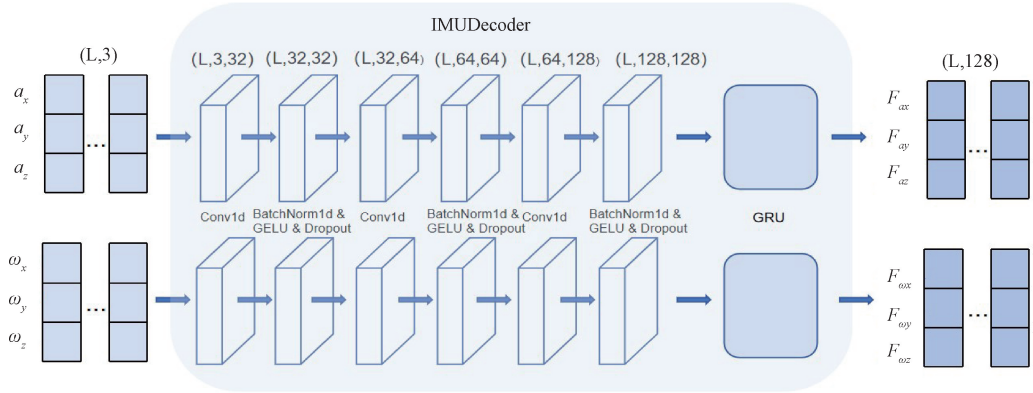


图 4　IMUEncoder 网络结构

Fig. 4　IMUEncoder network structure

得到加速度和角速度的特征 $F_{L,a}$，$F_{L,\omega}$ 后，将其输入到 IMUDecoder 模块中进行解码，得到加速度和角速度测量数据的修正量。如图 5 所示，经过编码器得到的高维特征 $F_{L,a}$ 和 $F_{L,\omega}$ 同样分别输入到具有相同结构的解码通道中，最终得到维度和原始 IMU 输入数据相同的 IMU 测量数据的修正量 $\Delta a_{x,y,z}$ 和 $\Delta\omega_{x,y,z}$：

$$[\Delta a, \Delta\omega] = f(F_{L,a}, F_{L,\omega}) \tag{14}$$

式中：f 表示解码操作。得到每条 IMU 数据的修正量后，即可对原始的 IMU 测量数据进行修正，从而得到最终输入到基于 Bi-LSTM 网络的 IMU 特征提取网络的 IMU 数据 (a, ω)：

$$[a, \omega] = [a, \omega] + [\Delta a, \Delta\omega] \tag{15}$$

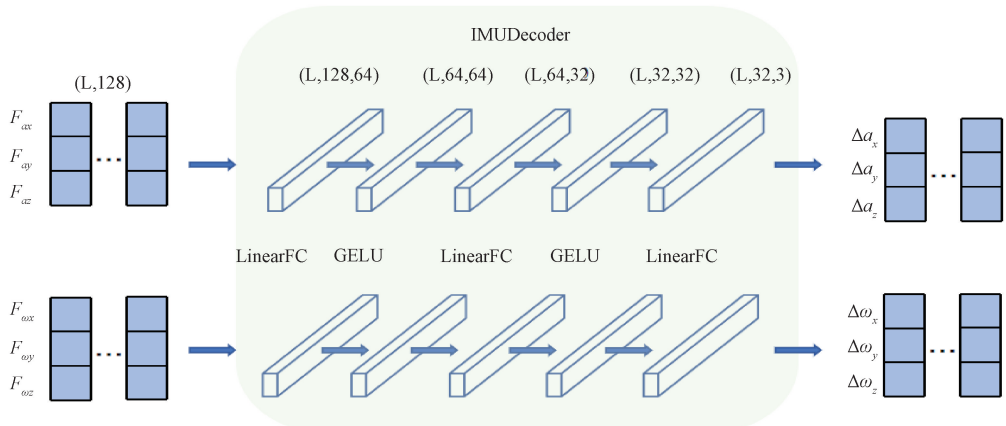


图 5　IMUDecoder 网络结构

Fig. 5　IMUDecoder network structure

由于 IMU 数据包含加速度数据和陀螺仪数据，这两者观测值的类型不同，同时分布具有较大的差异，因此如

果将这两种数据拼接到一起输入到网络中学习往往效果不理想。为解决这个问题,如图 6 所示,本文使用两个独立的网络模块分别学习经过去噪网络处理后的加速度测量值 a 和角速度测量值 ω 的时序特征,每个模块由两个 Bi-LSTM 网络构成,通过 Bi-LSTM 的双向结构,增强网络对 IMU 数据前向依赖和反向依赖的捕捉,提高模型对 IMU 的时序特征提取的鲁棒性。最后将分别提取到的加速度测量值的特征和角速度测量值的特征进行融合,得到最终的 IMU 特征 F_{imu}。

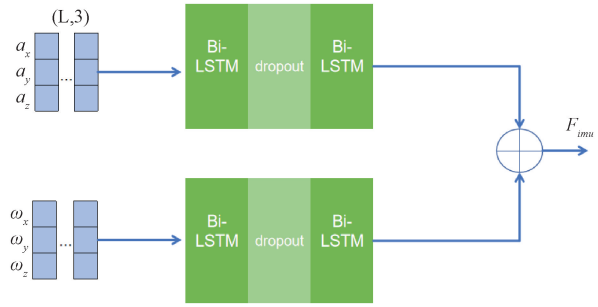


图 6　IMU 特征提取网络

Fig. 6　IMU feature extraction network

1.5　位姿估计网络

通过特征金字塔网络和 IMU 数据处理网络得到了点云特征和 IMU 特征后,对这两个特征进行软融合,即加权融合得到融合特征后,将其输入到有多层全连接层堆叠而成的位姿估计网络中,分别估计出旋转和平移的估计量。数学表达式如下:

$$C = Concat(L, I) \tag{16}$$

$$s_1 = \sigma(W_1 C + b_1) \tag{17}$$

$$s_2 = \sigma(W_2 C + b_2) \tag{18}$$

$$[L', I'] = [L \odot s_1, I \odot s_2] \tag{19}$$

$$[q, \omega] = \sigma(WX + b) \tag{20}$$

式中:C 表示将点云特征 L 和 IMU 特征 I 进行拼接得到的特征。W、W_1、W_2 表示权重矩阵,b、b_1、b_2 为偏置项,s_1、s_2 表示点云项和 IMU 项的权重,σ 表示激活函数,L'、I' 表示加权后的特征,X 表示拼接后的特征 $[L', I']$,$[q, \omega]$ 为最终得到的旋转和平移的估计。

1.6　损失函数

在本研究中,采用了一种基于同方差不确定性(homoscedastic uncertainty)的加权损失函数,通过计算局部帧和全局帧的损失,同时优化平移和旋转任务的预测结果。具体而言,损失函数由两部分组成:平移损失 L_x 和旋转损失 L_r。其中,平移损失通过均方误差(mean squared error, MSE)计算预测值和真实值之间的差异,包括局部帧间和局部帧到全局的平移误差。旋转损失同样也是如此。计算方式如下:

$$L_x = MSE(\hat{t}_{f2f}, t_{f2f}) + MSE(\hat{t}_{f2g}, t_{f2g}) \tag{21}$$

$$L_r = MSE(\hat{q}_{f2f}, q_{f2f}) + MSE(\hat{q}_{f2g}, q_{f2g}) \tag{22}$$

式中:下标 $f2f$ 表示局部帧间,即相邻帧之间。下标 $f2g$ 表示局部帧到全局之间,即每一组帧序列中,每一帧和当前帧序列中的第一帧之间的变换。

通过结合局部帧间损失和局部帧到全局损失的损失函数设计,可以抑制局部预测的误差由于时间累积带来的累积误差,提升全局轨迹的一致性。最后利用同方差不确定性加权的思想来结合平移损失和旋转损失,得到最终的损失函数:

$$\mathcal{L} = L_x \cdot \exp(-s_x) + s_x + L_r \cdot \exp(-s_q) + s_q \tag{23}$$

式中:由于平移和旋转存在单位和尺度上的差异,因此为了平衡平移和旋转这两个任务之间的权重,引入了可学习的不确定性参数 s_x 和 s_q。$\exp(-s_x)$ 和 $\exp(-s_q)$ 则分别用于缩放平移和旋转损失的权重,而 s_x 和 s_q 作为正则项,确保不确定性参数不会过度增长。这种设计不仅能够自适应的平衡多任务学习的权重,还能在训练过程中优化不确定性参数,从而提高模型的鲁棒性和泛化能力。

2　实验与结果分析

2.1　实验环境

为验证提出的激光惯导里程计算法的轨迹预测效果,实验所用的硬件环境包括 Inter(R) Xeon(R) W-1270P CPU @3.80 GHZ、64 GB 内存和 NVIDIA GeForce RTX 3090(24 G)显卡。软件环境为 Ubuntu 20.04 操作系统,深度学习框架为 Pytorch 1.12.0+cu116。实验参数设置如表 1 所示。

表 1　实验参数设置

Table 1　Experimental parameter settings

参数	数值
输入点云尺寸	6×8 192×3
输入 IMU 尺寸	6×10×6
最大 epoch	100
训练批量大小	2
学习率	0.000 1
优化器	Adam

2.2　实验数据

KITTI 数据集是自动驾驶和计算机视觉领域最具影响力的公开基准数据集之一,由德国卡尔斯鲁厄理工学院(Karlsruhe Institute of Technology, KIT)和丰田美国技术研究院(Toyota Technological Institute at Chicago, TTIC)于 2012 年联合发布。其里程计数据子集专为视觉里程计(visual odometry, VO)、激光里程计(lidar odometry, LO)和同步定位与建图研究设计,包含有 22 个连续的驾驶序列,其中前 11 个序列数据提供了真值轨迹,场景覆盖了城市道路、乡村、高速公路等多种环境,包含动态物体(车辆、

行人)复杂光照变化以及大规模场景。所有数据由搭载了双目灰度相机、64 线激光雷达和高精度惯导系统的移动车辆所测量。

在本实验中,在 KITTI 里程计数据集上对本文提出的融合里程计进行训练和测试,选取包含真实轨迹数据的序列 00～10,并将 00～06 划分为训练集,07～10 划分为测试集。

2.3　评估指标

KITTI 官方主要采用两种指标来评估里程计性能,一种是绝对轨迹误差(absolute trajectory error, ATE),通过最小二乘匹配预测轨迹与真值轨迹后计算均方根误差(root mean square error, RMSE),衡量全局一致性。另一种是相对姿态误差(relative pose error, RPE),即在固定间隔(如 100 m 或 200 帧)内计算相对位姿变化的旋转和平移误差,反应局部运动的估计精度。由于在里程计相关的研究工作中都是使用 RPE 来作为评估指标,因此本实验也采用第二种 RPE 作为算法的评估指标。主要测量对比算法预测轨迹与真值在 100、200、…、800 m 的平移和旋转的平均 RMSE。数学表达式如下:

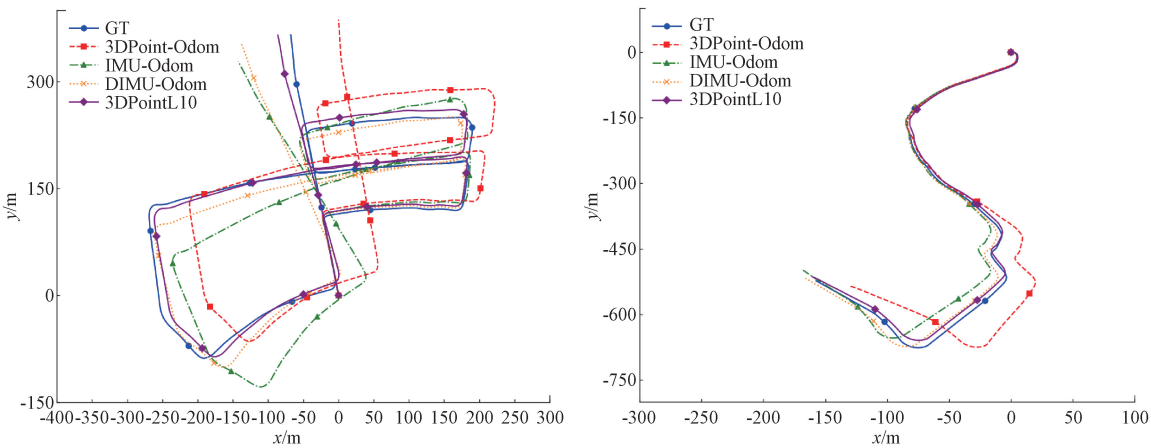
$$RPE_{q,\omega} = \| T_{i+1}^{-1} T_i - G_{i+1}^{-1} G_i \| \tag{24}$$

$$RPE_{T,rmse} = \frac{1}{K}\sum_{k=1}^{K}\sqrt{\frac{1}{N}\sum_{i}^{N} RPE_{(q,\omega),k}(i)^2} \tag{25}$$

式中:$RPE_{q,\omega}$ 表示两帧位姿之间的相对误差,T 和 G 分别表示估计位姿和真值。$RPE_{T,rmse}$ 表示 $RPE_{q,\omega}$ 在 100、200、…、800 m 间隔上的平均 RMSE 误差。K 表示固定间隔的数量,N 表示每个固定间隔内的帧数。

2.4　消融实验

为了评估点云与 IMU 融合里程计中的各个模块的效果,分别测试了仅使用三维点云、仅使用原始 IMU 数据、仅使用去噪后的 IMU 数据和三维点云融合去噪后 IMU 数据这 4 种情况下里程计的表现与真实轨迹的对比情况。表 2 展示了这 4 种情况下的里程计算法在训练数据序列 05 和测试数据序列 10 中对平移和旋转估计的具体评估指标。3DPoint-Odom 表示仅使用三维点云数据的里程计,IMU-Odom 表示仅使用原始 IMU 数据的里程计,DIMU-Odom 表示仅使用去噪后的 IMU 数据的里程计。图 7 则可视化了上述四种里程计在训练数据序列 05 和测试数据序列 10 上的预测轨迹与真实轨迹 GT 的对比效果。



(a) 里程计在序列05上的轨迹
(a) Odometry trajectory on sequence 05

(b) 里程计在序列10上的轨迹
(b) Odometry trajectory on sequence 10

图 7　里程计算法各模块效果对比

Fig. 7　Performance comparision of modules in odometry algorithm

表 2　序列 05 和序列 10 上的消融实验结果

Table 2　Ablation study results on sequence 05 and sequence 10

Type	IMU-Odom		DIMU-Odom		3DPoint-Odom		3DPointLIO	
Seq/Unit	t_{rel} (%)	r_{rel} (°)	t_{rel} (%)	r_{rel} (°)	t_{rel} (%)	r_{rel} (°)	t_{rel} (%)	r_{rel} (°)
05	5.07	1.55	2.40	0.75	5.53	0.73	1.71	0.49
10	3.52	0.66	1.23	0.54	4.00	1.16	1.22	0.27

从表 2 中可以看到,使用去噪模块对 IMU 数据进行去噪后,在序列 05 中,平移误差降低了 2.67%,旋转误差降低了 0.8°。在序列 10 中平移误差降低了 2.29%,旋转误差降低了 0.12°。仅使用三维点云的里程计在序列 05 中平移误差为 5.53%,旋转误差为 0.73°,结合 IMU 数据后,平移误差降低了 3.82%,旋转误差降低了 0.24°,在序列 10 中,平移和旋转误差也得到了一定的下降。结合图 7(a)和图 7(b)的轨迹可视化结果,可以更为明显的看到无论是在序列 05 还是序列 10 上,随着对 IMU 数据进行去噪后,IMU 里程计的预测效果得到了较大的提升,将 IMU

特征和点云特征结合后,轨迹进一步的向真值轨迹靠拢。由此说明提出的特征金字塔网络和 IMU 去噪网络以及结合二者的网络结构设计是合理的。

2.5　对比实验

表 3 展示了本文提出的 3DPointLIO 与目前主流的里程计算法 LIO-SAM、DeepLIO、Delo、TransLO 以及 TransFusionOdom 在序列 00～10 上轨迹预测的误差,采用 100、200、…、800 m 的 RMSE 作为评估指标。其中"/"表示原文数据缺失,"*"表示仅使用算法中的前端里程计进行测试,"_"表示实验结果处于第二,加粗表示处于第一。

表 3　各里程计算法轨迹预测误差对比

Table 3　Comparision of trajectory prediction errors among odometry algorithms

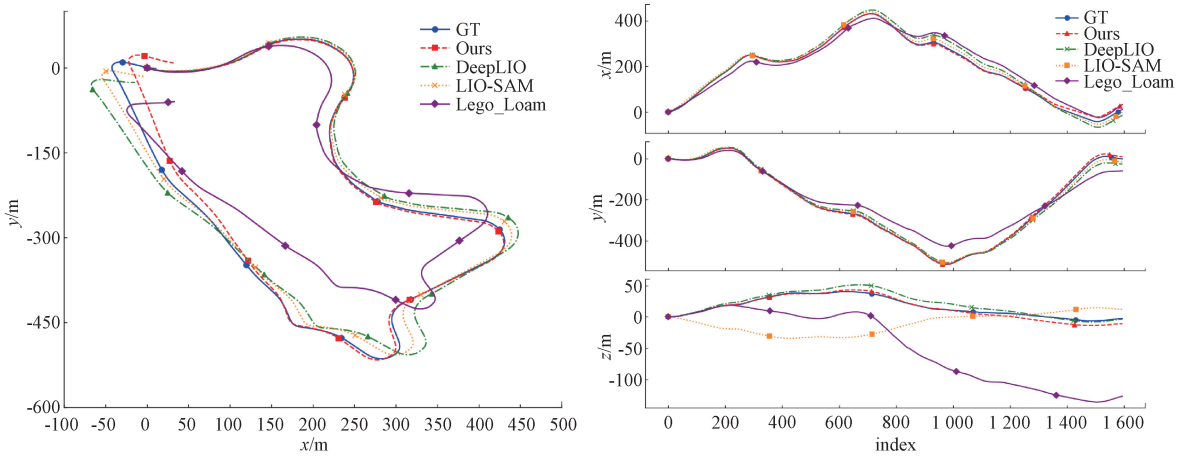
Type	LIO-SAM*[5]		DeepLIO[11]		Delo[15]		TransLO[14]		TransFusionOdom[12]		Ours	
Seq/Unit	t_{rel} (%)	r_{rel} (°)	t_{rel} (%)	r_{rel} (°)	t_{rel} (%)	r_{rel} (°)	t_{rel} (%)	r_{rel} (°)	t_{rel} (%)	r_{rel} (°)	t_{rel} (%)	r_{rel} (°)
00	/	/	1.6	**0.38**	2.97	1.30	**0.89**	0.41	/	/	2.53	0.62
01	1.23	0.67	5.28	1.51	11.99	2.19	1.16	0.45	**0.43**	0.46	1.38	**0.32**
02	/	/	1.96	**0.23**	4.88	1.71	**0.81**	0.46	/	/	1.51	0.51
04	/	/	3.7	**0.12**	2.42	7.42	**0.78**	0.67	/	/	1.66	0.52
05	/	/	1.24	0.93	2.17	1.00	0.73	0.55	**0.48**	0.73	1.71	**0.49**
06	/	/	1.97	0.82	2.58	1.01	**0.74**	0.56	/	/	1.58	**0.41**
07	/	/	1.92	1.33	1.97	1.44	**0.90**	0.81	/	/	2.21	**0.78**
08	3.88	1.67	2.34	0.94	9.02	3.48	1.29	**0.50**	**0.99**	0.75	2.31	0.82
09	1.28	0.83	4.4	1.21	2.26	1.54	0.95	0.46	**0.49**	0.63	0.72	**0.3**
10	1.34	0.78	4.0	1.51	3.54	2.16	1.18	0.61	**0.72**	0.78	1.22	**0.27**
Avg	2.16	1.09	3.58	1.22	4.94	2.39	1.14	0.52	**0.73**	0.72	1.41	**0.46**

Avg:测试集序列 08、09、10 的平移和旋转的 RMSE 均值

从表 3 中的算法在所有序列中的平移和旋转均方根误差的值可以发现,本文提出的里程计算法总体上对于旋转的估计有着较好的效果,在 01、05、06、07、09 和 10 的序列数据上取得了最好的效果,对于平移的估计虽然在多数序列上没有取到最好的效果,但是在序列 02、04、06、09 上也表现出了仅次于最优的效果。通过测试集的 3 个序列的均值可以看到,本文提出的 3DPointLIO 在旋转的估计上取得了最好的效果 0.46°,在平移的估计上虽然没有取得最好的效果,但也取得了 1.41%的效果,在所有比较的算法中位于第三。和目前主流算法 TransLO 相比,3DPointLO 总体上在旋转的估计上表现出了更好的效果。与基准模型 DeepLIO 相比,3DPointLIO 的平均平移误差降低了 2.17%,平均旋转误差降低了 0.76°,与具有更为复杂的网络结构的 TransFusionOdom 相比,平均平移误差表现较差,但是平均旋转误差表现最好,降低了 0.26°。总体上来说,得益于融合了鲁棒的 IMU 数据和点云数据,本文的里程计算法相比于纯点云数据的里程计和传统的 LIO-SAM 算法以及基准模型,在旋转估计上表现更好,平移估计也有不错的效果。

图 8 展示了本文提出的 3DPointLIO 与复现的当前常见算法 DeepLIO、LIO-SAM 和 Lego_LOAM 在序列 09 的预测轨迹及其在 x 轴、y 轴和 z 轴上的预测与真实轨迹 GT 的对比效果。

图 8 是各里程计算法对序列 09 的估计轨迹对比,可以发现在所有算法中,本文提出的 3DPointLIO 比另外三种算法具有更高的预测精度,预测出的轨迹更为贴近真实轨迹。如图 8(a)所示,随着距离的增长,四种算法都产生了累积误差,但是本文提出的算法估计出的轨迹发生明显偏移的时间最晚,因为提出的 IMU 去噪模块使得 IMU 数据的累积误差降低,使用多层 Bi-LSTM 网络能够提取更好的 IMU 特征。观察图 8(b),可以发现在 x 轴(向前)、y 轴(向右)、z 轴(向上)这 3 个方向,本文提出的算法预测效果更为准确,特别是 x 轴和 y 轴这两个方向,与基准模型 DeepLIO 相比得到较大的提升,这是因为 3DPointLIO 提出的特征金字塔网络结合注意力机制,能够提取到比图中传统算法和 DeepLIO 更好的环境特征,对于动态物体有着更好鲁棒性。最终,通过特征金字塔网络、IMU 数据去噪网络和多层 Bi-LSTM 网络以及位姿估计网络构建的 3DPointLIO 表现出较好的预测效果,一定程度上验证了本文提出的里程计模型的有效性。



(a) 算法估计轨迹对比图
(a)Algorithnm-estimated trajectory comparision plot

(b) x轴、y轴和z轴轨迹对比图
(b) x-y-z-axis trajectory comparision plot

图 8　各算法在序列 09 上的轨迹估计对比

Fig. 8　Comparision of trajectory estimation among algorithms on sequence 09

3　结　　论

本文提出了一种融合三维点云特征和 IMU 数据进行位姿估计的深度学习里程计算法。利用深度学习网络直接学习 3 D 点云的特征，通过特征金字塔结构提取不同密度下的特征信息，结合注意力机制逐步过滤一些动态信息。结合 CNN 和 GRU 网络降低原始 IMU 数据的累积误差，并使用多层 Bi-LSTM 提取降噪后的 IMU 数据的时序特征。在数据集 KITTI 上进行的实验和测试表明，与其他常见的算法相比较，本文提出的 3DPointLIO 里程计算法具有更高的精度，在数据集的各个序列上都得到了较低的误差。

虽然 3DPointLIO 表现出了不错的效果，但是与真实轨迹相比仍然存在着一定误差，因此，在今后的研究中可利用特征金字塔网络各层的特征信息来对环境的动态信息进行充分的过滤、提取更为鲁棒的点云特征，以进一步减少误差。

参考文献

[1] 曾庆轩，李锵，聂为之. 基于伪占用区在线点云去除的激光雷达 SLAM 算法[J]. 光学学报，2023，43(20)：289-298.
ZENG Q X，LI Q，NIE W ZH. LiDAR SLAM algorithm based on online pseudo-occupancy area point cloud removal[J]. Acta Optica Sinica，2023，43(20)：289-298.

[2] ZHANG J，SINGH S. LOAM：Lidar odometry and mapping in real-time [C]. Robotics：Science and systems，2014，2(9)：1-9.

[3] SHAN T，ENGLOT B. Lego-loam：Lightweight and ground-optimized lidar odometry and mapping on variable terrain [C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS). IEEE，2018：4758-4765.

[4] 张冰战，尹晨晨，李志远，等. 基于点云特征全局搜索的回环检测算法[J]. 电子测量与仪器学报，2024，38(3)：176-186.
ZHANG B ZH，YIN CH CH，LI ZH Y，et al. Loop closure detection algorithm based on global search of point cloud features [J]. Journal of Electronic Measurement and Instrumentation，2024，38(3)：176-186.

[5] SHAN T，ENGLOT B，MEYERS D，et al. Lio-sam：Tightly-coupled lidar inertial odometry via smoothing and mapping [C]. 2020 IEEE/RSJ international conference on intelligent robots and systems(IROS). IEEE，2020：5135-5142.

[6] XU W，ZHANG F. Fast-lio：A fast，robust lidar-inertial odometry package by tightly-coupled iterated kalman filter [J]. IEEE Robotics and Automation Letters，2021，6(2)：3317-3324.

[7] NICOLAI A，SKEELE R，ERIKSEN C，et al. Deep learning for laser based odometry estimation[C]. RSS workshop Limits and Potentials of Deep Learning in Robotics，2016，184：1.

[8] VELAS M，SPANEL M，HRADIS M，et al. CNN for IMU assisted odometry estimation using velodyne LiDAR[C]. 2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). IEEE，2018：71-77.

[9] LI Q，CHEN S，WANG C，et al. Lo-net：Deep real-time lidar odometry[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition，2019：

8473-8482.

[10] CHO Y, KIM G, KIM A. Deeplo: Geometry-aware deep lidar odometry [J]. ArXiv preprint arXiv: 1902.10562, 2019.

[11] IWASZCZUK D, ROTH S. Deeplio: Deep lidar inertial sensor fusion for odometry estimation [J]. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2021, 1: 47-54.

[12] SUN L, DING G, QIU Y, et al. TransFusionOdom: Transformer-based LiDAR-inertial fusion odometry estimation[J]. IEEE Sensors Journal, 2023,23(18): 22064-22079.

[13] 李隆,安毅,谢丽蓉,等.基于深度学习与卡尔曼滤波的多模态融合里程计[J].激光与光电子学进展,2024,61(18):1-12.
LI L, AN Y, XIE L R, et al. Multimodal fusion odometry based on deep learning and kalman filter[J]. Laser & Optoelectronics Progress, 2024, 61(18): 1-12.

[14] LIU J, WANG G, JIANG C, et al. Translo: A window-based masked point transformer framework for large-scale lidar odometry[C]. AAAI Conference on Artificial Intelligence. 2023, 37(2): 1683-1691.

[15] ALI S A, AOUADA D, REIS G, et al. Delo: Deep evidential lidar odometry using partial optimal transport[C]. IEEE/CVF International Conference on Computer Vision, 2023: 4517-4526.

[16] 李一凡, 杜世通, 李爽, 等. 基于点云分割的激光里程计算法[J]. 全球定位系统, 2023, 48(4): 37-43.
LI Y F, DU SH T, LI S, et al. Laser odometry algorithm based on point cloud segmentation [J]. Global Positioning System, 2023, 48(4): 37-43.

[17] MIN Z, YANG Y, DUNN E. Voldor: Visual odometry from log-logistic dense optical flow residuals[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: 4898-4909.

[18] YANG N, STUMBERG L, WANG R, et al. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry [C]. IEEE/CVF conference on computer vision and pattern recognition. 2020: 1281-1292.

[19] 纪泽源,于潇颖,付文兴.基于图神经网络特征点匹配的视觉 SLAM 算法[J].仪器仪表学报,2024,45(9):34-43.
JI Z Y, YU X Y, FU W X. Visual SLAM algorithm based on feature point matching with graph neural networks[J]. Chinese Journal of Scientific Instrument, 2024, 45(9): 34-43.

[20] 张震宇, 杨小冈, 卢瑞涛, 等. VFE-VO: 视觉特征增强的光流法视觉里程计算法[J]. 激光与光电子学进展, 2025, 62(6): 118-125.
ZHANG ZH Y, YANG X G, LU R T, et al. VFE-VO: Visual feature-enhanced optical flow visual odometry algorithm [J]. Laser & Optoelectronics Progress, 2025, 62(6): 118-125.

[21] 黄友锐,王照锋,韩涛,等.结合轻量化 YOLOv5s 的动态视觉 SLAM 算法[J].电子测量技术,2024,47(11):59-68.
HUANG Y R, WANG ZH F, HAN T, et al. Dynamic visual SLAM algorithm integrating lightweight YOLOv5s [J]. Electronic Measurement Technology, 2024, 47(11): 59-68.

[22] 史 涛, 校诺政, 丁垚, 等. 动态场景下融合改进 YOLOv7 的视觉 SLAM 算法[J].国外电子测量技术, 2024,43(7):90-96.
SHI T, XIAO N ZH, DING Y, et al. Visual SLAM algorithm fusing improved YOLOv7 for dynamic scenes [J]. Foreign Electronic Measurement Technology, 2024, 43(7): 90-96.

[23] HUANG Z. Bidirectional LSTM-CRF models for sequence tagging [J]. ArXiv preprint arXiv: 1508.01991, 2015.

作者简介

张乔,硕士研究生,主要研究方向为基于激光的同时定位与建图。

E-mail:zhang19855603167@163.com

黄瑞,本科生,主要研究方向为机器人定位与导航。

E-mail: henrysky64@qq.com

张裕(通信作者),讲师,硕士生导师,主要研究方向为机器人感知与决策规划。

E-mail:yuzhang@sit.edu.cn

陈筱彦,硕士研究生,主要研究方向为基于视觉的同时定位与建图。

E-mail:yanza30290c163.com