



DOI:10.12404/j.issn.1671-1815.2403047

引用格式:王志航,杨华实,杨维,等. HKRNet: 高实时性点云配准轻量化框架[J]. 科学技术与工程, 2025, 25(11): 4629-4637.

Wang Zhihang, Yang Huashi, Yang Wei, et al. HKRNet: lightweight framework for high-realtime point cloud registration[J]. Science Technology and Engineering, 2025, 25(11): 4629-4637.

HKRNet: 高实时性点云配准轻量化框架

王志航, 杨华实, 杨维, 庞明喜, 陈治中, 巩昊杨, 王鼎衡*

(西北机电工程研究所, 智能装备与技术研究室, 咸阳 712099)

摘要 为解决 LIO-SAM (tightly-coupled lidar inertial odometry *via* smoothing and mapping) 等基于 ICP (iterative closest point) 方法的传统点云配准策略和 HRegNet (hierarchical registration network) 等基于深度神经网络方法的新型点云配准模型均存在的算力消耗高、配准时间长等问题, 通过深入研究 HRegNet 神经网络点云配准模型框架, 提出具有轻量化、实时性特点的 HKRNet (hierarchical kcpstack registration network) 网络模型。首先, 使用点云体素化和高斯阈值降采样联用的滤波方法, 去除雷达扫描地面的大量无用点, 将点的数量从 13 万左右降至约 7 万。其次, 对 HRegNet 模型内部耗时大的 K 最近邻 (K -nearest neighbors, KNN) 点云聚类算法改进为 KD 树 (K -dimensional tree, KD-Tree) 算法, 能够在保证精度的前提下提升 25% 的处理速度。最后, 针对模型内部卷积模块内存消耗高、计算效率低的问题, 使用张量分解的轻量卷积模块并提出分层奇异值分解算法, 将模型压缩至原来的 86.1% 并节约 61.2% 的计算量。结果表明, HKRNet 网络相对于 HRegNet 网络可以在微小的精度损失下, 减少 40% 的配准时间, 单次配准时间不超过 84 ms, 满足实时配准的使用需求。

关键词 点云配准; 深度学习; 模型轻量化; 点云下采样

中图分类号 TP274+.5;

文献标志码 A

HKRNet: Lightweight Framework for High-realtime Point Cloud Registration

WANG Zhi-hang, YANG Hua-shi, YANG Wei, PANG Ming-xi, CHEN Zhi-zhong,
GONG Hao-yang, WANG Ding-heng*

(Intelligent Equipment and Technology Research Laboratory, Northwest Institute of Mechanical & Electrical Engineering, Xianyang 712099, China)

[Abstract] To tackle the computational cost and registration time challenges in traditional point cloud registration methods like ICP (iterative closest point) such as LIO-SAM (tightly-coupled lidar inertial odometry *via* smoothing and mapping) and newer models utilizing deep neural networks such as HRegNet (hierarchical registration network), a lightweight and real-time HKRNet (hierarchical kcpstack registration network) network model was proposed. The model was developed by thoroughly studying the HRegNet neural network point cloud registration framework. Initially, a combined filtering approach involving point cloud voxelization and Gaussian threshold downsampling was used to remove redundant points from ground radar scans, reducing the point count from around 130 000 to about 70 000. Subsequently, the computationally intense KNN (K -nearest neighbors) point cloud clustering algorithm within the HRegNet model was enhanced by optimizing it to a KD-Tree (K -dimensional tree) algorithm, resulting in a 25% improvement in processing speed while upholding accuracy. Lastly, to address high memory usage and low computational efficiency of the convolutional modules in the model, a lightweight convolutional module leveraging tensor decomposition and a hierarchical singular value decomposition algorithm was introduced. This led to a compressed model size of 86.1% of the original and a decrease of 61.2% in computational cost. The outcomes indicate that the HKRNet network, in comparison to the HRegNet network, can reduce registration time by 40% with minimal loss of accuracy, achieving a single registration time not exceeding 84ms, thus meeting real-time registration requirements.

[Keywords] point cloud registration; deep learning; model lightweighting; point cloud downsampling

精准定位是实现汽车自动驾驶的必要前提, 现有相关算法主要通过激光点云配准来获取精确的

位置信息。其中, 传统点云配准方法主要包括迭代最近点算法^[1] (iterative closest point, ICP)、基于正

收稿日期: 2024-04-25 修订日期: 2024-07-30

基金项目: 国家自然科学基金青年科学基金 (12302267); 咸阳市重大科技专项 (L2023-ZDKJ-JSGG-GY-018)

第一作者: 王志航 (1998—), 男, 汉族, 黑龙江哈尔滨人, 硕士研究生。研究方向: 模式识别。E-mail: 1350152250@qq.com。

* 通信作者: 王鼎衡 (1988—), 男, 汉族, 陕西西安人, 博士, 助理研究员。研究方向: 武器装备智能化。E-mail: wangdai11@163.com。

投稿网址: www.stae.com.cn

态分布变换的 NDT(normal distribution transform)算法^[2]等。ICP 算法不需要提取特征或计算点云的特征,在较好的初值下就可获得不错的配准效果,但随着数据量增加,计算效率会大幅下降,有配准失败的风险。相比于 ICP 算法,NDT 算法的计算效率有很大的提升,但是其配准精度却有所下降。通过一种基于几何特征由粗到细点云配准算法^[3],可以改善对初始位置敏感且收敛速度慢的问题。但在配准大规模室外点云时,传统点云配准方法依然受限于点云的无序性和非结构化特性,很难配准成功,且会消耗大量的 CPU 算力。因此,基于深度学习的点云配准算法在近年受到越来越多的关注。

深度学习在三维物体检测、语义分割等众多三维计算机视觉任务中取得了巨大成功^[4],随之涌现出了许多基于深度学习的点云配准方法。这些方法一般通过构建深层卷积网络作为点云关键点检测器或局部特征描述符以学习鲁棒的局部区域特征,进而实现场景间准确的关键点匹配并进一步完成配准。

相较于传统的点云配准算法,基于深度学习的点云配准算法可以利用卷积模块的出众特征提取能力获得点云的几何结构特征,还可去除无序点的影响,从而获得更精确的配准效果。大多基于深度学习的点云配准方法是针对物体级点云^[5-6]或室内点云^[7-8]设计的,而室外点云通常具有更高的稀疏性、更大的空间范围和更复杂多变的分布,这使得配准变得十分困难,容易造成部分室内场景下的成功方法可靠性降低、耗时增加,难以应用于室外点云配准。

HRegNet^[9](hierarchical registration network)是最新的典型深度学习配准模型之一,专注于大规模室外点云配准,其内部基于 K 最近邻算法(K -nearest neighbors, KNN)点云聚类算法的关键点提取模块和特征描述符配准模块被多数配准网络效仿采用,使 HRegNet 形成事实上的深度学习点云配准通用框架之一。然而,常规 HRegNet 模型需要 130 ms 左右的配准时间,无法满足自动驾驶场景对定位实时性的需求。为解决该问题,围绕 HRegNet 神经网络点云配准模型框架进行三方面的优化改进设计,提出具有轻量化、实时性特点的 HKRNet(hierarchical kcpstack registration network)网络,可以在微小的精度损失(小于 2%)下减少 40% 的配准时间,使单次配准耗时不超过 84 ms,满足实时配准的使用需求。

1 HRegNet

1.1 骨干网络框架

HRegNet 是针对大规模室外点云配准的网络模

型。该模型受到基于深度学习的三维特征点云配准上的成功启发^[10-11],利用基于关键点分层点云配准网络进行配准。骨干网络模型结构及其数据流向如图 1 所示。首先从粗配准开始,匹配全局描述符空间中的关键点;然后,基于空间邻域利用其浅层的精确位置信息对粗配准精化,得到更精确的配准结果。由于只使用少量关键点而不是整个点云进行配准,该网络具有高于其他模型^[12-13]的计算效率,对自动驾驶等需要实时性能的场景具有较高应用潜力。

以 KITTI odometry 数据集的两帧点云配准为例,如图 2 所示。可见预测结果与真实结果相吻合。

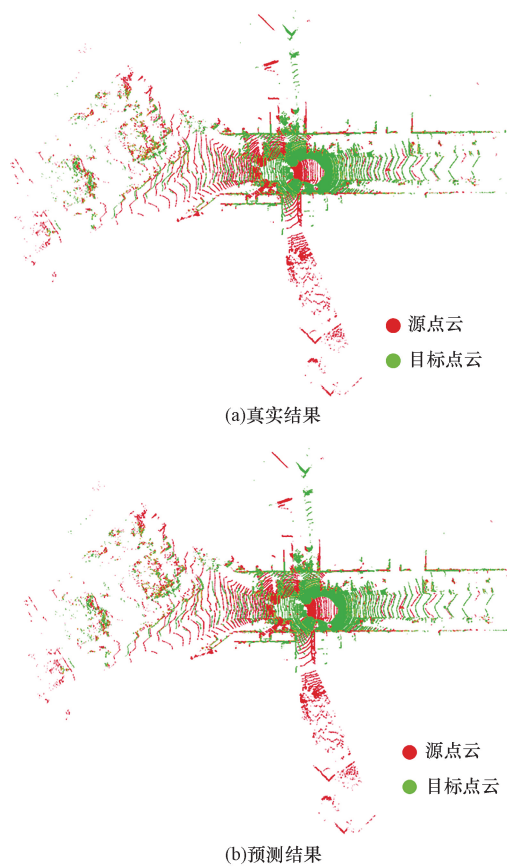


图 1 配准结果对比

Fig. 1 Comparison of registration results

1.2 推理瓶颈分析

尽管 HRegNet 相较其他模型有更少的配准时间以及参数量、计算量,但是在资源受限的条件下,约 130 ms 一帧的配准时间仍然无法满足工程化的实时性需求。分析模型主体结构可知其共有 6 种模块,统计推理一帧点云配准中各模块的运行时间如图 3 所示。可见 KNN 处理大量点云数据时会耗费一半左右的配准时间,且模型中大量的卷积操作会严重影响网络运行效率。根据上述分析,提出具有轻量化、实时性特点的 HKRNet 网络,具体针对 HRegNet 提出如下三方面改进。

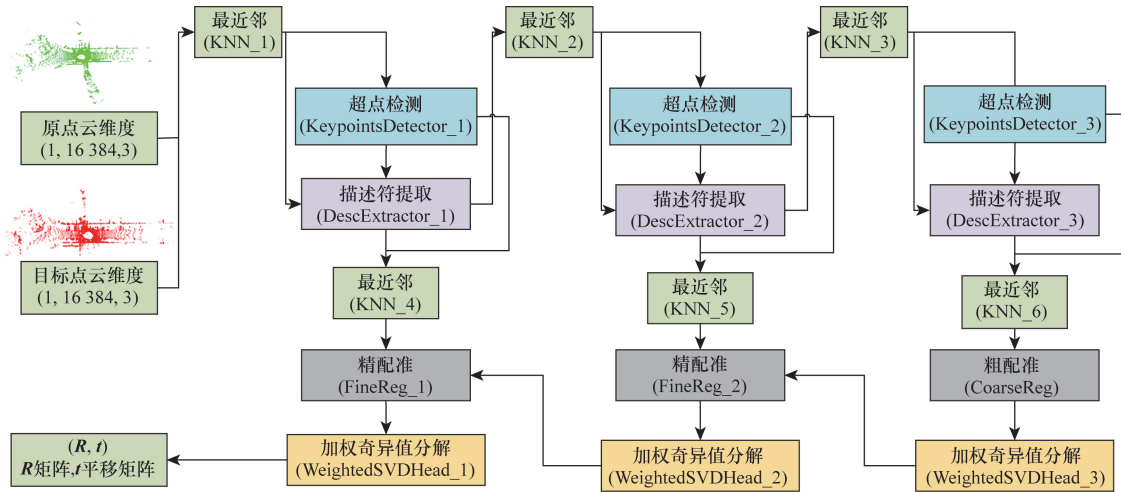


图2 HRegNet网络模型

Fig. 2 HRegNet network model

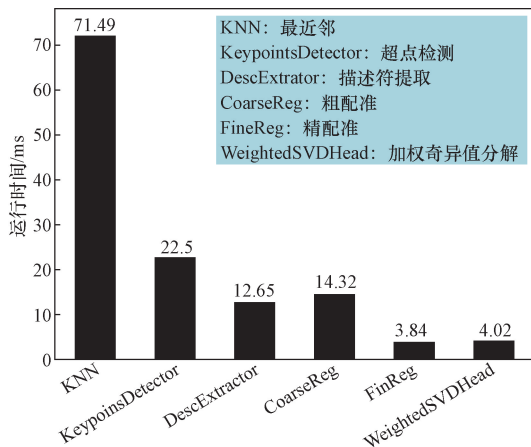


图3 HRegNet中各模块耗时

Fig. 3 Time consumption of each module in HRegNet

(1) 模型输入的点云数据(以KITTI odometry为例)由1个64线的Velodyne激光雷达采集,频率10 Hz,角分辨率为 0.09° ,每秒约一百三十万个点,水平视场 360° ,垂直视场 $\pm 26.8^\circ$,最多120 m的距离范围。受其垂直视场角以及探测范围的影响,采集点云中会有大量的地面点,而地面点对于配准任务属于无用点,去除地面点不会影响最终匹配效果。因此拟增加地面点云滤波算法处理输入点云数据以去除无用的地面点。

(2) 模型中大规模点云通过KNN算法进行聚类,从图3中可以看到,KNN聚类计算非常耗时,占用几乎一半的配准时间。因此本文中更高效、更适用于大规模室外点云配准场景的KD-Tree(K-dimensional tree)聚类算法作为替换。KD-Tree通过递归地将K维空间划分为嵌套的半空间,能够实现数据的快速检索和访问,大幅降低了计算复杂度,优化了聚类时间。

(3) 模型中有大量的卷积操作,使得该模型要面临卷积神经网络CNN(convolutional neural networks)在资源受限条件下的部署问题^[14]。基于一种具有可观压缩与加速能力的卷积核分层矩阵压缩算法KCPStack(convolutional kernel hierarchical matrix compression method)^[15],提出分层SVD(singular value decomposition)算法对模型中的大尺寸卷积进行压缩,以减少模型参数数量和计算量。

2 点云降采样

2.1 点云体素滤波

在数据预处理阶段首先对输入点云进行体素滤波处理。以KITTI odometry数据集中一帧点云为例,其点的数量约为13万。将该点云分配到一个由K个体素栅格组成的数据空间($K \times T \times F$)内,其中T为体素栅格中的最大点数,F为点的特征维度。然后计算每个体素栅格的中心点,也就是体素内所有点坐标的几何中心,以这些中心作为新的点云数据,此即体素滤波后的结果点云。本文中设置体素的大小为0.3 m,可将13万的数据量降至7万左右。

2.2 地面点云滤波

分析图4点云中待配准点的语义信息,超点皆为非地面点,可见非地面点才是对配准有益的特征点。因此考虑使用地面点云滤波算法来大量减少不影响最终匹配效果的原始点云数据,使其在同样数量输入点的情况下可以将更多的注意力集中在非地面点上。现有的地面点云滤波方法主要有阈值滤波^[16-17]、平面拟合^[18-19]、聚类方法^[20]等。考虑到算法的运行效率以及适用性问题,本文提出一种高斯阈值降采样方法,去除雷达扫描车身周围的地

面点。

考虑到自动驾驶场景中,车身相对道路面是平行的,该部分点云平面垂直雷达的 Z 坐标轴,因此滤波算法以点云 Z 坐标为基础,以 $-5 \sim 3$ 为统计区间,0.5 为区间间隔,计算数据量最大的区间范围,再去除该区间内的数据即可获得滤波结果。

图 5 所示为对原始点云采用本文所提两种滤波方法及联用之后的视觉效果。对比图 5(d)、图 5(b),在点云数据量相同的情况下,地面点云滤波可以排除 4 成的无用点,使配准时间缩短 30 ms,配准速度提升 27%。

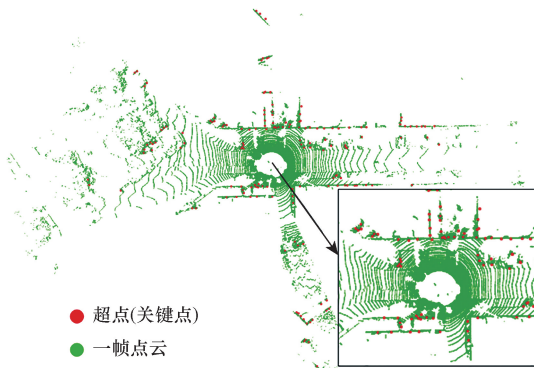


图 4 超点在点云中的位置

Fig. 4 The position of superpoints in point clouds

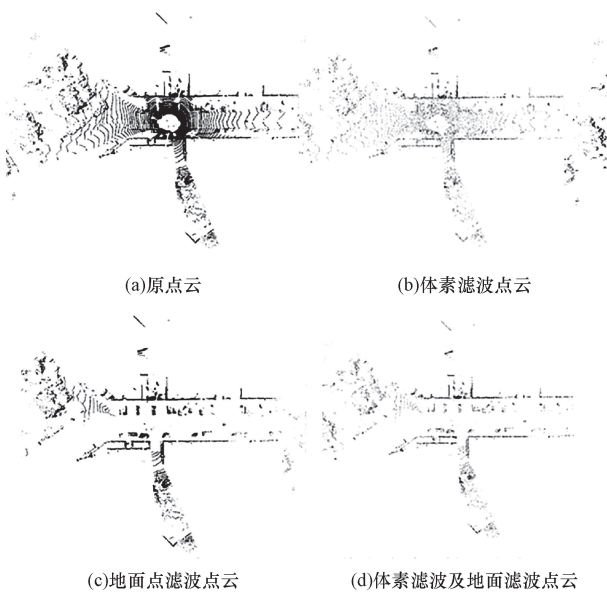


图 5 体素滤波和地面点滤波对比

Fig. 5 Comparison between voxel filtering and ground point filterin

3 最近邻优化

点云作为三维空间中的离散点集合,广泛应用于三维重建、物体识别、场景理解等领域。最近邻算法(KNN)作为一种简单而有效的机器学习方法,

在点云聚类中发挥着重要作用。然而,随着点云数据量的增加,KNN 的计算复杂度成为限制其性能的主要瓶颈。因此,需要探索更高效的数据结构以优化 KNN 算法,减少配准时间。

3.1 KD-Tree 算法

KD-Tree 是一种用于组织 K 维空间中实例点的数据结构,通过递归地将 K 维空间划分为嵌套的半空间,实现对数据的快速检索和访问,从而提高邻近搜索的效率。但是 KD-Tree 需要在数据预处理阶段构建树结构,这可能会增加算法的预处理时间和内存消耗。相比之下,KNN 是一种在线算法,可以在需要时直接计算样本之间的距离,无需提前构建任何数据结构。在不同的应用场景下,两种方法各有利弊。本文中主要面向大规模室外点云配准,具体以模型中实际输入数据为例,分析该场景下 KNN 和 KD-Tree 的适配性。

3.2 KNN 与 KD-Tree 性能对比分析

以图 2 模型中 KNN_1 模块的实际输入数据为例,构建点云 $A(16\ 384,3)$ 到点云 $B(1\ 024,3)$ 的 $K=64$ 个最近点的索引时,需要在点云 A 的每个点上查询点云 B 中的 K 个最近邻点,并且构建一个索引来加速这个过程。下面分别使用 KNN 和 KD-Tree 方法建立该索引,并对比两种方法的时间复杂度。

在 KNN 算法中,对于点云 A 中的每个点,需要计算它与点云 B 中所有点的距离,再通过排序找到最近的 K 个点。假设点云 A 有 N_1 个点,点云 B 有 N_2 个点, D 表示点特征维度。首先, A 中所有点到 B 中所有点距离的计算复杂度为 $O(N_1 N_2 D)$ 。其次,选择堆排序或快速选择算法来排序 A 中每个点到 B 中所有点的距离所需时间复杂度为 $O[N_1(N_2 + K \lg N_2)]$ 。最终,使用高效的排序方法后 KNN 算法整体的时间复杂度为 $O[N_1 N_2 D + N_1(N_2 + K \lg N_2)]$ 。

在 KD-Tree 算法中,需要先对点云 B 构建 KD-Tree,然后在点云 A 的每个点上执行最近邻搜索。其中构建 KD-Tree 的复杂度为 $O(N_2 \lg N_2)$ 。对于点云 A 中的每个点,在 KD-Tree 上执行 K 个最近邻搜索的复杂度通常远低于 $O(N_1 K \lg N_2)$ 且在最坏情况下不超过此值。因此通常情况下, KD-Tree 算法的总的时间复杂度远低于 $O(N_1 K \lg N_2 + N_2 \lg N_2)$ 。

针对给定的数据大小($N_1 = 16\ 384$, $N_2 = 1\ 024$, $D = 3$)和 $K = 64$,计算 KNN 和 KD-Tree 两种算法的时间复杂度并作具体数值对比。最优排序下 KNN 的时间复杂度约为 $O(67\ 386\ 084)$,最坏搜索情况下 KD-Tree 算法的时间复杂度为 $O(1\ 146\ 880)$ 。可

见即便是在最坏情况下, KD-Tree 的搜索效率都比 KNN 高很多。

为了进一步验证 KD-Tree 在模型中替换 KNN 方法的有效性,设计实验在相同输入条件下对比各自的运行时间。实验数据使用 KITTI odometry 的集合(8~10),输入数据与模型验证阶段的输入相同,首先从输入的 16 384 个点中截取 1 024 个点作为中心点,然后将源点云和裁剪后的点云作为输入,计算两种方法的运行时间,并对所有的点云聚类时间取平均,结果如表 1 所示。

表 1 两种方法的运行时间对比

Table 1 Comparison of runtime between two methods

算法	运行时间/ms			
	集合(8)	集合(9)	集合(10)	集合(8~10)
KNN	13.988 9	13.666 8	13.979 9	13.664 1
KD-Tree	11.612 2	10.227 2	10.080 7	10.261 9

实验结果表明,构建点云 $A(16\ 384, 3)$ 到点云 $B(1\ 024, 3)$ 的 $K=64$ 个最近点的索引时, KNN 耗时约 14 ms, KD-Tree 耗时约 10 ms, 其中构建 KD-Tree 消耗 1.5 ms, 搜索消耗 8.5 ms, 搜索时间占总时间的 85%。可见在大规模点云聚类中,需要的是更高效的搜索方法。在输入输出数据维度相同的情况下,基于 KD-Tree 的聚类方法速度可以提升 25%,且准确率与 KNN 方法相当。具体到 130 ms 的配准时间中, KNN 处理时间约占 70 ms, 使用 KD-Tree 优化后,只需约 47 ms,证明了 KD-Tree 替换 KNN 聚类的有效性。

综上所述,针对大规模点云配准的应用场景, KD-Tree 算法的搜索效率显著超越 KNN。KD-Tree 通过构建树形结构有效减少了冗余的比较次数,其在实际应用中的搜索复杂度往往远低于理论上的最坏情况。同时考虑到 KD-Tree 的构建时间占比较小甚至可忽略不计,因此本文中采用 KD-Tree 聚类算法来大幅缩短点云聚类的处理时间。

4 轻量卷积优化

HRegNet 网络的特征提取模块中,有大量 1×1 的 2D 卷积,且通道较宽,计算量、参数量都比较大。为减少模型参数量、计算量,降低模型部署压力,本文基于一种具有可观压缩与加速能力的卷积核分层矩阵压缩方法(KCPStack),提出分层 SVD 算法,对预训练模型中的大尺寸卷积进行压缩,简化卷积计算,解决模型推理计算效率较差的问题。

4.1 KCPStack 压缩算法

KCPStack 是一种具有可观压缩与加速能力的卷积核分层矩阵压缩方法。该方法首先将卷积运

算通过 im2col 算法的方式转换为矩阵乘法,其次借助 KCP 对卷积核转换后的权重矩阵进行 2 阶分解,接着将输入特征数据与 KCP 因子张量之间的运算过程优化为两个连续矩阵乘法,最后通过 col2im 的方式使卷积核及其计算转变为双层轻量卷积结构。具体地,设卷积的输入为 S 通道 $H \times W$ 尺寸的特征图 $X \in \mathbf{R}^{S \times H \times W}$,卷积核 $C \in \mathbf{R}^{T \times S \times h \times w}$ 的扫描窗口大小为 $h \times w$,输出特征图为 $Y \in \mathbf{R}^{T \times H' \times W'}$, T 为输出通道数, $H' \times W'$ 表示输出特征图的尺寸。先对卷积核的通道数进行 2 阶因数分解($S = S_1 S_2, T = T_1 T_2$),得到 $C^\dagger \in \mathbf{R}^{(S_1 S_2 h w) \times (T_1 T_2)}$,再据此对 C^\dagger 进行 2 阶 KCP 分解,得到 K 组因子张量,并将其堆叠成分块矩阵 $F^{(A)}, F^{(B)}$,即

$$\left\{ \begin{array}{l} F^{(A)} = \begin{bmatrix} \text{vec}[F_1^{(A_1)} (F_2^{(A_1)})^T] \\ \text{vec}[F_1^{(A_2)} (F_2^{(A_2)})^T] \\ \vdots \\ \text{vec}[F_1^{(A_K)} (F_2^{(A_K)})^T] \end{bmatrix}^T \in \mathbf{R}^{S_1 S_2 h w \times K} \\ F^{(B)} = \begin{bmatrix} \text{vec}[F_1^{(B_1)} (F_2^{(B_1)})^T] \\ \text{vec}[F_1^{(B_2)} (F_2^{(B_2)})^T] \\ \vdots \\ \text{vec}[F_1^{(B_K)} (F_2^{(B_K)})^T] \end{bmatrix}^T \in \mathbf{R}^{K \times T_1 T_2} \end{array} \right. \quad (1)$$

式(1)相对于原卷积核 C 减少了大量可训练参数。在推理计算时,可与 $F^{(A)}, F^{(B)}$ 依次相乘输出特征图 Y ,即

$$Y = (X F^{(A)}) F^{(B)} \quad (2)$$

4.2 分层奇异值分解(SVD)算法

KCPStack 算法通过直接定义 $F^{(A)}, F^{(B)}$ 两分块矩阵并替换原卷积核的方式设计整个 CNN,然后对其从头开始训练,并未充分利用相关基线预训练模型。具体到 HRegNet 模型,其中的卷积核大小都是 1×1 ,使用 KCPStack 直接设计压缩后的 HRegNet 模型再训练将会产生较大误差,无法满足精度要求。因此考虑充分利用已经具备较好精度的 HRegNet 预训练模型,即先对已训练模型的卷积核进行逐层正向分解,然后微调模型中的卷积压缩部分,避免模型整体的明显精度损失。为此,本文根据 KCP-Stack 形制特点提出分层 SVD 算法,具体计算过程如下。

首先将输入 X 和卷积核 C 通过 im2col 算法重构为 $X \in \mathbf{R}^{H' W' \times S h w}, C \in \mathbf{R}^{T \times S h w}$,其中 $S h w = S \times h \times w$ 。然后对重构后的卷积核进行第一次 SVD 分解,得到矩阵 $A \in \mathbf{R}^{T \times K}, B \in \mathbf{R}^{K \times S h w}$ 。再将矩阵 A 拆成 K 个 $T \times 1$ 的列向量,即 $A = \{a_1, a_2, \dots, a_K\}$,矩阵 B 拆成 K 个 $1 \times S h w$ 的行向量,即 $B =$

$\{\mathbf{b}_1^T, \mathbf{b}_2^T, \dots, \mathbf{b}_K^T\}^T$ 。再令 $S = S_1 \times S_2$, $T = T_1 \times T_2$, 将列向量 \mathbf{a}_i 转成矩阵 $\mathbf{A}_i \in \mathbf{R}^{T_1 \times T_2}$, 将行向量 \mathbf{b}_i 转成矩阵 $\mathbf{B}_i \in \mathbf{R}^{S_1 \times S_2 h w}$ 。进而对 K 组 \mathbf{A}_i 、 \mathbf{B}_i 矩阵进行第二次 SVD 分解, 得到 K 组因子张量 $\mathbf{F}_1^{(A_k)} \in \mathbf{R}^{S_1 \times R_k^{(A)}} \mathbf{F}_1^{(B_k)} \in \mathbf{R}^{T_1 \times R_k^{(B)}} \mathbf{F}_2^{(A_k)} \in \mathbf{R}^{S_2 h w \times R_k^{(A)}} \mathbf{F}_2^{(B_k)} \in \mathbf{R}^{T_2 \times R_k^{(B)}}$, 张量因子堆叠后即可获得式(1)中的分块矩阵 $\mathbf{F}^{(A)}$ 、 $\mathbf{F}^{(B)}$ 。其中, K 为第一层 SVD 分解的秩, $R_k^{(A)}$ 、 $R_k^{(B)}$ 为第二层 SVD 分解的秩。在推理时, 同样通过式(2)中简单的矩阵运算便可获得近似的推理结果。

综上, 分层 SVD 算法流程如下。

输入: 特征图 $\mathbf{X} \in \mathbf{R}^{S \times H \times W}$; 卷积核 $\mathbf{C} \in \mathbf{R}^{T \times S \times h \times w}$ 。

输出: 特征图 $\mathbf{Y} \in \mathbf{R}^{T \times H' \times W'}$ 。

过程:

1. 通过 im2col 算法重构输入 \mathbf{X} 和卷积核 \mathbf{C} , 得到 $\mathbf{X} \in \mathbf{R}^{H' \times W' \times S h w}$ 、 $\mathbf{C} \in \mathbf{R}^{T \times S h w}$;
2. 对卷积核 $\mathbf{C} \in \mathbf{R}^{T \times S h w}$ 进行第一层 SVD 分解, 得到 $\mathbf{A} \in \mathbf{R}^{T \times K}$ 、 $\mathbf{B} \in \mathbf{R}^{K \times S h w}$;
3. 将 \mathbf{A} 拆成 K 个 $T \times 1$ 的列向量, $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_K\}$;
4. 将 \mathbf{B} 拆成 K 个 $1 \times S h w$ 的行向量, $\mathbf{B} = \{\mathbf{b}_1^T, \mathbf{b}_2^T, \dots, \mathbf{b}_K^T\}^T$;
5. 令 $T = T_1 \times T_2$, 将列向量 $\mathbf{a}_{i(i=1,2,\dots,K)}$ 转成矩阵 $\mathbf{A}_i \in \mathbf{R}^{T_1 \times T_2}$;
6. 令 $S = S_1 \times S_2$, 将行向量 $\mathbf{b}_{i(i=1,2,\dots,K)}$ 转成矩阵 $\mathbf{B}_i \in \mathbf{R}^{S_1 \times S_2 h w}$;
7. 对 K 组 \mathbf{A}_i 、 \mathbf{B}_i 矩阵进行第二层 SVD 分解, 得到 K 组因子张量 $\mathbf{F}_1^{(A_k)} \in \mathbf{R}^{S_1 \times R_k^{(A)}} \mathbf{F}_1^{(B_k)} \in \mathbf{R}^{T_1 \times R_k^{(B)}} \mathbf{F}_2^{(A_k)} \in \mathbf{R}^{S_2 h w \times R_k^{(A)}} \mathbf{F}_2^{(B_k)} \in \mathbf{R}^{T_2 \times R_k^{(B)}}$;
8. 对 K 组因子张量进行堆叠获得式(1)中分块矩阵 $\mathbf{F}^{(A)}$ 、 $\mathbf{F}^{(B)}$;
9. 通过式(2)获得输出 $\mathbf{Y} \in \mathbf{R}^{T \times H' \times W'}$ 。

4.3 模型压缩效果

HRegNet 模型中, 卷积的输入通道数与输出通道数的倍数在 $[0.5, 6]$ 区间内, 从中选出输入通道 S 与输出通道 T 之比为 0.5、1、6 的三种情况进行对比, 结果如表 2 所示。设 K 为输出通道的一半, 当输出和输入通道数相同时, 参数量与分解前相同; 当输出通道数大于输入通道时, 参数量增加; 当输入通道数小于输出通道数时, 参数量减少。因此调整 K 的取值为 $T/3$, 并向下取偶数整数。此时模型中所有可压缩的卷积参数量都得到降低。

模型中满足压缩条件的是超点检测 (Keypoints-Detector) 和描述符提取 (DescExtractor) 两部分, 压缩前后参数量的对比如表 3 所示, HRegNet、HKRNet 是原模型和压缩模型, 可见压缩后两部分的三个模块参数量共下降了 35%。

HKRNet、HRegNet 的参数量和计算量对比如表 4 所示, 可见模型压缩至原来的 86.1% 并节约了 61.2% 的计算量, 实现了大幅降低模型参数量和计算量的效果。

表 2 不同 K 值下压缩前后参数量对比

Table 2 Comparison of parameter values before and after compression under different values

K	S	T	$T \times S \times h \times w$	$S h w \times K + K \times T$
T/2	32	32	1 024	1 024
	32	64	2 048	3 072
T/3	192	32	6 144	3 584
	32	32	1 024	640
T/3	32	64	2 048	1 920
	192	32	6 144	2 240

表 3 各模块压缩前后实际参数量对比

Table 3 Comparison of actual parameter values before and after compression of each module

模块	模型	模块一	模块二	模块三	全部模块
超点检测	HRegNet	12 094	50 817	199 937	262 848
	HKRNet	8 809	38 089	129 105	176 003
描述符提取	HRegNet	11 840	50 304	198 912	261 056
	HKRNet	8 064	32 920	129 072	170 056

表 4 模块压缩后实际参数量对比

Table 4 Comparison of actual parameter quantities after module compression

网络模型	计算量/G	参数量/M
HRegNet	10.635 4	2.467 8
HKRNet	4.146 8	2.119 5

5 HKRNet: 模型训练和评估

经上述点云降采样、最近邻优化和轻量卷积优化后的 HKRNet 网络框架如图 6 所示。

5.1 实验条件

训练集 KITTI odometry Seq (0 ~ 5), 验证集 Seq(6 ~ 7), 测试集 Seq(8 ~ 10)。模型为 HKRNet, 使用 PyTorch 实现, Adam 作为优化器, 迭代轮数 100, 学习率初始设置 0.001, 每 10 轮迭代下降 50%。网络在 NVIDIA RTX 3090 GPU 上进行训练, 测试集为 Seq(8 ~ 10), 测试设备 Intel Core i7-8700 CPU、NVIDIA GTX 1660 SUPER GPU。

5.2 评估指标

本文中采用式(3)定义的相对平移误差 (translation regression error, RTE) 和相对旋转误差 (rotation regression error, RRE) 评估配准结果 \mathbf{R} 、 \mathbf{t} , 其中 $(\hat{\mathbf{R}}, \hat{\mathbf{t}})$ 和 (\mathbf{R}, \mathbf{t}) 分别为预测变换矩阵和真实变换矩阵。当 RTE 和 RRE 在阈值 $\text{trans} = 1$ 和 $\text{rot} = 1$ 内时, 认为配准成功。

$$\begin{cases} \text{RTE} = \|\mathbf{t} - \hat{\mathbf{t}}\|_2 \\ \text{RRE} = \arccos \frac{\text{Tr}(\hat{\mathbf{R}}^T \mathbf{R}) - 1}{2} \end{cases} \quad (3)$$

5.3 模型训练

5.3.1 模型输入点数确定

HRegNet 模型训练的输入点数为 16 384,测试的输入点数为 10 000,模型的输入点数很大程度上决定了模型的运行时间,但是模型点数过少,则配准精度会大幅度下降。

选取 6 000 为模型的输入点数,原因有以下两点。

(1) 图 7 所示为 HRegNet 模型以 16 384 个点为输入进行训练的结果,测试集是 Seq10。从图 7 中可以看出,选择输入的点数为 6 000 点时,相对的

误差增加小,且配准成功率也较高,配准时间也下降了 10 ms 左右。

(2) 通过地面点云滤波处理,可以去除四成的无用点,因此原方法输入的 10 000 个点中,有约 6 000 个点对配准有益。

5.3.2 地面点云滤波效果

调整训练的输入点数为 6 000 点,使用 HRegNet 模型测试地面点云滤波方法对配准效果的影响,数据集为 Seq (8 ~ 10),输入点数为 6 000。结果如表 5 所示,经过地面点滤除后,配准精度、配准成功率等都有改善。

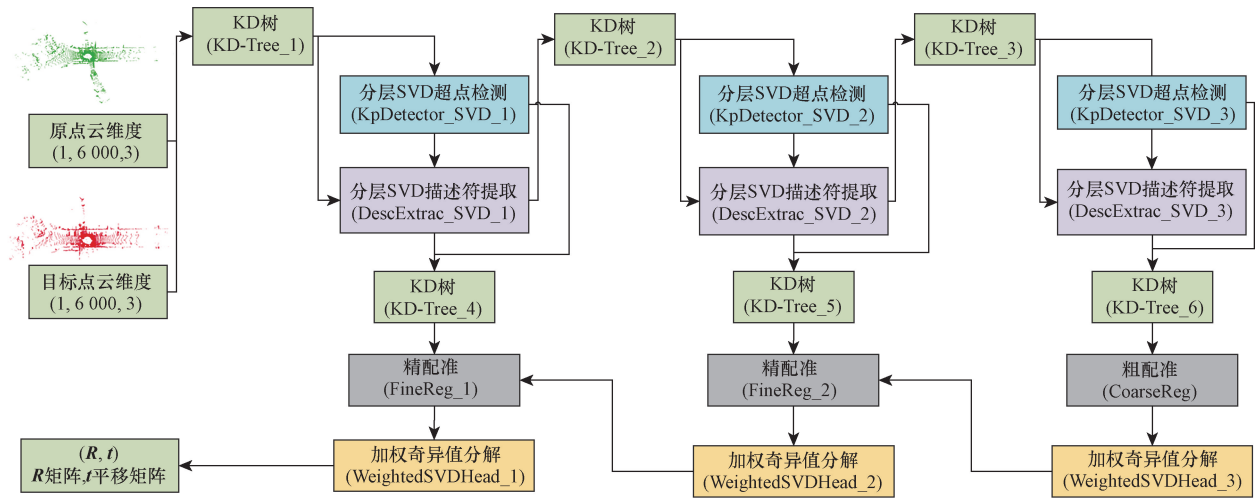


图 6 HKRNet 网络模型

Fig. 6 HKRNet network model

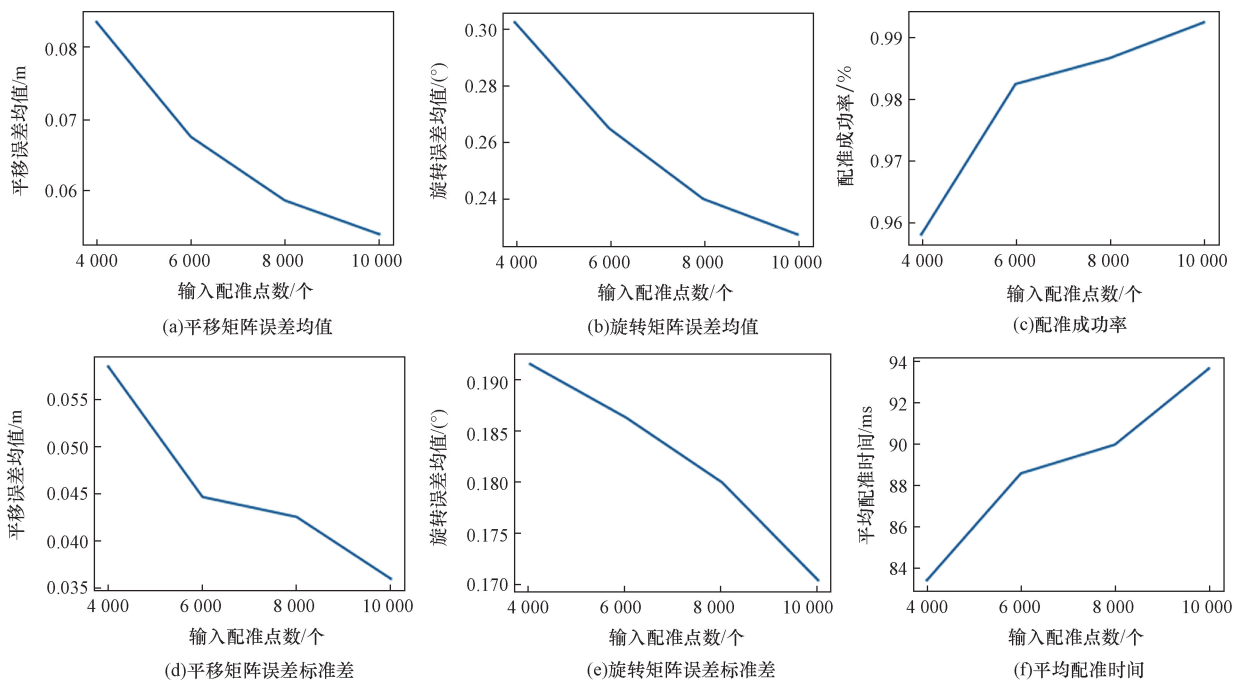


图 7 HRegNet 输入点数变化下的评价指标

Fig. 7 Evaluation metrics for HRegNet under changes in input points

表 5 有/无地面点云滤波的配准评估指标

Table 5 Registration evaluation indicators with/without ground point cloud filtering

评估指标	原方法	滤波方法
平移矩阵误差均值/m	0.074 2	0.069 8
平移矩阵误差标准差/m	0.053 7	0.050 4
旋转矩阵误差均值/(°)	0.268 7	0.225 6
旋转矩阵误差标准差/(°)	0.175 6	0.147 7
配准成功率/%	97.63	98.55
平均配准时间/ms	86.525 2	85.167 8

5.3.3 HKRNet 训练效果

训练 HKRNet 模型并与原 HRegNet 模型进行对比测试,测试集为 Seq (8 ~ 10)。如表 6 所示,可以看到使用分层 SVD 算法压缩卷积核,能够在大幅降低模型参数量和计算量的同时,保证对配准精度的最小影响。

表 6 优化前后模型配准评估指标

Table 6 Evaluation indicators for model registration before and after optimization

评估指标	HRegNet	HKRNet
平移矩阵误差均值/m	0.074 2	0.083 5
平移矩阵误差标准差/m	0.053 7	0.057 3
旋转矩阵误差均值/(°)	0.268 7	0.310 0
旋转矩阵误差标准差/(°)	0.175 6	0.215 0
配准成功率/%	97.63	95.96
平均配准时间/ms	134.525 2	82.833 4

6 结论

围绕 HRegNet 框架,本文提出高实时性点云配准轻量化框架 HKRNet,做出下述三处改进。使用点云体素化和高斯阈值降采样方法,去除点云中雷达扫描地面的大量无要点,将 13 万的数量点降至 7 万;改进 KD-Tree 算法处理点云,在保证精度的前提下提升 25% 的配准速度;提出分层 SVD 算法,先对已训练模型的卷积核进行逐层正向分解,然后微调模型中的卷积压缩部分无需再训练,该算法可将模型压缩至原来的 86.1% 并节约 61.2% 的计算量。改进后的 HKRNet 框架可以在精度微损(小于 2%)的情况下,减少 40% 的配准时间。

分层 SVD 算法体现出了张量分解灵活的拓扑结构变化,以及其优化轻量化神经网络模型推理计算效率的能力,但是对 1 × 1 的卷积压缩优势不够明显,模型中部分卷积无法压缩,针对该压缩方法的改进,是未来值得进一步研究的方向。

参 考 文 献

[1] Besl P J, McKay N D. A method for registration of 3-D shapes[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,

1992, 14(2): 239-256.

[2] Thierfelder T. Standard probability density functions of characterizing water quality variables in dimictic glacial/boreal lakes[J]. Water, Air, and Soil Pollution, 1999, 113: 295-317

[3] 胡加涛, 吴晓红, 何小海, 等. 一种基于几何特征由粗到细点云配准算法[J]. 科学技术与工程, 2020, 20(5): 1947-1952. Hu Jiatao, Wu Xiaohong, He Xiaohong, et al. A coarse to fine point cloud registration algorithm based on geometric features[J]. Science Technology and Engineering, 2020, 20(5): 1947-1952.

[4] Guo Y, Wang H, Hu Q, et al. Deep learning for 3D point clouds: a survey[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2021, 43(12): 4338-4364.

[5] Yuan W, Eckart B, Kim K, et al. DeepGMR: Learning latent gaussian mixture models for registration [EB/OL]//arXiv. org. libyc. nudd. edu. cn: 80. (2020-08-20) [2024-02-29]. https://arxiv.org/libyc.nudd.edu.cn/80/abs/2008.09088v1.

[6] Choy C, Dong W, Koltun V. Deep global registration[EB/OL]//arXiv. org. libyc. nudd. edu. cn: 80. (2020-04-24) [2024-02-29]. https://arxiv.org/libyc.nudd.edu.cn/80/abs/2004.11540v2.

[7] Pais G D, Ramalingam S, Govindu V M, et al. 3DRegNet: a deep neural network for 3d point registration [EB/OL]//arXiv. org. libyc. nudd. edu. cn: 80. (2019-04-02) [2024-02-29]. https://arxiv.org/libyc.nudd.edu.cn/80/abs/1904.01701v2.

[8] Gojcic Z, Zhou C, Wegner J D, et al. Learning multiview 3D point cloud registration [EB/OL]//arXiv. org. libyc. nudd. edu. cn: 80. (2020-01-15) [2024-02-29]. https://arxiv.org/libyc.nudd.edu.cn/80/abs/2001.05119v2.

[9] Lu F, Chen G, Liu Y, et al. HRegNet: A hierarchical network for efficient and accurate outdoor LiDAR point cloud registration[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(10): 1-14 .

[10] Bai X, Luo Z, Zhou L, et al. D3Feat: Joint learning of dense detection and description of 3D local features[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle: IEEE, 2020: 6358-6366

[11] Lu F, Chen G, Liu Y, et al. RSKDD-Net: random sample-based keypoint detector and descriptor [EB/OL]//arXiv. org. libyc. nudd. edu. cn: 80. (2020-10-23) [2024-02-29]. https://arxiv.org/libyc.nudd.edu.cn/80/abs/2010.12394v1.

[12] Qin Z, Yu H, Wang C, et al. GeoTransformer: fast and robust point cloud registration with geometric transformer [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(8): 9806-9821.

[13] Tang C, Li Y, Du S, et al. HD2Reg: hierarchical descriptors and detectors for point cloud registration[M/OL]. arXiv, 2023 [2023-12-07]. http://arxiv.org/abs/2305.03487.

[14] Deng B L, Li G, Han S, et al. Model compression and hardware acceleration for neural networks: a comprehensive survey[J]. Proceedings of the IEEE, 2020, 108(4): 485-532.

[15] 王鼎衡, 刘保荣, 杨维, 等. KCPStack: 张量分解的卷积核分层矩阵压缩方法 [J]. 西安交通大学学报, 2024, 58(3): 137-148. Wang Dingheng, Liu Baorong, Yang Wei, et al. KCPStack: Tensor decomposed compression method with layered matrices for convolutional kernels[J]. Journal of Xi'an Jiaotong University, 2024,

- 58(3): 137-148.
- [16] 丁少鹏, 刘如飞, 蔡永宁, 等. 一种顾及地形的点云自适应坡度滤波方法[J]. 遥感信息, 2019, 34(4): 108-113.
Ding Shaopeng, Liu Rufe, Cai Yongning, et al. A point cloud adaptive slope filtering method considering terrain [J]. Remote Sensing Information, 2019, 34(4): 108-113.
- [17] 董涛涛, 宋宇博. 基于自适应阈值的循环增长地面点云分割算法[J]. 科学技术与工程, 2024, 24(9): 3526-3532.
Dong Taotao, Song Yubo. Ground point cloud segmentation algorithm with cyclic growth based on adaptive threshold [J]. Science Technology and Engineering, 2024, 24(9): 3526-3532.
- [18] 储栋, 王磊, 李靖宇, 等. 基于 WPA-CSF 的地面点云滤波方法研究[J]. 地球物理学进展, 2023, 38(5): 1919-1930.
Chu Dong, Wang Lei, Li Jingyu, et al. Research on ground point cloud filtering method based on WPA-CSF [J]. Progress in Geophysics, 2023, 38(5): 1919-1930.
- [19] 李昊, 陈壮, 董振川, 等. 一种改进布料模拟滤波的地面点云提取方法[J]. 测绘科学, 2023, 48(5): 120-128.
Li Hao, Chen Zhuang, Dong Zhenchuan, et al. A ground point-cloud extraction method based on improved cloth simulation filtering [J]. Science of Surveying and Mapping, 2023, 48(5): 120-128.
- [20] 金立生, 贺阳, 王欢欢, 等. 基于自适应阈值 DBSCAN 的路侧点云分割算法[J]. 汽车工程, 2022, 44(7): 987-996.
Jin Lisheng, He Yang, Wang Huanhuan, et al. Point cloud segmentation algorithm based on adaptive threshold DBSCAN for roadside LiDAR [J]. Automotive Engineering, 2022, 44(7): 987-996.