

北京理工大学

本科生毕业设计（论文）

基于深度学习的端到端多实例点云配准

Deep Learning Based End-To-End Multi-instance Point Cloud
Registration

学 院： 自动化学院

专 业： 自动化

班 级： 06111902

学生姓名： 杨润一

学 号： 1120191211

指导教师： 由育阳

2023 年 5 月 19 日

原创性声明

本人郑重声明：所呈交的毕业设计（论文），是本人在指导老师的指导下独立进行研究所取得的成果。除文中已经注明引用的内容外，本文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

特此申明。

本人签名：_____ 日期：_____ 年 _____ 月 _____ 日

关于使用授权的声明

本人完全了解北京理工大学有关保管、使用毕业设计（论文）的规定，其中包括：①学校有权保管、并向有关部门送交本毕业设计（论文）的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存本毕业设计（论文）；③学校可允许本毕业设计（论文）被查阅或借阅；④学校可以学术交流为目的，复制赠送和交换本毕业设计（论文）；⑤学校可以公布本毕业设计（论文）的全部或部分内容。

本人签名：_____ 日期：_____ 年 _____ 月 _____ 日

指导老师签名：_____ 日期：_____ 年 _____ 月 _____ 日

基于深度学习的端到端多实例点云配准

摘要

三维点云配准是点云处理中的一项基本任务，其在机械臂、自动驾驶、自主运动机器人、即时定位与地图构建（Simultaneous Localization and Mapping, SLAM）等众多基于视觉方法的应用中起着关键的作用。随着人工智能、元宇宙、自动驾驶等技术的兴起，将深度学习应用于点云配准的工作已经日趋成熟。目前大多数点云配准任务研究主要集中在成对配准上。然而，在实际应用中，目标场景可能包含多个重复实例，我们需要估计模板点云与目标点云中这些重复实例之间的多个刚性变换，也就是多实例点云配准。

本文针对三维点云配准问题，特别是多实例点云配准，进行了深入研究。我们调研并整理了多实例点云配准的国内外研究现状，并总结了前人研究的优势和劣势。现有解决方案需要对大量假设进行采样以检测可能的实例并拒绝异常值，当实例和异常值的数量增加时，其鲁棒性和效率会显著降低。我们建议根据距离不变矩阵将噪声对应集直接分组到不同的簇中。通过聚类自动识别实例和异常值，达到稳健且快速表现。

基于这一研究背景，我们提出了新的多实例点云配准架构，高效对应聚类的多实例点云配准 (ECC)。分析聚类方法的点云配准，发现对于描述子的依赖性高，离群点比例高的时候效果会大大下降，所以我们提出了基于深度学习的多实例点云配准 (DMR)，首先利用对比学习来学习输入推定对应关系的分布良好的深度表示。然后基于这些表示，我们提出了异常值修剪策略和聚类策略，以有效地删除异常值并将剩余的对应关系分配给正确的实例，达到了更好的效果。

综上所述，我们的研究表明，基于深度学习的多实例点云配准方法在处理复杂的多实例点云配准问题上具有优秀的性能，提供了新的研究思路和方法。

关键词：点云配准；多实例；聚类；点对匹配；深度学习；对比学习

Deep Learning Based End-To-End Multi-instance Point Cloud Registration

Abstract

Three-dimensional point cloud registration is a fundamental task in point cloud processing, which plays a crucial role in a wide range of vision-based applications such as robotic arms, autonomous driving, autonomous mobile robots, and Simultaneous Localization and Mapping (SLAM). With the rise of artificial intelligence, metaverse, autonomous driving, and other technologies, the application of deep learning to point cloud registration has become increasingly mature. Current research on point cloud registration mostly focuses on pairwise registration. However, in practical applications, the target scene may contain multiple repeated instances, and we need to estimate multiple rigid transformations between the template point cloud and these repeated instances in the target point cloud, that is, multi-instance point cloud registration.

This paper conducts in-depth research on the problem of three-dimensional point cloud registration, especially multi-instance point cloud registration. We have surveyed and summarized the domestic and international research status of multi-instance point cloud registration and summarized the advantages and disadvantages of previous studies. Existing solutions require sampling a large number of hypotheses to detect possible instances and reject outliers, and their robustness and efficiency degrade significantly when the number of instances and outliers increases. We propose to directly group the noisy correspondence set into different clusters based on a distance invariance matrix. Through clustering, instances and outliers are automatically identified, achieving robust and fast performance.

Based on this research background, we propose a new multi-instance point cloud registration framework, Efficient Correspondence Clustering for Multi-instance Point Cloud Registration (ECC). After analyzing the point cloud registration of the clustering method, we find that the effect will drop significantly when it has high dependency on the

descriptor and a high proportion of outliers. Therefore, we propose Deep Learning-based Multi-instance Point Cloud Registration (DMR), which first uses contrastive learning to learn a good deep representation of the input estimation correspondence. Then, based on these representations, we propose an outlier pruning strategy and a clustering strategy to effectively remove outliers and allocate the remaining correspondences to the correct instances, achieving better results.

In conclusion, our research shows that the deep learning-based multi-instance point cloud registration method has excellent performance in dealing with complex multi-instance point cloud registration problems and provides new research ideas and methods.

Key Words: **Point Cloud Registration; Multi-instance; Clustering; Point Cloud Correspondence; Deep Learning; Contrastive Learning**

目 录

摘要	I
Abstract	II
第1章 绪论	1
1.1 研究背景和意义	1
1.1.1 三维点云配准	1
1.1.2 多实例点云配准	2
1.2 国内外研究现状	3
1.2.1 点云配准	3
1.2.2 三维目标检测和实例分割	4
1.2.3 多模型拟合	4
1.3 论文结构安排	5
1.4 小结	5
第2章 点云配准相关背景知识介绍	6
2.1 点云数据	6
2.1.1 点云数据特点	6
2.1.2 点云特征描述	7
2.2 刚体运动表示	8
2.2.1 旋转矩阵	8
2.2.2 变换矩阵	9
2.2.3 欧拉角	10
2.2.4 四元数	11
2.3 运动学参数估计	13
2.3.1 基于奇异值分解的线性代数求解	13
2.3.2 基于 Levenberg-Marquardt 算法的非线性优化求解	14
2.4 小结	17
第3章 基于深度学习的点云配准	18
3.1 标准数据集	18
3.1.1 ModelNet40	18
3.1.2 ShapeNet	19
3.1.3 Scan2CAD	20
3.1.4 多实例点云配准仿真数据集	20

3.1.5 多实例点云配准真实数据集	21
3.2 点云配准评价指标	21
3.2.1 基于特征提取与匹配的评价指标	22
3.2.2 基于刚体运动的评价指标	22
3.2.3 多实例点云配准评价指标	23
3.3 点云处理与配准相关深度学习技术	25
3.3.1 深度学习概述	25
3.3.2 多层感知机	25
3.3.3 卷积神经网络	28
3.3.4 注意力机制	29
3.4 基于深度学习的点云配准方法	29
3.4.1 PointNet & PointNet++ 网络结构简述	30
3.4.2 PREDATOR 架构简述	34
3.5 小结	38
第 4 章 基于深度学习的多实例点云配准	39
4.1 问题陈述	39
4.2 高效对应聚类的多实例点云配准 ECC	40
4.2.1 结构模型	40
4.2.2 基于不变型矩阵的聚类	40
4.2.3 快速点对关系聚类	41
4.2.4 递归簇细化	41
4.2.5 合并重复变换	42
4.2.6 从簇中提取变换	42
4.2.7 处理大量点对关系	43
4.2.8 训练细节	43
4.3 基于深度学习的多实例点云配准 DMR	43
4.3.1 对比学习介绍	43
4.3.2 系统模型	45
4.3.3 特征提取器	45
4.3.4 剪枝策略	46
4.3.5 聚类和变换估计	47
4.3.6 训练细节	48
4.4 小结	48

第 5 章 实验结果与分析	49
5.1 数据集	49
5.1.1 仿真数据集	49
5.1.2 真实数据集	49
5.2 高效对应聚类的多实例点云配准 ECC	49
5.2.1 仿真数据集	49
5.2.2 真实数据集	52
5.3 基于深度学习的多实例点云配准 DMR	52
5.3.1 仿真数据集	52
5.3.2 真实数据集	53
5.3.3 消融实验	54
5.4 小结	56
结 论	57
参考文献	58
致 谢	64

第 1 章 绪论

1.1 研究背景和意义

1.1.1 三维点云配准

21 世纪以来，人工智能技术的发展对于社会有着重大的影响，智能化成为工程技术突破的内核。机器能够进行快速计算、存储和处理大量数据，并通过互联网将社会连为一体。现在由人工智能驱动的新一代机器，它们可以越来越自主地解决复杂的任务，其中以视觉为核心的机器技术快速发展，机械臂、自动驾驶、自主运动机器人等进入了人们的视野。随着 2012 年 AlexNet^[1] 问世以来，深度学习方法打开了计算机视觉的新大门。越来越多的深度学习方法比如 VGG^[2]、ResNet^[3]、ViT^[4] 被用在了图像分类、分割、场景理解等任务中。为了更好的理解真实世界，人们开始尝试将深度学习方法用于三维数据中，随着激光雷达和 Kinect 等高精度传感器的快速发展，点云已经成为表示三维世界的主要数据格式。2017 年 PointNet^[5] 出现后，深度学习方法也同样被广泛应用在了点云处理中。

三维点云配准是点云处理中的一项基本任务^[5-7]，其在机械臂、自动驾驶、自主运动机器人等众多基于视觉方法的应用中起着关键的作用。首先是三维重建，生成完整的三维场景是各种计算机视觉应用的基础和重要技术，包括自动驾驶中的高精度三维地图重建、机器人技术中的三维环境重建等。例如，配准可以为机器人应用程序中的路线规划和决策构建三维环境。

其次，三维场景中的定位。三维场景中的定位和重定位对于机器人技术尤其重要。例如，无人驾驶汽车会估计其在地图上的位置及其与道路边界线的距离。点云配准可以将当前的实时三维视图与其所属的三维环境准确匹配，提供高精度定位服务。此应用表明，点云配准提供了机器和三维环境交互一种解决方案。

第三，位姿估计。将点云 A 与另一个点云 B 对齐可以生成与点云 B 相关的点云 A 的位姿信息。这个位姿信息可用于机器人决策。例如，点云配准可以获取环境中物体的位姿信息，以决定机械臂移动到哪里以准确抓取并移动物体。位姿估计为机器人三维环境理解提供了重要信息。

在国防安全、信息安全、环境安全等领域，无人机系统、自主导航、环境感知等技术应用愈发广泛。在这些应用中，点云配准也发挥着重要作用。例如，无人机系统

需要对目标进行跟踪，而点云配准可以提供目标的位姿信息，来实现目标跟踪。点云配准可以用于环境感知，用于分割、检测、识别等任务，从而实现环境感知。在自动驾驶、机器人自主导航中，高精度的点云配准算法可以提供高精度的 3D 地图场景重建，为自主机器提供视觉定位、路径规划、障碍物检测等技术保障^[8]。

1.1.2 多实例点云配准

点云配准旨在通过对源点云和目标点云之间进行刚性变换，使得源点云和目标点云尽可能重合。点云配准的输入是两个点云，输出是一个刚性变换矩阵。点云配准的目标是找到一个刚性变换矩阵，使得源点云和目标点云之间的距离最小。传统方法中，一般流程为查找匹配点，通过 SVD 等方法求解出变换矩阵。随着机器学习和深度学习算法的广泛使用，基于深度学习方法和组合优化方法进一步提高了点云配准的准确率^[9-11]。

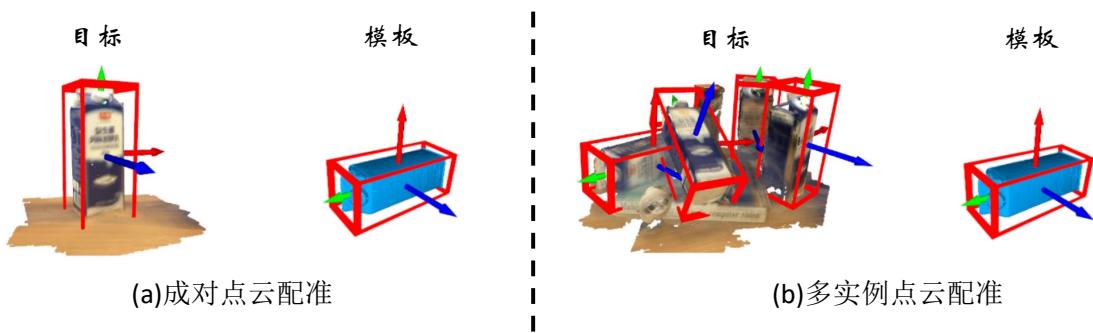


图 1-1 多实例点云配准：给定目标的模板点云，成对点云配准（左）侧重于估计模板点云和目标点云之间的单个刚性变换，而多实例点云配准（右）旨在估计目标点云中相同物体的 6D 位姿。

目前大多数点云配准任务研究主要集中在成对配准上。然而，在实际应用中，目标场景可能包含多个重复实例，我们需要估计模板点云与目标点云中这些重复实例之间的多个刚性变换。比如说在室内场景中，我们希望机器人能够将屋子中所有的椅子摆正，那么首先需要将多个椅子点云和模板椅子点云进行配准，求的目标椅子的位姿，通过机械运动来达到位姿改变的效果。图 1-1 展示了一个示例。这个问题被命名为多实例点云配准，它比成对点云配准更具挑战性。针对该任务已有的现有文献研究较少，扩展现有的点云配准方法来解决这个问题并非易事。多实例点云配

准不仅需要从嘈杂的对应中拒绝异常值，还需要识别单个实例的异常值集，这使得它比传统的配准问题更具挑战性。

与传统的两两配准方法相比，多实例点云配准需要解决更复杂的问题，同时也具有更广泛的应用价值。比如，在大规模场景重建任务中，通常需要处理成千上万个点云数据。单纯采用两两配准的方法可能导致累积误差，从而影响重建结果的精度。因此，研究多实例点云配准算法具有重要的实际意义。在机械臂抓取任务中，多实例点云配准算法可以在全局范围内考虑点云之间的约束关系，有助于消除局部误差和噪声的影响，从而提高配准结果的鲁棒性^[12]。

尽管多实例点云配准技术在近年来取得了显著的进展，仍然存在许多亟待解决的问题。例如，现存的多实例点云配准一般采用多任务的方式，也就是先对点云分割或者三维目标检测，然后进行两两点云配准，这样的方式需要先训练点云分割或者目标检测网络，泛化性差。并且如果见到了不存在先验的点云，下游的配准任务仍然会失效。所以，本文我们会对多实例点云配准进行研究，通过点云直接进行多实例点云配准，不需要先进行点云分割或者目标检测，从而提高多实例点云配准的泛化性。

1.2 国内外研究现状

1.2.1 点云配准

点云配准长期以来一直是计算机视觉和机器人领域的一项基本任务，大致可分为直接方法^[7,13] 和基于特征的方法^[5,14-15]。近年来，由于深度学习的发展，许多基于特征的方法取得了最先进的性能。这些方法通常通过特征匹配产生对应关系，然后移除异常值以稳健地估计转换。尽管深度特征^[5,14-16]发展迅速，但特征匹配生成的对应关系仍然包含异常值。因此，去除异常点在点云配准中具有重要意义。过去，已经提出了许多传统方法来去除异常值，包括基于 RANSAC 的方法^[17-19]、基于分支和边界的方法^[20] 以及许多其他方法^[14,21]。最近，一系列基于学习的方法^[22-23]被提出，并在异常值去除方面取得了显着的效果。以上的方法都是基于成对点云配准来完成的。然而，与成对配准不同，一个实例的内点构成多实例点云配准中所有其他实例的异常值。这种伪异常值使得很难将上述二元分类模型直接推广到多实例点云配准的情况。现有该问题解决方案包括采用目标检测方法或对目标点云应用实例分割，将多实例点云配准问题转化为多个成对点云配准问题，但是这种方法需要预先训练一

个目标检测或者点云分割网络，这样的方法对于已有点云类别是有效的，但是对于未知的类别是不适用的。另一种解决方案是通过多模型拟合，但是现有的多模型拟合方法依赖于抽样有效假设，当模型数量或离群率变高时，会涉及大量的抽样步骤，使得这些算法的效率和鲁棒性急剧下降。

1.2.2 三维目标检测和实例分割

三维物体的目标检测和实例分割与多实例点云配准有着密切的关系。输入一帧点云，目标检测模型^[24]可以用来对获取每个目标对象的边界框，三维实例分割^[25-26]为每个点生成实例标签。

这样的方法产生的结果类似于多实例点云配准的结果，但是它们需要将特定对象或类别的先验训练到网络中。基于点云匹配的筛选和聚类方法来进行多实例点云配准通过直接将模板点云和目标点云中的多个实例对齐来处理两组点云，而不使用任何关于输入的点云的先验信息。

1.2.3 多模型拟合

多实例配准也可以通过多模型拟合来实现，其目的是根据多个模型生成的数据点来进行建模。例如在点云中拟合多个平面^[27]，在运动分割中估计基本矩阵^[28]，在多实例点云配准中计算刚性变换^[29]等。但是由于一个实例的正常值构成所有其他实例的离群值，所以多模型拟合比单模型拟合更具挑战性。

现有的多模型拟合方法大致可以分为两类。第一类按顺序拟合模型^[17,20,30-31]，通过重复采样和筛选模型来进行建模。比如，Progressive-X^[30] 和 Progressive-X+^[17] 使用了表现更好的 Graph-cut RANSAC^[19] 作为采样方法来生成假设。CONSAC^[20] 首次将深度模型引入多模型拟合中，使用类似 PointNet^[5] 的网络来引导采样。通过重复采样来恢复单个实例，从输入中删除正常值和，以顺序的方式来检测实例。

第二类模型同时拟合多个模型^[29,32-35]。许多基于偏好分析的方法^[32,35] 最初对一系列假设进行采样，然后根据假设的残差对输入点进行聚类。ECC^[29] 利用点云刚性变换空间一致性^[36] 和以自下而上的方式基于距离不变矩阵对对应关系进行聚类。PointCLM^[37] 使用了一种新的深层表示方法来与空间一致性相结合，得到了更好的结果。

1.3 论文结构安排

本论文共分为 6 章。

第一章为绪论，本章主要阐述了三维点云配准研究的重要性、背景以及现有的研究进展。通过对国内外三维点云配准研究文献的梳理和调查，我们对该领域的发展轨迹和现状进行了总结，并分析了先前研究所展现出的优点与缺点。同时，我们也详细介绍了本研究所完成的主要任务和做出的关键贡献。考虑到过去的研究中仍存在未完全解决的问题和挑战，我们特别指出了本研究所应用的策略和解决方案。最后，我们规划了清晰的文章结构，以引导读者理解本研究的整体框架。

在第二章，我们主要阐述了三维点云的基本数据格式和在本研究中所涉及的基础数学知识。同时，我们对在本研究中所用到的关于三维刚体变换的基础知识进行了详尽的总结和描述。

第三章在介绍了主要的数据集和评价指标后，引出深度学习方法在三维点云配准领域的研究与应用，并结合深度学习在三维点云配准领域做出的代表性工作 PointNet^[5] 和 PointNet++^[38] 进行介绍，并介绍和使用 PREDATOR^[14] 作为骨干网络提取点云的点对特征供后续使用。

第四章主要介绍了本文提出的多实例点云配准，首先我们根据点云的特征进行聚类，然后对每个聚类进行对应聚类，最后进行刚性变换估计；其次，我们引入深度学习的方法，学习鲁棒的点对特征，进行谱聚类，得到了点云配准更好的效果和更高的指标。

第五章主要介绍了本文的实验结果，包括多实例点云配准的定性和定量实验，证明了本文提出的方法的有效性和鲁棒性。

1.4 小结

本章主要介绍了点云配准任务以及多实例点云配准任务的研究背景和研究意义、国内外研究现状。然后介绍了本文的研究内容和结构安排。

第 2 章 点云配准相关背景知识介绍

本章将对点云配准的基本理论和相关背景知识进行系统阐述，首先会介绍点云数据的特征，然后会对点云配准的刚体运动和运动参数估计进行介绍。

2.1 点云数据

三维世界中的数据表征有多种类型，如点云数据 (Point Clouds)^[39]、三角网格 (Triangle Mesh)^[40]、体素 (Voxel Grids)^[41]、深度相机数据 (RGB-D Camera)^[42] 等，可视化数据如图 2-1 所示，本文主要研究的是点云数据，因此本节将对点云数据进行简要介绍。

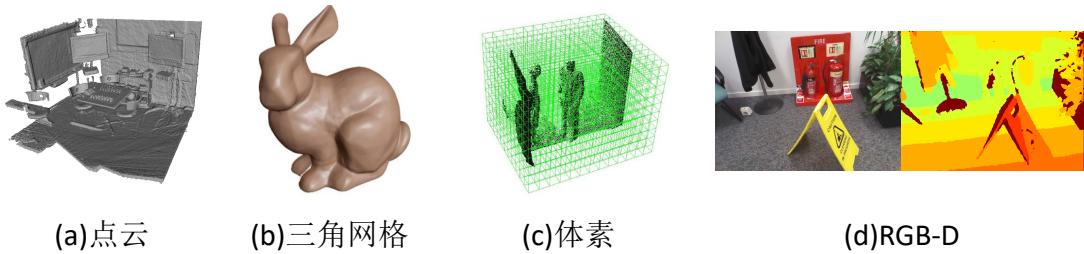


图 2-1 点云、三角网格、体素、深度相机数据可视化示意图

2.1.1 点云数据特点

点云数据是一种常用的三维数据表征形式，由一系列具有三维坐标 (X, Y, Z) 的点组成。这些点可以通过激光雷达、结构光传感器、多视角立体视觉等方式采集。点云数据具有以下特点：

1. 无序性：点云数据通常是无序的，即点在数据结构中的顺序并不表示它们在空间中的相对位置。这使得处理点云数据时需要额外关注邻域搜索等问题。
2. 不完整性：由于采集设备的视野限制以及物体遮挡等因素，点云数据往往只能捕捉到物体表面的部分信息，无法完整地描述物体的几何结构。
3. 稀疏性：点云数据在空间中分布可能是不均匀的，有些区域可能点密集，而有些区域可能点稀疏。这会对点云处理算法的性能产生影响。
4. 噪声敏感性：点云数据容易受到测量噪声、环境光照等因素的影响。为了提高数据质量，通常需要对点云进行预处理，如滤波、降采样等。

5. 缺乏拓扑信息：点云数据仅包含物体表面的几何信息，不包含拓扑信息。在需要考虑物体结构的应用场景中，点云数据需要进行进一步处理，如重建三角网格或提取骨架结构等。

6. 可扩展性：点云数据可以方便地扩展以包含其他属性信息，如颜色、法向量、强度等。这有助于提高点云处理算法的性能和鲁棒性。

7. 易于处理和存储：由于点云数据直接表示了物体表面的几何信息，其数据结构简单，便于处理和存储。此外，点云数据可以通过各种数据结构（如 KD 树、八叉树等）进行高效的组织和检索。

在应用深度学习算法时，由于点云数据的无序性，深度学习算法，比如全连接层、卷积层等无法直接应用于点云数据。因此，需要对点云数据进行预处理，将其转换为有序的数据表征形式，如三角网格^[40]、体素^[41]等。但是这样的方法会造成空间中很多点的浪费，使得输入的数据变得更加稀疏，使得训练数据变少。PointNet^[5]是一种用于处理点云数据的深度学习网络结构，于 2017 年首次提出。它是一个端到端的神经网络，可以直接从原始点云数据中学习特征表示。PointNet 通过使用对称函数（如最大池化）处理输入点云的无序性，同时具有对输入点顺序的不变性。

2.1.2 点云特征描述

三维点云的特征描述，也叫描述子 (Descriptor)，是一种用于表示点云数据中每个点周围的局部几何特征的向量。描述子捕捉了点云中每个点的几何结构和形状信息，这对于解决诸如点云配准、物体识别、分类和分割等问题至关重要。点云描述子应具有以下特性：鲁棒性、区分性、旋转不变性、尺度不变性和噪声不敏感性。点云描述子对于点云的后处理有着非常巨大的影响，对于不同的任务和数据特征，应该选用合适的描述子来作为网络的预处理。

有许多不同类型的点云描述子，它们根据计算方法和考虑的几何属性而有所不同。以下是一些常见的点云描述子：

1. Spin Images (旋转图像)^[43]：通过在点云中每个点周围投影二维图像来表示局部形状信息。

2. Normal Aligned Radial Features (NARF)^[44]：基于局部表面法线的方向和强度来描述点云中的局部特征。

3. Fast Point Feature Histograms (FPFH)^[45]：通过计算每个点周围的点对的几何

特征直方图来表示局部特征。

4. Signature of Histograms of Orientations (SHOT)^[46]：结合局部点的颜色信息和表面法线分布，为每个点计算描述子。

5. Point Pair Features (PPF)^[10]：描述点云中两个点之间的几何关系，PPF 是一个四元组 $(\alpha, d, \theta, \phi)$ ，其中 α 是两点之间的距离， d 是两点的法向量之差， θ 是两点的法线之间的角度， ϕ 是两点的法线在两点之间连线所确定的平面上的角度。PPF 特征在处理点云配准问题时具有很高的鲁棒性，因为它仅依赖于点云的几何信息。

2.2 刚体运动表示

刚体运动表示是点云配准任务的最后一个阶段的任务。对于位移来说，常用的是位移向量 (Transition)；旋转的运动参数有不同的表示形式，如欧拉角^[47]、四元数^[48]、旋转矩阵^[49]、轴角^[50]等。下面我们进行简要介绍。

2.2.1 旋转矩阵

旋转矩阵是一种与向量相乘时旋转向量同时保持其长度的矩阵。所有 3×3 旋转矩阵的特殊正交群表示为 $SO(3)$ 。因此，如果 $\mathbf{R} \in SO(3)$ ，那么

$$\det(\mathbf{R}) = \pm 1 \quad \text{且} \quad \mathbf{R}^{-1} = \mathbf{R}^T \quad (2-1)$$

对于满足 $\det(\mathbf{R}) = 1$ 的旋转矩阵，称为正规旋转；对于满足 $\det(\mathbf{R}) = -1$ 的旋转矩阵，称为非正规旋转。非正规旋转也称为旋转倒数，由旋转后接反演操作组成。我们将分析限制在正规旋转上，因为非正规旋转不是刚体变换。我们按如下方式引用旋转矩阵的元素：

$$\mathbf{R} = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix} \quad (2-2)$$

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2-3)$$

对于正规旋转矩阵，我们可以使用群的定义来进行更精准的定义。

$$SO(n) = \{ \mathbf{R} \in \mathbf{R}^{n \times n} | \mathbf{R}^T \mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1 \} \quad (2-4)$$

其中， $SO(n)$ 是特殊正交群 (Special Orthogonal Group)，这个集合由 n 维空间的旋转矩阵组成。 $SO(n)$ 是一个群，因为它满足群的四个条件：封闭性、结合律、单位元和逆元。 $SO(n)$ 的单位元是 \mathbf{I} ，逆元是 \mathbf{R}^T 。 $SO(n)$ 的元素称为正规旋转矩阵。通过旋转矩阵可以直接描述相机的旋转。

为了描述两个坐标之间的相对旋转，我们假设某个单位正交基 $\mathbf{i}, \mathbf{j}, \mathbf{k}$ ，并且我们希望将其旋转到另一个正交基 $\mathbf{i}', \mathbf{j}', \mathbf{k}'$ 。假设对于同一个向量 $\mathbf{a} = [a_1, a_2, a_3]^T$ ，在两个坐标系中的表示分别为 $\mathbf{a} = a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k}$ 和 $\mathbf{a}' = a'_1\mathbf{i}' + a'_2\mathbf{j}' + a'_3\mathbf{k}'$ ，根据坐标的定义，有

$$[\mathbf{i}, \mathbf{j}, \mathbf{k}]\mathbf{a} = [\mathbf{i}', \mathbf{j}', \mathbf{k}']\mathbf{a}' \quad (2-5)$$

我们对上述等式同时左乘 $[\mathbf{i}, \mathbf{j}, \mathbf{k}]^T$ ，我们可以通过旋转矩阵 \mathbf{R} 来表示两个坐标系之间的旋转关系，即

$$\mathbf{a}' = \begin{bmatrix} \mathbf{i}^T \mathbf{i}' & \mathbf{i}^T \mathbf{j}' & \mathbf{i}^T \mathbf{k}' \\ \mathbf{j}^T \mathbf{i}' & \mathbf{j}^T \mathbf{j}' & \mathbf{j}^T \mathbf{k}' \\ \mathbf{k}^T \mathbf{i}' & \mathbf{k}^T \mathbf{j}' & \mathbf{k}^T \mathbf{k}' \end{bmatrix} = \mathbf{R}\mathbf{a} \quad (2-6)$$

因为旋转矩阵是正交矩阵，它的逆就可以用来描述一个相反的旋转，按照上面的推到，则有：

$$\mathbf{a} = \mathbf{R}^T \mathbf{a}' = \mathbf{R}^{-1} \mathbf{a}' \quad (2-7)$$

在欧式变换中，除了旋转还有平移，将 \mathbf{a} 经过一次旋转 \mathbf{R} 和平移 \mathbf{t} ，最终得到了 \mathbf{a}' ，把旋转和平移组合起来，得到：

$$\mathbf{a}' = \mathbf{R}\mathbf{a} + \mathbf{t} \quad (2-8)$$

2.2.2 变换矩阵

在三维空间中，我们可以通过平移和旋转来描述一个刚体的变换。我们将平移和旋转组合起来，得到一个变换矩阵，用来描述一个刚体的变换。假设我们有一个刚体，它的初始位置在原点，我们将它平移到 \mathbf{t} ，然后将它旋转到 \mathbf{R} ，则这个刚体的

变换矩阵为：

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (2-9)$$

将向量改写为齐次形式 $\tilde{\mathbf{a}} = [a_1, a_2, a_3, 1]^T = [\mathbf{a}, 1]^T$, 那么我们可以将式 2-8 重写为：

$$\tilde{\mathbf{a}}' = \mathbf{T}\tilde{\mathbf{a}} = \mathbf{R}\mathbf{a} + \mathbf{t} \quad (2-10)$$

依靠变换矩阵，我们可以对多次变换的累加进行很简洁的数学表达。假设我们有一个刚体 \mathbf{a} , 它的初始位置在原点, 我们将它经过 $\mathbf{T}_1, \mathbf{T}_2$ 两次变换, 则这个刚体的最终位置为 $\tilde{\mathbf{a}}' = \mathbf{T}_1\mathbf{T}_2\tilde{\mathbf{a}}$ 。

2.2.3 欧拉角

三次坐标旋转依次可以描述任意旋转。我们考虑三次旋转, 其中第一次旋转是关于 \mathbf{k} 轴的角度 ψ , 第二次旋转是关于 \mathbf{j} 轴的角度 θ , 第三次旋转是关于 \mathbf{i} 轴的角度 ϕ , 如图 2-2 (a, b, c) 所示。为了简化符号, 我们将这些角度排列成一个三维向量, 称为欧拉角向量, 定义为

$$\mathbf{u} := [\phi, \theta, \psi]^T \quad (2-11)$$

将欧拉角向量映射到其对应的旋转矩阵的函数, $\mathbf{R}_{ijk} : \mathbf{R}^3 \rightarrow SO(3)$, 定义为

$$\mathbf{R}_{ijk}(\phi, \theta, \psi) := \mathbf{R}_i(\phi)\mathbf{R}_j(\theta)\mathbf{R}_k(\psi) \quad (2-12)$$

与一般情况相同, 如果 $\mathbf{z} \in R^3$ 是世界坐标系中的一个向量, $\mathbf{z}_0 \in R^3$ 是以体固定坐标表示的相同向量, 那么以下关系成立:

$$\mathbf{z}_0 = \mathbf{R}_{ijk}(\mathbf{u})\mathbf{z} \quad (2-13)$$

$$\mathbf{z} = \mathbf{R}_{ijk}(\mathbf{u})^T\mathbf{z}_0 \quad (2-14)$$

也就是说, 欧拉角并不是定义在三维线性空间的群, 欧拉角的相互转换需要借助旋转矩阵来进行表达。

在自动化或者航空航天领域比较常用的欧拉角的分解方式是 $Z - Y - X$, 也就是说, 先绕 z 轴旋转 ψ 角, 然后绕 y 轴旋转 θ 角, 最后绕 x 轴旋转 ϕ 角。这种分解

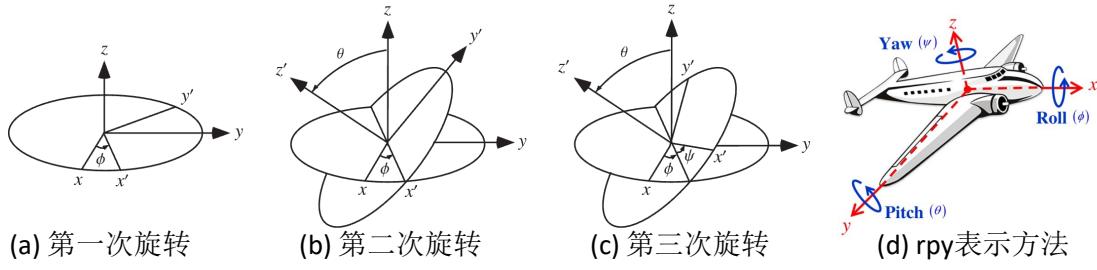


图 2-2 欧拉角旋转表示

方式也被称为航空分解 (Aerospace decomposition)，通常这种方式中的 $Z - Y - X$ 会被称为“偏航 - 俯仰 - 旋转 (yaw - pitch - roll)”。在这种分解方式中，欧拉角向量的顺序是 $\mathbf{u} = [\psi, \theta, \phi]^T$ ，如图 2-2 (d) 所示。

欧拉角的一个缺点是会碰到著名的万象锁问题^[51]，也就是说，当 $\theta = \pm\frac{\pi}{2}$ 时，旋转矩阵 $\mathbf{R}_{ijk}(\mathbf{u})$ 的值会变成奇异的。理论上可以证明，只要想要用 3 个数来表达三位旋转时，基本都会碰到奇异性问题^[52]。由于这种原理，欧拉角不适用于迭代和插值，在传统的 SLAM 中和深度学习中几乎都不会是用这种方式来进行迭代，但是欧拉角对于人机交互是友好的。

2.2.4 四元数

旋转矩阵使用了 9 个未知数来描述 3 自由度的旋转，这在数学表达上具有冗余性；欧拉角使用了 3 个未知数来描述 3 自由度的旋转，在数学表达上是紧凑的，但是具有奇异性。因为我们找不到不带奇异性，且使用 3 个未知数的方法来描述 3 维旋转的描述方式^[52]。这个性质，可以类比于用两个坐标描述地球表面 (二维流形) 上的点，我们可以使用经纬度来描述地球表面上的点，但是在极点处，经纬度的表达就会出现奇异性，即纬度为 $\pm 90^\circ$ 时，经度无意义。

三位旋转是一个三维流形，因此，为了最优平衡无奇异性和数学表达的紧凑性，我们需要引入一个新的数学工具来描述 3 自由度的旋转，这个数学工具就是四元数。四元数是 Hamilton 找到的一种超复数，它在数学形式上既是紧凑的，又是非奇异的。四元数的定义如下：

$$\mathbf{q} = q_0 + q_1 i + q_2 j + q_3 k \quad (2-15)$$

其中， $q_0, q_1, q_2, q_3 \in R$ ， i, j, k 是四元数的三个虚数单位，满足如下关系：

$$\begin{cases} i^2 = j^2 = k^2 = ijk = -1, \\ ij = k, jk = i, ki = j, \\ ji = -k, kj = -i, ik = -j \end{cases} \quad (2-16)$$

假设某个旋转是绕着单位向量 $\mathbf{u} = [u_x, u_y, u_z]^T$ 旋转 θ 角，那么这个旋转可以用四元数来表示为：

$$\mathbf{q} = \cos \frac{\theta}{2} + \sin \frac{\theta}{2} (u_x i + u_y j + u_z k) \quad (2-17)$$

同时，我们也可以从单位四元数种计算出对应的旋转轴和旋转角：

$$\begin{cases} \theta = 2 \arccos q_0, \\ u_x = \frac{q_1}{\sin \frac{\theta}{2}}, u_y = \frac{q_2}{\sin \frac{\theta}{2}}, u_z = \frac{q_3}{\sin \frac{\theta}{2}} \end{cases} \quad (2-18)$$

因为四元数是定义在复向量空间中而非旋转流形中，并且四元数没有死锁的特性，所以在深度学习的梯度下降算法中，常常用四元数作为回归的目标。并且，因为三维流形的性质，我们在四元数中同乘一个常数，可以得到相同的旋转，即 $[q_0, q_1, q_2, q_3] \iff k[q_0, q_1, q_2, q_3]$ 。由于这个性质，在回归任务中，我们通常回归单位四元数，即 $\sum_0^3 q_i^2 = 1$ 。

用四元数表示旋转。用四元数同样也可以表达对一个点的旋转，假设一个空间中的三维坐标点为 $\mathbf{a} = [x, y, z] \in \mathbb{R}^3$ 。以及一个旋转轴 \mathbf{n} 和旋转角 θ ，这个点绕着旋转轴旋转 θ 角后的坐标为 \mathbf{a}' ，则：

$$\mathbf{p}' = \mathbf{R}\mathbf{p} = \mathbf{q}\mathbf{p}\mathbf{q}^{-1} \quad (2-19)$$

其中， $\mathbf{q} = [\cos \frac{\theta}{2}, \mathbf{n} \sin \frac{\theta}{2}]$ 。

2.3 运动学参数估计

2.3.1 基于奇异值分解的线性代数求解

奇异值分解^[53] (Singular Value Decomposition, SVD) 是线性代数中一种重要的矩阵分解，在信号处理、计算机视觉、运动估计中有着非常广泛的运用。下面，我们对奇异值分解的线性代数部分进行简要介绍。

奇异值分解对一个矩形数据矩阵 (定义为 \mathbf{A} , 其中 \mathbf{A} 是一个 $n \times p$ 矩阵) 进行处理，其中 n 行代表观察值， p 列代表变量。SVD 定理如下：

$$\mathbf{A}_{n \times p} = \mathbf{U}_{n \times n} \mathbf{S}_{n \times p} \mathbf{V}_{p \times p}^T \quad (2-20)$$

其中

$$\mathbf{U}^T \mathbf{U} = \mathbf{I}_{n \times n} \quad (2-21)$$

$$\mathbf{V}^T \mathbf{V} = \mathbf{I}_{p \times p} \quad (2-22)$$

即 \mathbf{U} 和 \mathbf{V} 是正交的。 \mathbf{U} 的列是左奇异向量 (观察值系数向量)； \mathbf{S} (与 \mathbf{A} 的维度相同) 具有奇异值，并且是对角的 (模振幅)； \mathbf{V}^T 的行是右奇异向量 (变量水平向量)。SVD 表示了原始数据在协方差矩阵为对角线的坐标系中的扩展。

计算 SVD 包括寻找 $\mathbf{A}\mathbf{A}^T$ 和 $\mathbf{A}^T\mathbf{A}$ 的特征值和特征向量。 $\mathbf{A}^T\mathbf{A}$ 的特征向量构成 \mathbf{V} 的列， $\mathbf{A}\mathbf{A}^T$ 的特征向量构成 \mathbf{U} 的列。此外， \mathbf{S} 中的奇异值是来自 $\mathbf{A}\mathbf{A}^T$ 或 $\mathbf{A}^T\mathbf{A}$ 的特征值的平方根。奇异值是 \mathbf{S} 矩阵的对角线条目，并按降序排列。奇异值总是实数。如果矩阵 \mathbf{A} 是实数矩阵，那么 \mathbf{U} 和 \mathbf{V} 也是实数。

奇异值分解 (SVD) 在点云配准问题中也具有重要应用。点云配准是将两个或多个点云数据集合并为一个统一坐标系的过程。在这种情况下，我们的目标是找到一个最优的刚体变换，包括旋转和平移，使得两个点云之间的距离最小化。

假设我们有两个点云数据集 \mathbf{P} 和 \mathbf{Q} ，每个点云包含 n 个点。我们首先计算两个点云的质心 \mathbf{p}_c 和 \mathbf{q}_c ，然后将点云平移到原点。接下来，我们计算点云 \mathbf{P} 和 \mathbf{Q} 之间的距离矩阵 \mathbf{H} ，其中 $\mathbf{H} = \mathbf{P}^T \mathbf{Q}$ 。

在这种情况下，我们可以使用 SVD 来求解最优的旋转矩阵 \mathbf{R} ，使得两个点云之间的距离最小化。具体来说，我们对距离矩阵 \mathbf{H} 进行奇异值分解：

$$\mathbf{H} = \mathbf{U} \mathbf{S} \mathbf{V}^T \quad (2-23)$$

然后，我们计算旋转矩阵 \mathbf{R} :

$$\mathbf{R} = \mathbf{U} \mathbf{V}^T \quad (2-24)$$

如果 \mathbf{R} 是一个不合适的旋转矩阵(即 $\det(\mathbf{R}) \neq 1$)，我们可以通过修改 \mathbf{S} 矩阵来修复它。具体而言，我们将 \mathbf{S} 的最小奇异值设为 -1 ，然后重新计算旋转矩阵 \mathbf{R} 。

最后，我们可以计算最优的平移向量 \mathbf{t} :

$$\mathbf{t} = \mathbf{q}_c - \mathbf{R}\mathbf{p}_c \quad (2-25)$$

通过应用旋转矩阵 \mathbf{R} 和平移向量 \mathbf{t} ，我们可以将点云 \mathbf{P} 对齐到点云 \mathbf{Q} ，从而实现点云配准。

2.3.2 基于 Levenberg-Marquardt 算法的非线性优化求解

在数学和计算领域，Levenberg-Marquardt 算法 (LMA 或简称 LM)^[54]，也称为阻尼最小二乘法 (DLS)，用于求解非线性最小二乘问题。这些最小化问题尤其出现在最小二乘曲线拟合中。LMA 在高斯-牛顿算法 (GNA) 和梯度下降法之间进行插值。LMA 比 GNA 更稳健，这意味着在许多情况下，即使它从距离最终最小值非常远的地方开始，也能找到解决方案。对于行为良好的函数和合理的初始参数，LMA 往往比 GNA 慢。LMA 也可以被视为使用信任区域方法的高斯-牛顿算法。

需要 LM 算法求解的问题被称为非线性最小二乘最小化问题。这意味着要最小化的函数具有以下特殊形式：

$$f(x) = \frac{1}{2} \sum_{j=1}^m r_j^2(x) \quad (2-26)$$

其中 $x = (x_1, x_2, \dots, x_n)^T$ 是一个向量，每个 r_j 是一个从 \mathbb{R}^n 到 \mathbb{R} 的函数。 r_j 被称为残差，并且假定 $m \geq n$ 。

为了简化问题，我们用残差向量 $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 来表示 f ，定义为：

$$r(x) = \begin{bmatrix} r_1(x) & r_2(x) & \cdots & r_m(x) \end{bmatrix} \quad (2-27)$$

现在, f 可以重写为 $f(x) = \frac{1}{2}\|r(x)\|^2$ 。 f 的导数可以用相对于 x 的 r 的雅可比矩阵 $J(x)$ 来表示, 定义为 $J(x) = \frac{\partial r_j}{\partial x_i}$, 其中 $1 \leq j \leq m$, $1 \leq i \leq n$ 。

首先考虑线性情况, 其中每个 r_i 函数都是线性的。在这里, 雅可比矩阵是常数, 我们可以将 r 表示为空间中的超平面, 这样 f 给出二次型:

$$f(x) = \frac{1}{2}\|Jx - r_0\|^2 \quad (2-28)$$

我们还得到 $\nabla f(x) = J^T(Jx - r_0)$ 和 $\nabla^2 f(x) = J^T J$ 。通过将 $\nabla f(x) = 0$ 求解最小值, 我们得到 $x_{\min} = (J^T J)^{-1} J^T r$, 这是一组无约束优化问题的解。

回到一般的非线性情况, 我们有:

$$\nabla f(x) = J(x)^T r(x) \quad (2-29)$$

$$\nabla^2 f(x) = J(x)^T J(x) + \sum_{j=1}^m r_j(x) \nabla^2 r_j(x) \quad (2-30)$$

最小二乘问题的独特属性是, 给定雅可比矩阵 J , 如果可以将 r_j 用线性函数近似 ($\nabla^2 r_j(x)$ 较小) 或残差 $r_j(x)$ 本身较小, 我们可以实质上免费获得海森矩阵 ($\nabla^2 f(x)$)。在这种情况下, 海森矩阵简化为:

$$\nabla^2 f(x) = J(x)^T J(x) \quad (2-31)$$

这与线性情况相同。这里通常使用的近似是在解附近的 r_i 近似线性, 这样 $\nabla^2 r_j(x)$ 较小。还要注意, 只有在残差较小时, 式 2-31 才有效。对于大残差问题, 不能使用二次近似求解^[55]。

考虑到点云配准问题, 我们可以将 Levenberg-Marquardt 算法应用于点云之间的非线性最小二乘优化问题。在这种情况下, 我们寻求点云对应度量的最小距离, 同时考虑旋转和平移变换。通过迭代地优化变换参数, Levenberg-Marquardt 算法可以在点云配准问题中找到一个稳定的解决方案。点云对应度量之间的误差主要有点到

点 (point-to-point)、点到面 (point-to-plane)、面到面 (plane-to-plane) 这几类方式。如图 2-3 所示，我们以最小化点到面距离为例，来说明 Levenberg-Marquardt 算法的求解过程。

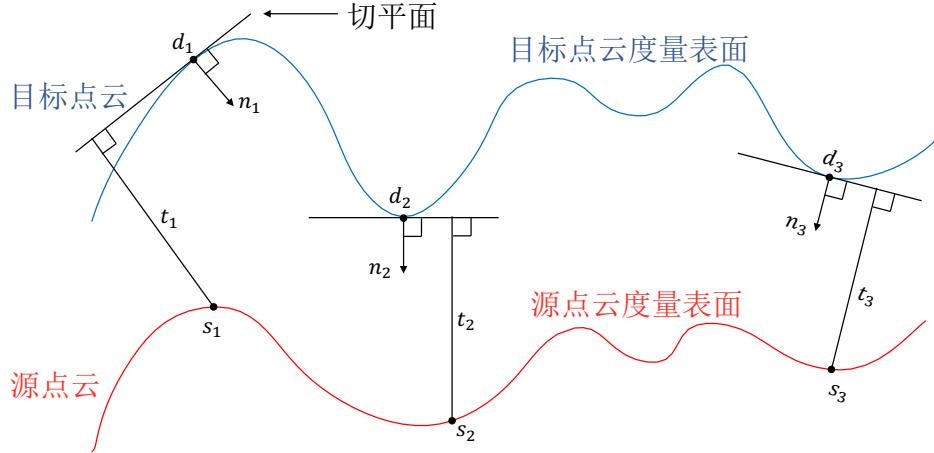


图 2-3 最小化点到面距离示意图

在刚体运动参数估计任务中，LM 算法的关键步骤包括：

- (1) 在运动前的点云数据中找到特征点 \mathbf{p}_i 和运动后的点云数据中对应的特征点 \mathbf{q}_j 以及特征面 S_l ；
- (2) 计算特征点 \mathbf{p}_i 到特征面 S_l 的欧氏距离 d_e ；
- (3) 基于特征点到特征面的距离度量，构建匹配对的约束关系：

$$f(\mathbf{p}_i, \mathbf{T}) = d_e \quad (2-32)$$

其中， \mathbf{T} 代表刚体运动变换参数。

利用 LM 非线性优化方法，估计位姿变换矩阵：

$$\hat{\mathbf{T}} = \mathbf{T} - (\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I})^{-1} \mathbf{J}^T f \quad (2-33)$$

通过链式法则计算雅可比矩阵：

$$\mathbf{J} = \frac{\partial f}{\partial \mathbf{T}} = \frac{\partial f}{\partial \mathbf{p}_i} \frac{\partial \mathbf{p}_i}{\partial \mathbf{T}} \quad (2-34)$$

将雅可比矩阵带入公式 2-33，可以求解刚体运动变换参数 \mathbf{T} 。

(4) 检查变换参数是否为最优估计，如果满足以下条件：

$$0 < \frac{f(\mathbf{p}_i, \hat{\mathbf{T}}) - f(\mathbf{p}_i, \mathbf{T})}{|f(\mathbf{p}_i, \mathbf{T})|} < T_d \quad (2-35)$$

则当前变换参数为最优估计。否则，调整阻尼因子 λ ，继续迭代求解最优变换参数。其中， T_d 是距离阈值。

(5) 将求解得到的最优变换参数应用于运动前的特征点 \mathbf{p}_i ，得到 \mathbf{q}'_j ，如果满足以下条件：

$$|\mathbf{p}_i - \mathbf{q}_j| > |\mathbf{p}_i - \mathbf{q}'_j| \quad (2-36)$$

则认为 \mathbf{q}_j 是点 \mathbf{p}_i 的最佳匹配点。否则，减小阻尼因子 λ ，继续迭代求解最优匹配点对。如果达到最大迭代次数仍未找到最优匹配点对，将点 \mathbf{p}_i 视为噪点并从原始点云数据中剔除，继续处理其他特征点。

总之，使用 LM 算法进行点云配准的过程可以分为以下几个步骤：

1. 在运动前后的点云数据中提取特征点和特征面；
2. 计算特征点之间的欧氏距离并建立约束关系；
3. 使用 LM 算法进行非线性优化，求解刚体运动变换参数；
4. 根据求解结果判断当前变换参数是否为最优估计；如非最优，调整阻尼因子并继续迭代；
5. 应用求解得到的最优变换参数，寻找最佳匹配点对；
6. 若达到最大迭代次数仍未找到最优匹配点对，将当前特征点视为噪点并剔除，继续处理其他特征点。

2.4 小结

本章主要介绍了点云配准任务的基础知识，包括点云数据的特点，运动学表达和刚体运动参数估计方法。主要介绍了两种最常用的点云配准求解刚体运动最优变换参数的方法，一个是基于奇异值分解 (SVD) 的线性代数求解方法，一个是基于 Levenberg-Marquardt 算法的非线性优化求解。

第3章 基于深度学习的点云配准

本章将结合深度学习，对点云配准进行研究。首先介绍深度学习中点云配准常用的公开数据集，然后介绍点云配准的评价指标并推广到多实例点云配准，最后介绍点云配准相关的深度学习技术。

3.1 标准数据集

本节将介绍用于点云配准的标准数据集。在评估不同指标的性能时，我们需要用到不同的公开数据集。点云配准任务的数据集大致可以分为仿真数据集和真实数据集。仿真数据集中的物体对象是完整的三维模型，比如 ModelNet40^[56]。而真实数据集中，因为存在遮挡、测量误差、相干光干扰、背景影响等问题，往往会有缺失、噪声、过密/稀疏等问题，比如 Scan2CAD^[57] 和 ShapeNet^[58]。下表 3-1 列出了常用的点云配准数据集。

表 3-1 常见的点云信息处理任务数据集列表

数据集名称	年份	场景数量	类别数量	训练集	测试集	类型	数据结构
McGill Benchmark ^[59]	2008	456	19	304	152	仿真场景	三角网格
Sydney Urban Objects ^[60]	2013	588	14	-	-	真实场景	点云
ModelNet10 ^[61]	2015	4899	10	3991	605	仿真场景	三角网格
ModelNet40 ^[56]	2015	12311	40	9843	2468	仿真场景	三角网格
ShapeNet ^[58]	2015	51190	55	-	-	仿真场景	三角网格
ScanNet ^[62]	2017	12283	17	9677	2606	真实场景	RGB-D
ScanObjectNN ^[63]	2016	2902	15	2321	581	真实场景	三角网格
S3DIS ^[64]	2017	271	13	-	-	真实场景	点云
7Scenes ^[65]	2013	7	1	26000	17000	真实场景	RGB-D
Scan2CAD ^[57]	2019	1512	1	1209	303	真实场景	点云

在本次任务中，我们使用 ModelNet40、ShapeNet、Scan2CAD 组成多实例点云配准仿真数据集进行训练和测试。所以，我们对这 3 个数据集和多实例点云配准数据集进行了详细的介绍。

3.1.1 ModelNet40

ModelNet 是一个用于三维物体识别和检索的大规模数据集。它包含两个子集：ModelNet10 和 ModelNet40。ModelNet10 包含 10 个类别的 4,899 个模型，而 ModelNet40 包含 40 个类别的 12,311 个模型。ModelNet 数据集中的模型主要是合成的 CAD 模型，并以网格 (mesh) 的形式呈现。本文主要使用 ModelNet40 数据集。

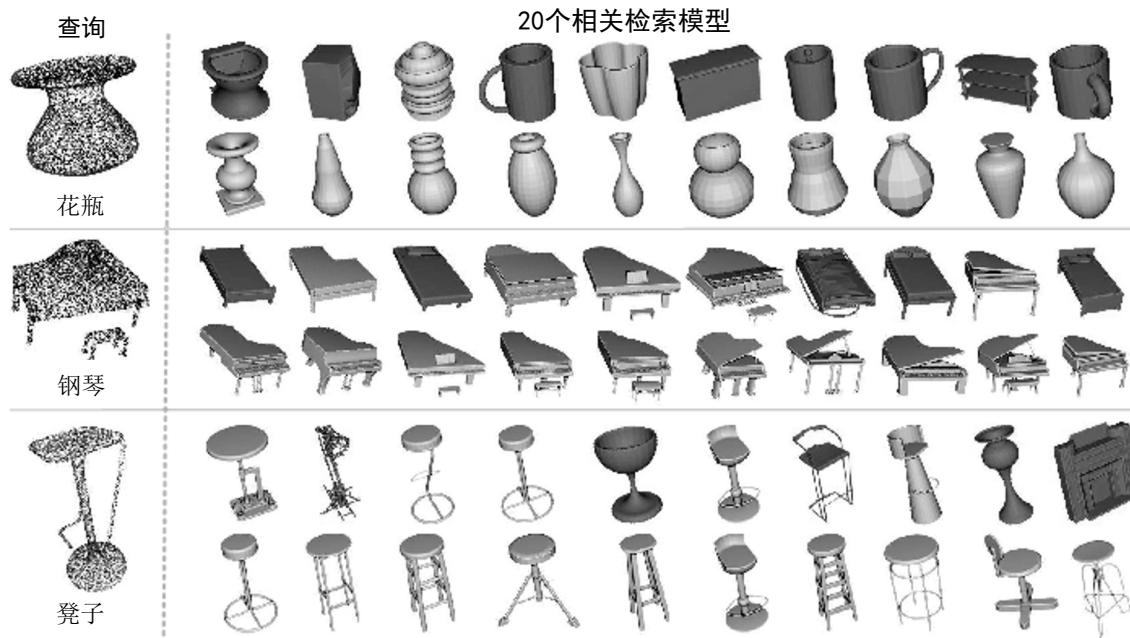


图 3-1 ModelNet40 数据集部分可视化结果

ModelNet40 主要涵盖了各种常见物体，如办公桌、椅子、沙发、书架、飞机、自行车、花盆等。其中例如杯子、桌子、飞机等这种具有规则对称结构的模型，对于位姿回归、点云配准来说是比较困难的，容易因为对称轴而产生歧义。对于人、花这样具有复杂结构的模型，也很难准确识别模型的关键信息，在点云处理中具有一定的挑战性。ModelNet40 数据集部分可视化结果如图 3-1 所示。

3.1.2 ShapeNet

ShapeNet 数据集是一个大规模的三维模型数据库，其目的是为计算机视觉和图形学领域的研究者提供一个丰富且多样化的数据源。该数据集涵盖了许多不同的物体类别，包括家具、交通工具、家电等。ShapeNet 数据集中的模型具有详细的几何形状和丰富的语义标注，这使得它在多种三维任务中都具有很高的实用价值。

ShapeNet 数据集包含了 55 个类别，共有 51,190 个三维模型。这些模型主要来源于网上的 CAD 模型库，经过清洗和处理后整合成一个统一的数据集。数据集中的每个模型都以网格 (mesh) 格式存储，并附有与物体相关的语义信息，如类别标签、实例标签等。此外，ShapeNet 数据集还包含了一些额外的元数据，如模型的尺寸、位置和方向等。部分可视化结果如图 3-2 (a) 所示。

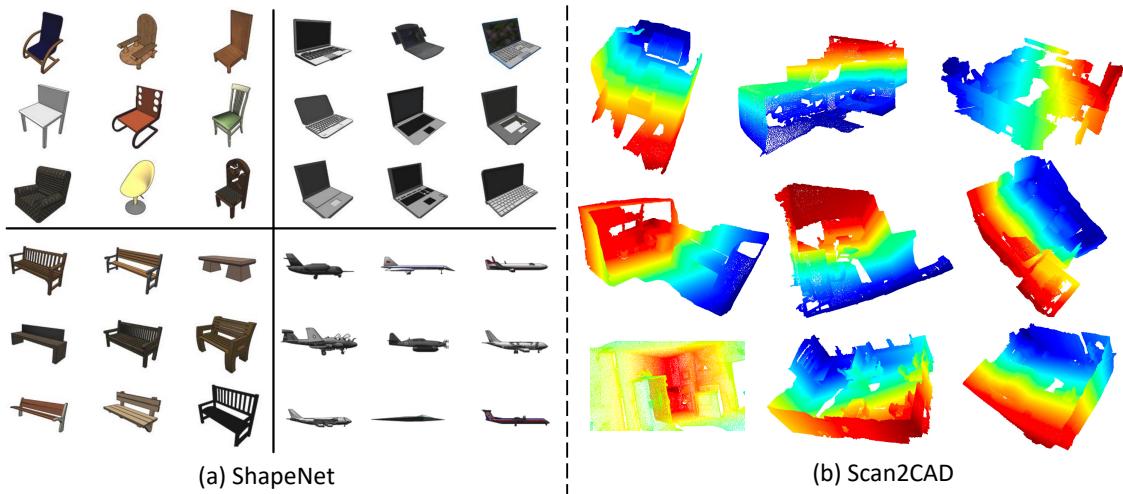


图 3-2 ShapeNet 数据集和 Scan2CAD 数据集部分可视化结果

3.1.3 Scan2CAD

Scan2CAD 数据集是一个用于场景理解和 CAD 模型对齐的大规模数据集。该数据集的目标是将真实场景的三维扫描与合成的 CAD 模型相关联。Scan2CAD 数据集包含了从 Matterport3D^[66] 和 ScanNet^[62] 等公共数据集中的扫描数据，与来自 ShapeNet 等数据集的 CAD 模型。这些扫描和 CAD 模型之间的对应关系是通过专家注释来确定的。

Scan2CAD 数据集中的扫描数据包含了多种室内场景，如住宅、办公室和公共场所等。这些场景包括了各种物体，如家具、装饰品和建筑元素。与此同时，CAD 模型包含了各种详细的几何和拓扑结构。部分可视化结果如图 3-2 (b) 所示。

3.1.4 多实例点云配准仿真数据集

我们的合成数据集是从 ModelNet40 构建的，它包括来自 40 个类别的 12311 个网格化 CAD 模型。为了构建我们的合成数据集，对于每个模型，我们从中均匀地下采样 1024 个点以形成源点云，然后将其旋转和平移 5-10 次以生成多个实例。这些实例与噪声点混合，形成如图 3 所示的目标点云。沿每个轴的旋转在 $[0, 180^\circ]$ 内均匀采样，平移在 $[0, 5]$ 内。我们为合成数据集生成的输入对应关系是通过将真实对应关系与异常值混合而生成的。我们将每个实例的内点比率控制在约 2%。我们使用 12311 个模型生成了 12311 个这样的合成源-目标点云对。我们使用 9843 对进行训练，2468 对进行测试。我们在训练集中随机留出 10% 的对作为验证集。

3.1.5 多实例点云配准真实数据集

我们的真实数据集是 Scan2CAD，它利用 ShapeNet 和 ScanNet 构建而成。此数据集使用 ShapeNet 中的 CAD 模型替换真实扫描场景中的点云，并提供精确的注释，包括模型类别、旋转和平移等。数据集提供了 1506 个带注释的场景，每个场景至少包含一个实例类别。因此，我们充分利用这些注释，并将包含多种实例的场景分割成多个源-目标点云对，用于多实例配准。通过这种方式，我们得到了 2184 对点云，其中大多数点云在目标点云中包含 2-5 个相同类别的实例。我们按照 7:1:2 的比例将样本划分为训练集、验证集和测试集。我们使用微调的 FCGF^[67] 生成局部特征并通过特征匹配生成假设的对应关系。

表 3-2 FCGF 网络参数设置和预训练

FCGF 网络参数设置和预训练	
模型	RESUNETBN2C
下采样体素大小	2.5cm (0.025)
特征维度	32
预训练数据集	3DMatch
特征归一化	是

表 3-3 FCGF 网络微调参数设置

FCGF 网络微调参数设置	
批量大小	4
学习率	10^{-3}
迭代次数	20
优化器	SGD

这里，我们主要关注如何通过特征匹配生成假设的对应关系。我们使用 FCGF 作为特征提取器，参数设置如表 3-2 所示。FCGF 网络在 3DMatch 数据集上预训练，然后使用表 3-3 中的参数设置进行微调。微调过的 FCGF 提取 L2 归一化的局部特征 $F_{local}^X = \{f_{local}^{x_i} \in \mathbb{R}^{32} | i = 1, \dots, |X|\}$ ，用于源点云 X 和 $F_{local}^Y = \{f_{local}^{y_i} \in \mathbb{R}^{32} | i = 1, \dots, |Y|\}$ ，用于目标点云 Y ，其中 $|X|$ 和 $|Y|$ 分别表示源点云和目标点云中的点数。给定目标点云 Y 中的每个点 y_j ，我们找到满足 $i = \arg \max_i \langle f_{local}^{y_j}, f_{local}^{x_i} \rangle$ 的源点云 X 中的点 x_i 以构建对应关系 (x_i, y_j) ，其中 $\langle f_{local}^{y_j}, f_{local}^{x_i} \rangle$ 是两点特征之间的余弦相似度。这样，我们获得 $|Y|$ 个对应关系，我们将余弦相似度 $\langle f_{local}^{x_i}, f_{local}^{y_j} \rangle$ 定义为对应关系 (x_i, y_j) 的显著性得分。我们选择显著性得分最大的 K 个对应关系，然后将这些 K 个对应关系随机下采样为 N 个对应关系作为输入的假设对应关系。这里，我们将 K 设置为 10000，以使输入对应关系尽可能覆盖更多实例。

3.2 点云配准评价指标

对于配准任务，评估指标可以主要分为两类，一类是用于特征提取与匹配的评价指标，另一类是用于配准结果运动参数评价的指标。

3.2.1 基于特征提取与匹配的评价指标

3.2.1.1 特征匹配召回率 (Feature Matching Recall)

特征匹配召回率衡量了匹配算法在正确提取匹配点对数量和所有点对数量之间的比率，反映了匹配算法的查全率。从数学角度来看，特征匹配召回率可以表示为：

$$R_{fa} = \frac{1}{n} \sum_{s=1}^n \mathbb{1} \left(\frac{1}{|\omega|} \sum_{(i,j) \in \omega} (\|T\mathbf{p}_i - \mathbf{q}_j\| < \tau_1) > \tau_2 \right) \quad (3-1)$$

其中 n 是所有点对的数量， ω 是匹配点对 (\mathbf{p}, \mathbf{q}) 之间的对应关系集合， $\mathbf{p} = (x_p, y_p, z_p)$ ， $\mathbf{q} = (x_q, y_q, z_q)$ ， $T \in SE(3)$ 是真实姿态变换矩阵。此外， τ_1 是内部距离阈值， τ_2 是内部召回率阈值。

3.2.1.2 点云配准召回率 (Point Cloud Registration Recall)

配准召回率衡量了在具有真实姿态变换且存在重叠部分的两组点云中，有多少重叠部分的点云组可以通过匹配算法被正确恢复。具体而言，配准召回率使用如下误差矩阵来定义真阳性：

$$E = \sqrt{\frac{1}{|\omega^*|} \sum_{(\mathbf{p}^*, \mathbf{q}^*) \in \omega^*} \|\hat{T}_{i,j}\mathbf{p}^* - \mathbf{q}^*\|^2} < \tau_3 \quad (3-2)$$

其中， ω^* 是重叠部分一组匹配点对 $(\mathbf{p}^*, \mathbf{q}^*)$ 之间的对应关系集合， $\mathbf{p}^* = (x_p^*, y_p^*, z_p^*)$ ， $\mathbf{q}^* = (x_q^*, y_q^*, z_q^*)$ 。对于重叠部分， τ_3 是用于判断匹配点对是否正确的阈值。

3.2.2 基于刚体运动的评价指标

3.2.2.1 投影的均方根误差

投影的均方根误差 (RMSE Projection) 是在应用变换之后，计算点到点投影误差的平均值。计算公式为：

$$\text{RMSE}(\mathbf{p}_i) = \frac{1}{n} \sum_{i=1}^n \sqrt{(\mathbf{p}'_i - \mathbf{p}_i)^2}, \quad (3-3)$$

其中 n 是点云中的点数， \mathbf{p}'_i 是应用变换后的点， \mathbf{p}_i 是原始点。

3.2.2.2 变换的均方根误差

变换的均方根误差 (RMSE Transformation) 代表估计的变换 $\hat{\mathbf{T}}_s$ 和真实变换 \mathbf{T}_s^* 之间的均方根误差。计算公式为：

$$\text{RMSE}(\mathbf{T}_s) = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{\mathbf{T}}_s - \mathbf{T}_s^*)^2}, \quad (3-4)$$

3.2.2.3 相对旋转误差

我们定义相对旋转误差 (Relative Rotation Error) 为真实变换 \mathbf{T}_s^* 和估计变换 $\hat{\mathbf{T}}_s$ 之间的旋转误差。计算公式为：

$$\text{RRE}_s = \arccos \left(\frac{\text{tr}(\hat{\mathbf{R}}_s \mathbf{R}_s^*) - 1}{2} \right), \quad (3-5)$$

其中 tr 表示矩阵的迹 (对角线元素之和)， $\hat{\mathbf{R}}_s$ 和 \mathbf{R}_s^* 分别表示估计变换和真实变换的旋转矩阵。

3.2.2.4 相对平移误差

我们定义相对平移误差 (Relative Translation Error) 为真实变换 \mathbf{T}_s^* 和估计变换 $\hat{\mathbf{T}}_s$ 之间的平移误差。计算公式为：

$$\text{RTE}_s = \|\mathbf{t}_s^* - \hat{\mathbf{t}}_s\|, \quad (3-6)$$

其中 \mathbf{t}_s^* 和 $\hat{\mathbf{t}}_s$ 分别表示真实变换和估计变换的平移向量。

3.2.3 多实例点云配准评价指标

用于评估多实例点云配准性能的三个指标分别是平均命中召回率、平均命中精度和平均命中 F1 分数。

3.2.3.1 平均命中召回率 (Mean Hit Recall)

平均命中召回率衡量了正确恢复的点云对数占所有点云对数的比例。平均命中召回率 (Mean Hit Recall) 在两个已配准的点云之间的定义如下：

$$MHR = \frac{1}{S} \sum_{s=1}^S I_s \quad (3-7)$$

其中 $I_s = 0, 1$ 表示一个真实估计对是否为“命中”。具体而言，

$$I_s = I(RRE_s < \tau_r) \times I(RTE_s < \tau_t) \quad (3-8)$$

其中 $I(\cdot) = 0, 1$ 表示一个指示函数。RRE_s 和 RTE_s 是第 s 个真实估计对的相对旋转误差和相对平移误差。两个阈值 τ_r 和 τ_t 分别设置为 20° 和 $0.5m$ 。通过对所有点云对的 MHR 求平均值，得到最终的平均命中召回率。

3.2.3.2 平均命中精确度 (Mean Hit Precision)

两个已配准的点云之间的平均命中精确度定义如下：

$$MHP = \frac{1}{M} \sum_{s=1}^S I_s \quad (3-9)$$

通过对所有点云对的 MHP 求平均值，得到最终的平均命中精确度。

3.2.3.3 平均命中 F1 (Mean Hit F1)

两个已配准的点云之间的平均命中 F1 定义如下：

$$MHF1 = \frac{2 \times MHP \times MHR}{MHP + MHR} \quad (3-10)$$

通过对所有点云对的 MHF1 求平均值，得到最终的平均命中 F1。

3.3 点云处理与配准相关深度学习技术

3.3.1 深度学习概述

在过去的十年中，深度学习在计算机视觉、自然语言处理等各个领域取得了重要进展^[68]。随着激光雷达、RGB-D 相机和其他三维扫描设备获取的 3D 数据日益增多，深度学习技术逐渐应用于点云处理任务，包括点云配准^[69-70]。点云配准是 3D 计算机视觉中的一个基本问题，目的是找到将两个点云精确对齐的最优变换。

近年来，针对点云配准的深度学习方法已经在自动学习区分性特征和从大规模数据集中学习鲁棒匹配策略方面展示出优于传统方法的性能^[71-72]。这些方法利用各种类型的网络架构，例如 PointNet^[5]、PointNet++^[38] 和 DGCNN^[69]，有效地处理点云数据的非结构化和无序特性。

训练深度神经网络涉及使用反向传播算法，该算法通过应用链式法则计算损失函数相对于每个权重的梯度。使用基于梯度的优化方法（如随机梯度下降（SGD）、Adam^[73] 或 MoCo^[74]）更新权重。为了防止过拟合，采用了如 dropout 等正则化技术，即在训练过程中随机将一层中的一部分神经元设置为零。这种策略迫使网络学习更鲁棒的特征，提高了对未见数据的泛化能力。

$$L(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^N L_i(\mathbf{W}) + \lambda \sum_{l=1}^L \|\mathbf{W}^{(l)}\|^2 \quad (3-11)$$

在上述方程中， $L(\mathbf{W})$ 表示总损失， $L_i(\mathbf{W})$ 表示第 i^{th} 个样本的损失， $\mathbf{W}^{(l)}$ 表示第 l 层的权重， L 是总层数， λ 是正则化参数， $\|\cdot\|^2$ 表示平方 Frobenius 范数。

为了方便设计、训练和部署深度学习模型，开发了几种深度学习框架，包括 TensorFlow^[75]、PyTorch^[76]、PaddlePaddle^[77] 和 MXNet^[78]。这些框架提供了神经网络层、优化算法和其他实用工具的高效实现，使设计、训练和部署深度学习模型变得更加容易。此外，它们还支持使用 GPU (torch.cuda) 或专用加速器（如 TPU）进行硬件加速，以加快训练过程。

3.3.2 多层感知机

多层感知器 (Multilayer Perceptron, MLP) 是一种广泛应用于各种机器学习任务的前馈神经网络^[68]。MLP 的基本结构包括输入层、一个或多个隐藏层和输出层。每一层由若干个神经元组成，相邻层的神经元之间通过权重连接。通常，MLP 采用全连

接结构，即每个神经元都与上一层和下一层的所有神经元连接。

在训练 MLP 时，需要先进行前向传播计算，即从输入层到输出层计算预测值。接下来，通过反向传播算法计算损失函数对权重的梯度。最后，使用梯度下降或其变体更新权重。

考虑一个简单的两层 MLP，其输入层有 n 个神经元，输出层有 m 个神经元。对于输入向量 $\mathbf{x} \in \mathbb{R}^n$ 和权重矩阵 $\mathbf{W} \in \mathbb{R}^{m \times n}$ ，MLP 的输出为：

$$\mathbf{y} = f(\mathbf{W}\mathbf{x} + \mathbf{b}), \quad (3-12)$$

其中 $\mathbf{b} \in \mathbb{R}^m$ 是偏置向量， f 是激活函数。在实际应用中，激活函数通常为非线性函数，如 Sigmoid、ReLU 或 Tanh。这些非线性激活函数使得 MLP 能够捕捉复杂的非线性关系。

现在我们考虑一个具有 L 个隐藏层的 MLP，设第 l 层的权重矩阵为 $\mathbf{W}^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ ，偏置向量为 $\mathbf{b}^{(l)} \in \mathbb{R}^{n_l}$ ，激活函数为 $f^{(l)}$ 。前向传播过程可表示为：

$$\mathbf{a}^{(l)} = f^{(l)}(\mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}), \quad (3-13)$$

其中 $\mathbf{a}^{(0)} = \mathbf{x}$, $\mathbf{a}^{(L)} = \mathbf{y}$ 。对于损失函数 $L(\mathbf{y}, \mathbf{t})$ ，其中 \mathbf{t} 是目标输出，我们使用梯度下降方法优化权重和偏置：

$$\mathbf{W}^{(l)} \leftarrow \mathbf{W}^{(l)} - \alpha \frac{\partial L}{\partial \mathbf{W}^{(l)}}, \quad (3-14)$$

$$\mathbf{b}^{(l)} \leftarrow \mathbf{b}^{(l)} - \alpha \frac{\partial L}{\partial \mathbf{b}^{(l)}}, \quad (3-15)$$

其中 α 是学习率。

为了计算梯度，我们使用反向传播算法。首先，计算输出层关于损失函数的梯度：

$$\delta^{(L)} = \frac{\partial L}{\partial \mathbf{a}^{(L)}} \odot f'^{(L)}(\mathbf{W}^{(L)}\mathbf{a}^{(L-1)} + \mathbf{b}^{(L)}), \quad (3-16)$$

其中 \odot 表示哈达玛积（元素对应相乘）， $f'^{(L)}$ 是激活函数的导数。

然后，我们从输出层向输入层反向计算梯度：

$$\boldsymbol{\delta}^{(l)} = \left(\mathbf{W}^{(l+1)}\right)^T \boldsymbol{\delta}^{(l+1)} \odot f'(l)(\mathbf{W}^{(l)} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}). \quad (3-17)$$

最后，我们计算损失函数关于权重和偏置的梯度：

$$\frac{\partial L}{\partial \mathbf{W}^{(l)}} = \boldsymbol{\delta}^{(l)} (\mathbf{a}^{(l-1)})^T, \quad (3-18)$$

$$\frac{\partial L}{\partial \mathbf{b}^{(l)}} = \boldsymbol{\delta}^{(l)}. \quad (3-19)$$

在实际应用中，为了提高训练速度和泛化能力，我们通常采用 mini-batch 随机梯度下降或其变体（如 Adam、RMSprop 等）进行训练。此外，我们还可以使用正则化技术（如 L1、L2 正则化、Dropout 等）防止过拟合。在深度学习框架（如 TensorFlow、PyTorch、PaddlePaddle 和 MXNet 等）的支持下，我们可以方便地实现和训练 MLP 模型^[68]。

MLP 是深度学习领域的一个基本概念，尽管现在有许多更复杂的神经网络结构（如卷积神经网络、循环神经网络和 Transformer 等）已经在各种任务上表现出更好的性能，但 MLP 仍然是一个重要的基础知识。通过学习 MLP 的基本原理，我们可以更好地理解神经网络的工作原理，从而有助于设计更复杂的模型和解决更高级的问题。

在实际应用中，MLP 可以处理各种类型的数据，如图像、文本、语音和其他结构化或非结构化数据。由于 MLP 可以近似任意连续函数，因此具有很强的表达能力。然而，MLP 也存在一些局限性。首先，MLP 的参数数量随着层数和每层神经元数量的增加而快速增长，这可能导致过拟合和计算效率低下的问题。其次，MLP 对输入数据的缩放和平移具有较强的敏感性，因此需要对输入数据进行预处理，如归一化。此外，MLP 并不能直接处理变长输入数据和序列数据，需要采用其他神经网络结构，如循环神经网络（RNN）。

尽管如此，MLP 作为一种基础的神经网络结构，在许多任务中仍具有竞争力。例如，在监督学习任务中，MLP 可以用于分类和回归问题；在无监督学习任务中，MLP 可以用于降维和聚类。通过合适的设计和训练策略，MLP 能够在各种应用场景中实现良好的性能。

总之，多层感知器（MLP）是深度学习领域的一个基本概念，作为前馈神经网络的一种，它具有广泛的应用价值。MLP 主要包括输入层、隐藏层和输出层，通过前

向传播、反向传播算法进行训练。

3.3.3 卷积神经网络

卷积神经网络 (CNN) 是一类在各种任务中取得显著成功的深度学习模型，尤其在图像识别、目标检测和自然语言处理等领域。CNN 旨在学习空间特征层次结构，使其特别适用于处理网格状数据 (如图像)，其中局部依赖关系至关重要^[79]。

CNN 的基本构建模块是卷积层。在这个层中，小滤波器 (也称为内核) 应用于输入，计算滤波器和输入之间的局部点积。这个操作生成一个特征映射，对输入和滤波器之间的空间关系进行编码^[3]。滤波器权重在训练过程中学习，使网络能够自动学习任务特定特征。

从数学上讲，CNN 中的卷积运算可以描述为以下形式：

$$y_{i,j} = \sum_m \sum_n x_{i-m,j-n} \cdot w_{m,n}, \quad (3-20)$$

其中 x 是输入， w 是滤波器， y 是输出特征映射， i 、 j 、 m 和 n 是输入和滤波器的空间维度对应的索引。

CNN 的一个关键方面是在整个输入中使用共享权重。这种参数共享显著减少了可训练参数的数量，使模型比全连接网络更具计算效率且不容易过拟合。此外，它使网络能够学习平移不变特征，因为相同的滤波器应用于输入的不同空间位置。

CNN 中的另一个重要组件是池化层，用于减少特征映射的空间尺寸。池化层通常在一个或多个卷积层之后应用，以实现局部平移不变性并减少网络的计算复杂性。有不同类型的池化操作，如最大池化、平均池化和全局池化。最大池化是最常用的池化方法，可以定义为：

$$z_{i,j} = \max_{m,n \in P} x_{i+m,j+n}, \quad (3-21)$$

其中 x 是输入特征映射， z 是池化后的输出， P 是池化窗口， i 、 j 、 m 和 n 是输入和池化窗口的空间维度对应的索引。

除了卷积和池化层之外，CNN 通常包括一个或多个全连接层，对学习到的特征进行高级推理。最后一个全连接层的输出通常通过 softmax 激活函数传递，以生成任务特定类别的概率。整个网络使用反向传播和基于梯度的优化算法 (如随机梯度下降

或 Adam) 进行训练。

CNN 已成功应用于广泛的任务，包括图像分类、目标检测、语义分割和自然语言处理。CNN 的成功可以归因于其学习输入数据的分层表示、共享权重和平移不变特征的能力。此外，通过修改其架构和损失函数，CNN 可以轻松地适应不同的任务^[80]。

3.3.4 注意力机制

注意力机制已经成为深度学习模型的重要组成部分，特别是在计算机视觉和点云处理任务中。注意力机制的核心概念是让模型在处理输入数据时，根据上下文信息有选择地关注不同部分，从而提高性能和可解释性。

在计算机视觉领域，注意力机制可以帮助模型更有效地识别和关注图像中的重要区域，以实现更准确的对象识别、语义分割和目标检测。在点云处理任务中，注意力机制可以帮助模型学习点云中的局部和全局结构特征，从而在点云配准、点云分割和分类任务中实现更高的性能。

数学上，注意力机制可以通过以下方式进行描述：

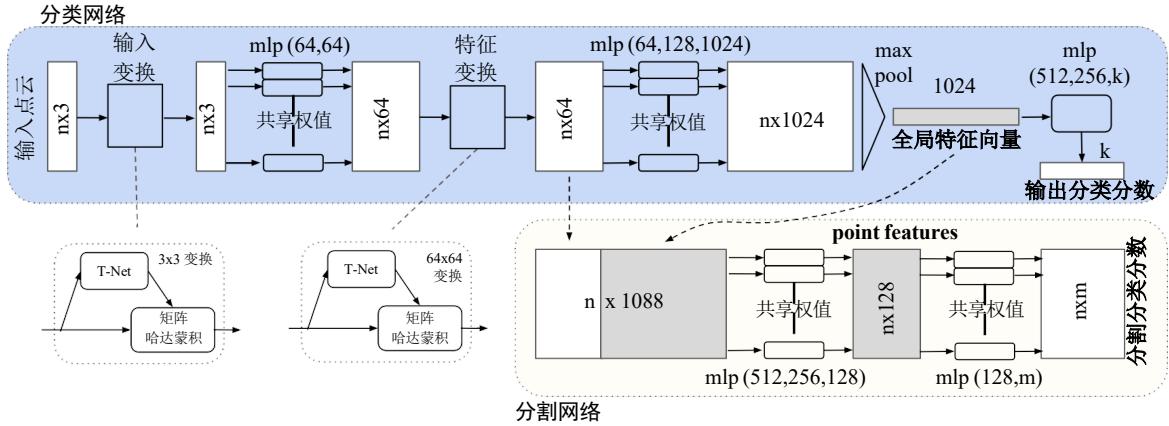
$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \quad (3-22)$$

其中 \mathbf{Q} 、 \mathbf{K} 和 \mathbf{V} 分别表示查询 (Query)、键 (Key) 和值 (Value) 矩阵， d_k 是键和查询向量的维度。这个计算过程可以解释为：查询向量与键向量的点积表示它们之间的相似性，通过应用 softmax 函数将这些相似性归一化为概率分布，然后使用这些概率分布加权值向量，得到最终的输出。

在计算机视觉和点云处理任务中，注意力机制通常与卷积神经网络 (CNN) 或图神经网络 (GNN) 结合使用，以捕获局部和全局上下文信息。这些组合模型在大量基准数据集上表现出卓越的性能，证明了注意力机制在这些领域的有效性。

3.4 基于深度学习的点云配准方法

本节主要阐述主要的基于深度学习的点云配准方法作为基准线，主要阐述 PointNet^[5,38] 系列框架、PREDATOR^[14] 描述子和 TEASER^[21] 配准方法。


 图 3-3 PointNet^[5] 网络结构图

3.4.1 PointNet & PointNet++ 网络结构简述

3.4.1.1 PointNet 网络结构简述

点云处理任务中，我们首先可以朴素地想到直接用多层感知机^[68] 对作为点云的输入，但是这样的方法对于 N 个点云来说，需要 $N \times 3$ 大小的输入维度，点云数量多时，对于多层感知机过于庞大。所以 PointNet 提出，对于输入点云，采用共享权值的 MLP 进行特征提取

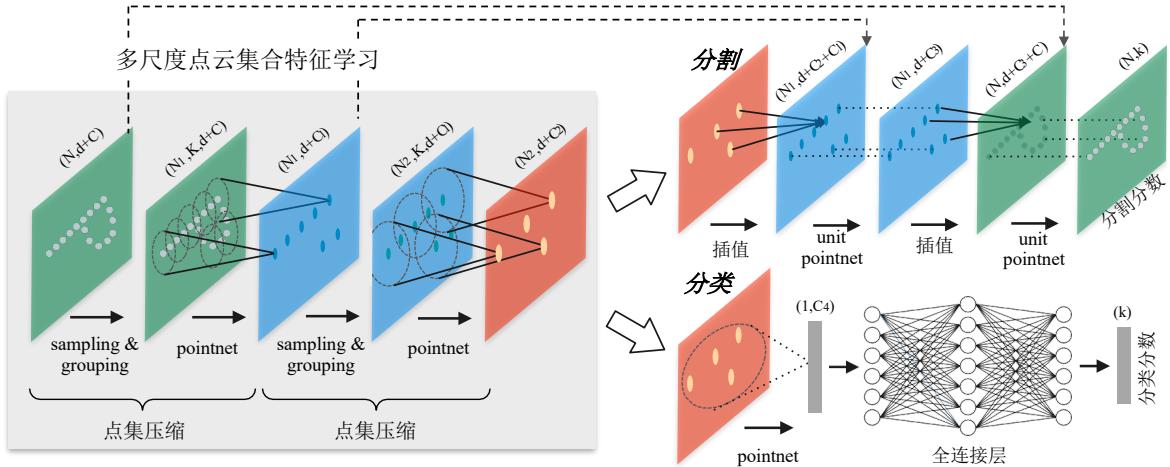
$$\mathbf{f}_{\mathbf{p}_i} = f(\theta; \mathbf{p}_i) \quad (3-23)$$

然后对特征进行特征变换，对于变换后的特征再次进行升维操作，最终通过 n 个点云数据，得到 $n \times 1024$ 的特征图。将特征图进行最大池化操作，得到全局特征向量。

对于分类任务，每一个点云对应一个类别，所以全局特征可以直接用来判断类别信息，通过一个 MLP 输出最终的分类分数，达到分类的效果。对于点云分割任务，全局特征不足以描述分割的模式，需要加入局部特征。所以将全局特征向量和 $\mathbf{f}_{\mathbf{p}_i}$ 进行拼接操作，得到局部 + 全局特征，通过 MLP 输出分割的分类分数。

3.4.1.2 PointNet++ 网络结构简述

PointNet++^[38] 则在 PointNet 上做了改进。前文提到的 PointNet 在点云处理中有一个重大缺陷，即点云是具有丰富的表面几何特征的数据格式，也就是说，每一个点都和周围的点产生了交互，但是 PointNet 仅仅提取出了点的特征和全局特征，缺


 图 3-4 PointNet++^[38]网络结构图

失了提取局部特征这一过程。这在训练神经网络的时候，往往会导致模型的泛化能力有限^[38]。所以 PointNet++ 对此作出了调整，在 PointNet 的基础上提去了不同尺度的局部特征，图示如图 3-4。

这个方法是受到 CNN 的启发，在 3D 点集中找到具有相似结构的子区域，和对应的区域特征 embedding，提出了 Sampling 和 Grouping 来整合局部邻域的点云，提取局部特征，代码如3.1。其中，`farthest point sample` 为本段的核心算法，即最远端采样算法来实现从 N 个点中抽取采样 N' 个点。

- 最远端采样算法 (FPS) 的流程如下：
1. 用蒙特卡洛方法筛选初始点，即第 0 次采样点；
 2. 计算未采样点与已采样点之间的欧几里得或者推土机距离，将距离最大的点加入已采样点集；
 3. 更新距离函数，重复 1-3，直到获得了目标数量的采样点；

```

1  def sample_and_group(npoin, radius, nsample, xyz, points, knn=False,
2      use_xyz=True):
3      """
4          Input:
5              npoin: int32
6              radius: float32
7              nsample: int32
8              xyz: (batch_size, ndataset, 3) TF tensor
9              points: (batch_size, ndataset, channel) TF tensor, if None will just use

```

```

xyz as points
9      knn: bool, if True use kNN instead of radius search
10     use_xyz: bool, if True concat XYZ with local point features, otherwise
just use point features
11     Output:
12         new_xyz: (batch_size, npoint, 3) TF tensor
13         new_points: (batch_size, npoint, nsample, 3+channel) TF tensor
14         idx: (batch_size, npoint, nsample) TF tensor, indices of local points as
in ndataset points
15         grouped_xyz: (batch_size, npoint, nsample, 3) TF tensor, normalized
point XYZs
16             (subtracted by seed point XYZ) in local regions
17     """
18
19     new_xyz = gather_point(xyz, farthest_point_sample(npoint, xyz)) # (
batch_size, npoint, 3)
20
21     if knn:
22
23         _,idx = knn_point(nsample, xyz, new_xyz)
24     else:
25
26         idx, pts_cnt = query_ball_point(radius, nsample, xyz, new_xyz)
27         grouped_xyz = group_point(xyz, idx) # (batch_size, npoint, nsample, 3)
28         grouped_xyz -= tf.tile(tf.expand_dims(new_xyz, 2), [1,1,nsample,1]) #
translation normalization
29
30     if points is not None:
31
32         grouped_points = group_point(points, idx) # (batch_size, npoint, nsample
, channel)
33
34         if use_xyz:
35
36             new_points = tf.concat([grouped_xyz, grouped_points], axis=-1) # (
batch_size, npoint, nample, 3+channel)
37
38         else:
39
40             new_points = grouped_points
41
42     else:
43
44         new_points = grouped_xyz
45
46     return new_xyz, new_points, idx, grouped_xyz

```

代码 3.1: 最远端采样算法

上面介绍了如何从采样点到局部特征提取的过程，但是在实践中，提取到的不同尺度的特征如何融合对于下游任务的影响也同样很大。PointNet++ 提出了两种特征融合方式 Multi-Scale Grouping (MSG) 和 Multi-Resolution Grouping (MRG)。如图3-

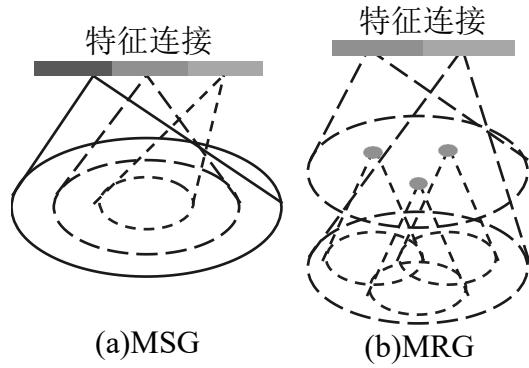


图 3-5 多尺度特征融合方法示意图

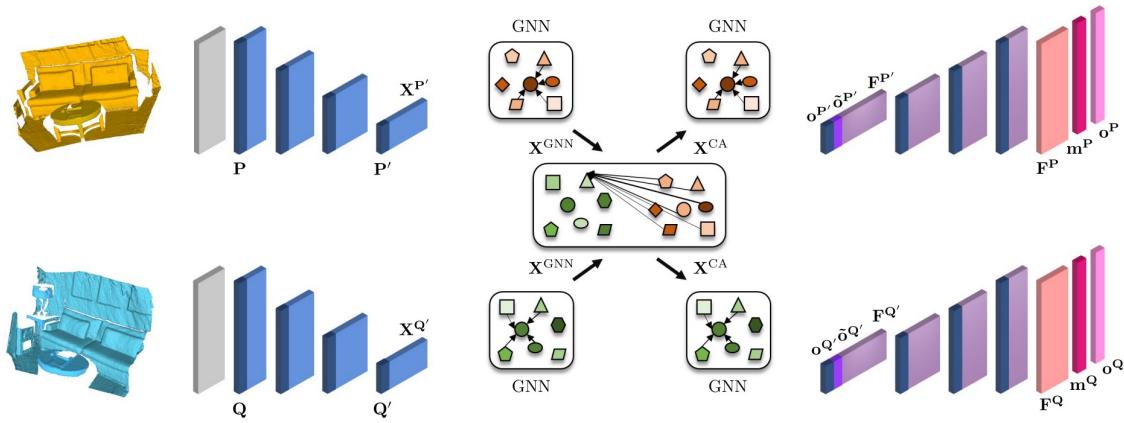


图 3-6 PREDATOR 的网络结构。体素化的点云 P 和 Q 被送入编码器，它提取出超点 P' 和 Q' 及其潜在特征 $\mathbf{X}_{P'}$ 和 $\mathbf{X}_{Q'}$ 。重叠注意力模块在一系列 (GNN) 和交叉注意力 (Cross Attention) 块中，更新特征的共同语境信息，并将它们投影到重叠度 $\mathbf{o}_{P'}$ 、 $\mathbf{o}_{Q'}$ 和交叉重叠度 $\tilde{\mathbf{o}}_{P'}$ 、 $\tilde{\mathbf{o}}_{Q'}$ 的得分。最后，解码器将条件特征和重叠度得分转换为每点特征描述符 \mathbf{F}_P 、 \mathbf{F}_Q ，重叠度得分 \mathbf{o}_P 、 \mathbf{o}_Q ，以及匹配度得分 \mathbf{m}_P 、 \mathbf{m}_Q 。

5(a)、(b) 所示。

对 MSG 方法，具体操作是对不同半径的子区域进行特征提取后进行特征堆叠，特征提取过程还是采用了 PointNet，所以对于不同的子区域进行参数计算的时候，使用了不同的网络，计算量过大。对于 MRG 方法来说，使用两个 PointNet 网络对连续的两层分别做特征提取与聚合，然后再进行特征拼接，可以大大减少参数计算量。

由于 PointNet 系列网络架构的经典性和鲁棒性，后面的点云处理工作都会借鉴 PointNet 系列的架构和方法，比如 Geometric Transformer^[11]，采用了 PointNet 相同的端到端架构，实现了高质量的点云配准；还有 PPFNet^[9]、PPF-FoldingNet^[10]、CapsuleNet^[8] 等经典网络，都对该系列进行了借鉴。

3.4.2 PREDATOR 架构简述

PREDATOR^[14]是一种低重叠度的三维点云配准方法，如图 3-6 所示，PREDATOR 的架构可以分解为三个主要模块：

1. 将两个点云编码为更小的超点集合和相关的潜在特征编码，权重共享；
2. 重叠注意力模块（在瓶颈中）提取两个点云的特征编码之间的共同语境信息，并为每个超点分配两个重叠得分，这两个得分量化了超点本身和其软对应点位于两个输入之间的重叠区域的可能性；
3. 将相互调节的瓶颈表示解码为点对点描述符以及精化的每点重叠和匹配度得分。

3.4.2.1 特征编码

使用大小为 V 的体素网格滤波器对原始点云进行下采样，使得 \mathbf{P} 和 \mathbf{Q} 具有相当均匀的点密度。在共享编码器中，一系列类似于 ResNet 的块和跨步卷积将原始点聚合到超点 $\mathbf{P}' \in \mathbf{R}^{N' \times 3}$ 和 $\mathbf{Q}' \in \mathbf{R}^{M' \times 3}$ ，以及相关特征 $\mathbf{X}_{\mathbf{P}'} \in \mathbf{R}^{N' \times b}$ 和 $\mathbf{X}_{\mathbf{Q}'} \in \mathbf{R}^{M' \times b}$ 。需要注意的是，超点对应于固定的接收场，因此它们的数量取决于输入点云的空间范围，并且对于两个输入可能是不同的。

首先，我们使用大小为 V 的体素网格滤波器对原始点云进行下采样，使得 \mathbf{P} 和 \mathbf{Q} 具有相当均匀的点密度。在共享编码器中，一系列类似于 ResNet 的块和跨步卷积将原始点聚合到超点 $\mathbf{P}' \in \mathbf{R}^{N' \times 3}$ 和 $\mathbf{Q}' \in \mathbf{R}^{M' \times 3}$ ，以及相关特征 $\mathbf{X}_{\mathbf{P}'} \in \mathbf{R}^{N' \times b}$ 和 $\mathbf{X}_{\mathbf{Q}'} \in \mathbf{R}^{M' \times b}$ 。需要注意的是，超点对应于固定的接收场，因此它们的数量取决于输入点云的空间范围，并且对于两个输入可能是不同的。

到目前为止，瓶颈中的特征 $\mathbf{X}_{\mathbf{P}'}$ 和 $\mathbf{X}_{\mathbf{Q}'}$ 编码了两个点云的几何形状和上下文。但是， $\mathbf{X}_{\mathbf{P}'}$ 对点云 \mathbf{Q} 一无所知，反之亦然。为了推理它们各自的重叠区域，需要进行一些交叉对话。我们认为，在瓶颈中的超点级别添加这种交叉对话是有意义的，就像人类操作员首先会粗略地了解整体形状以确定可能的重叠区域，然后才会在这些区域中识别出精确的特征点。

图卷积神经网络：在连接两个特征编码之前，我们首先使用图神经网络 (GNN) 进一步聚合并强化它们各自的上下文关系。首先，使用 k-NN 方法将 \mathbf{P}' 中的超点链接成一个欧几里得空间图。设 $\mathbf{x}_i \in \mathbf{R}^b$ 表示超点 \mathbf{p}'_i 的特征编码， $(i, j) \in \mathbf{E}$ 表示超

点 \mathbf{p}'_i 和 \mathbf{p}'_j 之间的图边。然后，编码器特征按照以下方式迭代更新：

$$(k+1)\mathbf{x}_i = \max_{(i,j) \in E} h_\theta [\text{cat} [(k)\mathbf{x}_i, (k)\mathbf{x}_j - (k)\mathbf{x}_i]] \quad (3-24)$$

这个更新操作执行两次，参数 θ 并未共享，最终的 GNN 特征 $\mathbf{x}_{GNN_i} \in \mathbf{R}_b^d$ 如下所得：

$$\mathbf{x}_{GNN_i} = h_\theta (\text{cat} [{}^{(0)}\mathbf{x}_i, {}^{(1)}\mathbf{x}_i, {}^{(2)}\mathbf{x}_i]) \quad (3-25)$$

其中， $h_\theta(\cdot)$ 表示一个线性层，后面跟着实例归一化和 LeakyReLU 激活函数。

上述描述的 GNN 是针对点云 \mathbf{P}' 的。对于 \mathbf{Q}' 的 GNN 是相同的。

3.4.2.2 交叉注意力机制

首先，将点云 \mathbf{P}' 中的每个超点连接到 \mathbf{Q}' 中的所有超点，形成一个二分图。受到 Transformer 架构的启发，我们使用向量值查询 $\mathbf{s}_i \in \mathbf{R}^b$ 来根据其键 $\mathbf{q}_j \in \mathbf{R}^b$ 检索其他超点的值 $\mathbf{v}_j \in \mathbf{R}^b$ ，其中：

$$\begin{aligned} \mathbf{q}_j &= \mathbf{W}_q \mathbf{x}_{GNN_j} \\ \mathbf{v}_j &= \mathbf{W}_v \mathbf{x}_{GNN_j} \\ \mathbf{s}_i &= \mathbf{W}_s \mathbf{x}_{GNN_i} \end{aligned} \quad (3-26)$$

这里， \mathbf{W}_q , \mathbf{W}_v 和 \mathbf{W}_s 是可学习的权重矩阵。计算为值的加权平均，

$$\mathbf{m}_{i \leftarrow} = \sum_{j:(i,j) \in E} a_{ij} \mathbf{v}_j \quad (3-27)$$

其中注意力权重 $a_{ij} = \text{softmax}(\mathbf{s}_i^T \mathbf{k}_j / \sqrt{b})$ 。也就是说，为了更新超点 \mathbf{p}'_i ，将该点的查询与所有超点 \mathbf{q}'_j 的键和值相结合。与文献一致，实际上我们使用具有四个并行注意力头的多注意力层。共同上下文特征计算为

$$\mathbf{x}_{CA_i} = \mathbf{x}_{GNN_i} + \text{MLP}(\text{cat}[\mathbf{s}_i, \mathbf{m}_{i \leftarrow}]) \quad (3-28)$$

其中 $\text{MLP}(\cdot)$ 表示一个三层全连接网络，在第一二层后有实例归一化和 ReLU

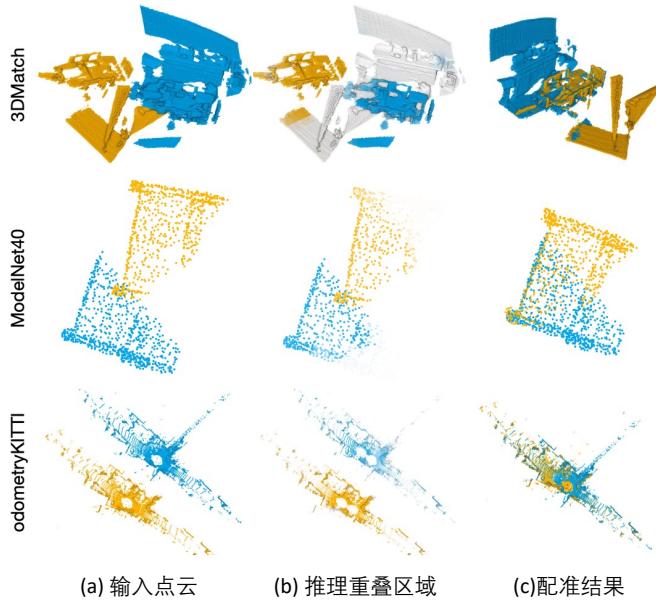


图 3-7 PREDATOR 配准示例结果。

激活函数。相同的交叉注意块也反向应用，使信息在两个方向上流动， $\mathbf{P}' \rightarrow \mathbf{Q}'$ 和 $\mathbf{Q}' \rightarrow \mathbf{P}'$ 。

3.4.2.3 瓶颈点的重叠得分

以上更新共同上下文信息是针对每个超点单独进行的，没有考虑每个点云内的局部上下文。因此，在交叉注意块后，我们使用另一个具有相同架构和底层图(点云内链接)的 GNN 显式更新局部上下文，但参数 θ 是独立的。这产生了最终的潜在特征编码 $\mathbf{F}_{\mathbf{P}'} \in \mathbf{R}^{N' \times b}$ 和 $\mathbf{F}_{\mathbf{Q}'} \in \mathbf{R}^{M' \times b}$ ，现在它们是基于其他点云特征的条件。这些特征被线性投影到重叠分数 $\mathbf{o}_{\mathbf{P}'} \in \mathbf{R}^{N'}$ 和 $\mathbf{o}_{\mathbf{Q}'} \in \mathbf{R}^{M'}$ ，这可以被解释为一个特定超点位于重叠区域的概率。此外，可以计算超点之间的软对应关系，并从对应关系权重预测超点 \mathbf{p}'_i 的交叉重叠分数，即其在 \mathbf{Q}' 的对应点位于重叠区域的概率：

$$\begin{aligned}\tilde{\mathbf{o}}_{\mathbf{P}'i} &:= \mathbf{w}_i^T \mathbf{o}_{\mathbf{Q}'}, \\ \mathbf{w}_{ij} &:= \text{softmax} \left(\frac{1}{t} \langle \mathbf{f}_{\mathbf{P}'i}, \mathbf{f}_{\mathbf{Q}'j} \rangle \right),\end{aligned}\tag{3-29}$$

其中 $\langle \cdot, \cdot \rangle$ 是内积， t 是控制软分配的温度参数。在 $t \rightarrow 0$ 的极限下，公式 3-29 收敛到最近邻硬分配。

3.4.2.4 解码器

我们的解码器从条件特征 $\mathbf{F}_{\mathbf{P}'}$ 开始，将其与重叠分数 $\mathbf{o}_{\mathbf{P}'}$ 和 $\tilde{\mathbf{o}}_{\mathbf{P}'}$ 连接，并输出每点特征描述符 $\mathbf{F}_{\mathbf{P}} \in \mathbf{R}^{N \times 32}$ 和每点的重叠分数和匹配性得分 $\mathbf{o}_{\mathbf{P}}, \mathbf{m}_{\mathbf{P}} \in \mathbf{R}^N$ 。匹配性可以被视为“条件显著性”，量化了给定另一个点云 \mathbf{Q} 中的点（或特征）时，一个点正确匹配的可能性。

解码器的架构将 NN 上采样与线性层相结合，并包括来自相应编码器层的跳跃连接。我们有意将重叠得分和匹配性分开，以便解开一个点是好的/坏的匹配候选的原因：原则上，一个点可以被明确地匹配，但位于重叠区域之外，或者它可以位于重叠区域，但具有模糊的描述符。经验上，我们发现网络学习预测高匹配性主要是在重叠区域的点，这可能反映出用于训练的真实对应关系，总是在重叠区域内。结果如图 3-7 所示。

3.4.2.5 损失函数

PREDATOR 采用端到端的训练，使用三种与真实对应关系相关的损失作为监督。

环形损失：为了监督点特征描述符，我们遵循^[22] 并使用环形损失^[82]，这是更常见的三元损失的一种变体。考虑一对重叠点云 \mathbf{P} 和 \mathbf{Q} ，这次与真实变换对齐。我们首先提取 \mathbf{P} 中的点 $\mathbf{p}_i \in \mathbf{P}_p \subset \mathbf{P}$ ，该点在 \mathbf{Q} 中至少有一个（可能是多个）对应点，其中对应点集 $\mathbf{E}_p(\mathbf{p}_i)$ 定义为 \mathbf{Q} 中在 \mathbf{p}_i 周围半径 r_p 内的点。同样， \mathbf{Q} 中所有在半径 r_s （较大）之外的点形成负面集 $\mathbf{E}_n(\mathbf{p}_i)$ 。然后从 \mathbf{P}_p 中随机抽取 n_p 个点计算环形损失：

$$L_P^c = \frac{1}{n_p} \sum_{i=1}^{n_p} \log \left(1 + \sum_{j \in E_p} e^{\beta_j^p(d_j^i - \Delta_p)} \cdot \sum_{k \in E_n} e^{\beta_k^n(\Delta_n - d_k^i)} \right), \quad (3-30)$$

其中 $d_j^i = \|\mathbf{f}_{\mathbf{p}_i} - \mathbf{f}_{\mathbf{q}_j}\|_2$ 表示特征空间中的距离， Δ_n 和 Δ_p 分别为负面和正面边距。权重 $\beta_j^p = \gamma(d_j^i - \Delta_p)$ 和 $\beta_k^n = \gamma(\Delta_n - d_k^i)$ 分别针对每个正面和负面例子确定，使用经验边距 $\Delta_p := 0.1$ 和 $\Delta_n := 1.4$ 以及超参数 γ 。以相同的方式计算反向损失 L_Q^c ，总的环形损失为 $L_c = \frac{1}{2}(L_P^c + L_Q^c)$ 。

重叠损失：重叠概率的估计被视为二元分类并使用重叠损失 $L_o = \frac{1}{2}(o_Q + o_P)$ 进

行监督，其中

$$L_P^o = \frac{1}{|P|} \sum_{i=1}^{|P|} o_{p_i} \log(o_{p_i}) + (1 - o_{p_i}) \log(1 - o_{p_i}). \quad (3-31)$$

点 p_i 的真实标签 o_{p_i} 定义为

$$o_{p_i} = \begin{cases} 1, & \text{if } \|\mathbf{T}_Q^P(p_i) - \text{NN}(\mathbf{T}_Q^P(p_i), Q)\|_2 < r_o \\ 0, & \text{otherwise,} \end{cases} \quad (3-32)$$

其中 r_o 为重叠阈值。反向损失 L_Q^o 以相同方式计算。正面和负面例子的贡献通过与其相对频率成反比的权重进行平衡。

可匹配性损失：监督可匹配性得分更为困难，因为预先并不清楚在对应关系搜索中应该考虑哪些正确的点。我们遵循一个简单的直觉：好的关键点是那些在训练期间在给定点可以成功匹配的点，使用当前的特征描述符。因此，我们将预测视为二元分类并即时生成真实标签。再次，我们将两个对称损失求和， $L_m = \frac{1}{2}(L_P^m + L_Q^m)$ ，其中

$$L_P^m = \frac{1}{|P|} \sum_{i=1}^{|P|} m_{p_i} \log(m_{p_i}) + (1 - m_{p_i}) \log(1 - m_{p_i}), \quad (3-33)$$

真实标签 m_{p_i} 通过在特征空间中进行最近邻搜索 $\text{NNF}(\cdot, \cdot)$ 即时计算：

$$m_{p_i} = \begin{cases} 1, & \text{if } \|\mathbf{T}_Q^P(p_i) - \text{NNF}(p_i, Q)\|_2 < r_m \\ 0, & \text{otherwise.} \end{cases} \quad (3-34)$$

3.5 小结

本章我们简要介绍了本文所使用的数据集，和点云配准评价指标。然后围绕基于深度学习的点云配准层层递进，由深度学习的基础知识介绍到了目前最常用的深度学习点云处理模型 PointNet, PointNet++ 和 PREDATOR。后面，我们会使用预训练的 PointNet 和 PREDATOR 作为点云特征提取器，对于描述子进行后处理。

第 4 章 基于深度学习的多实例点云配准

现有的工作主要集中在单对单的物体或场景同级别的点云配准，该方向上的工作已经十分成熟，传统的 ICP 配准方法^[17,83-84]和深度学习方法^[11,17,30]都能达到很好的效果。但是，目前的工作对于物体对场景中多个物体的不同级别的多个配准表现仍然有待提升。本文主要构建了一种基于对应聚类方法的多实例点云配准模型和一种基于对比学习的多实例点云配准模型，将在本章展开讨论。

4.1 问题陈述

在多实例点云配准问题中，源点云 \mathbf{X} 提供了一个 3D 模型的实例，目标点云 \mathbf{Y} 包含了这个模型的 K 个实例，其中这些实例是一组点的集合，这些点可能只采样了 3D 模型的一部分。如果我们将第 k^{th} 个实例写为 \mathbf{Y}_k ，那么目标点云 \mathbf{Y} 可以分解为 $\mathbf{Y} = \mathbf{Y}_0 \cup \mathbf{Y}_1 \cup \dots \cup \mathbf{Y}_k \dots \cup \mathbf{Y}_K$ 。这里我们使用 \mathbf{Y}_0 表示点云中不属于任何实例的部分。多实例 3D 配准的目标是找到刚性变换 $(\mathbf{R}_k, \mathbf{t}_k)$ ，将源实例 \mathbf{X} 对准到每个目标实例 \mathbf{Y}_k 。如果我们设法获得源实例与每个目标实例 $\mathbf{X} \leftrightarrow \mathbf{Y}_k$ 之间的点对关系，那么通过最小化对齐误差之和 (4-1)^[53]，可以从点对关系集合 $\mathbf{X} \leftrightarrow \mathbf{Y}_k$ 中求解目标点云中第 k^{th} 个实例的位姿 $(\mathbf{R}_k, \mathbf{t}_k)$ ：

$$\min_{\mathbf{R}_k, \mathbf{t}_k} \sum_i \|\mathbf{y}_{ki} - (\mathbf{R}_k \mathbf{x}_i + \mathbf{t}_k)\|^2. \quad (4-1)$$

考虑到我们已经获得了源点云和目标点云之间的一组点对关系 \mathcal{C} 。多实例配准任务的关键是将这些点对关系分类为与不同实例相关的独立集合，即：

$$\mathcal{C} = \mathcal{C}_0 \cup \mathcal{C}_1 \dots \cup \mathcal{C}_K. \quad (4-2)$$

这里， \mathcal{C}_0 用来表示异常值集合。如我们所见，多实例配准不仅需要剔除异常值点对关系，还需要解决来自不同实例的点对关系的歧义。这个任务并不容易，因为所有实例看起来都一样，而且通常存在大量的异常值点对关系。

4.2 高效对应聚类的多实例点云配准 ECC

4.2.1 结构模型

我们提出的方法的概述如图 4-1 所示。配准我们的方法以点对关系作为输入。接着通过检查点对关系之间的距离一致性来构建一个不变性一致矩阵。然后，通过将列或行向量视为这些点对关系的“特征”，将这些点对关系快速聚类成不同的组。通过凝聚聚类方法高效地进行聚类，然后通过交替合并相似的变换和重新分配簇标签进行多次迭代来进一步优化。在点对关系数量较大的情况下，我们可以选择性地应用降采样和上采样过程。详细内容将在接下来的章节中介绍。

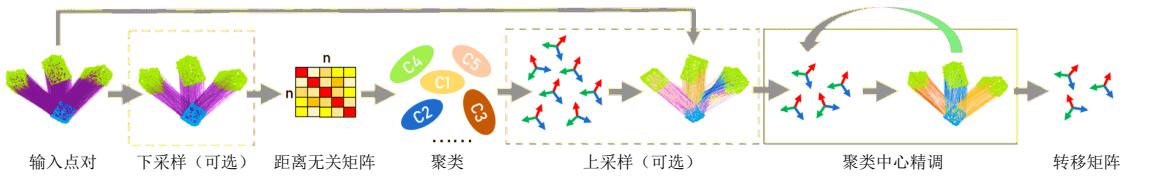


图 4-1 我们提出的多实例点云配准方法的流程。从输入点对关系构建距离不变性矩阵，用于将点对关系聚类为不同的簇（聚类），并进行优化（簇优化）。最后，从每个点对关系簇中估计与每个实例相关的刚体变换（变换）。为了处理大量的点对关系，采用两个附加过程（下采样和上采样）。

4.2.2 基于不变型矩阵的聚类

距离不变性特性在 3D 配准领域已经被研究多年^[21,36,85]，该特性描述了在刚性变换之后两点之间的距离保持不变。具体来说，如果 $c_i : \mathbf{x}_i \leftrightarrow \mathbf{y}_i$ 和 $c_j : \mathbf{x}_j \leftrightarrow \mathbf{y}_j$ 是两个真实的点对关系，那么它们应该满足

$$G_{ij} = |d_{ij} - d'_{ij}| < \delta \quad (4-3)$$

其中 $d_{ij} = |\mathbf{x}_i - \mathbf{x}_j|$, $d'_{ij} = |\mathbf{y}_i - \mathbf{y}_j|$, δ 是一个用于考虑噪声的阈值。因此， d_{ij} 和 d'_{ij} 之间的差异可以用作度量是否存在异常值，或者两个点对关系是否来自不同的刚性变换的指标。我们参考^[86]，使用相对差异作为度量，而不是在 (4-3) 中定义的绝对差异，

$$G_{ij} = s_{ij}^2, s_{ij} = \min\left(\frac{d_{ij}}{d'_{ij}}, \frac{d'_{ij}}{d_{ij}}\right) \in (0, 1). \quad (4-4)$$

通过计算所有点对关系对之间的分数，可以获得一个距离不变性矩阵 G （我们令 $G_{ii} = 1$ ）。距离不变性矩阵是对称的，其中每一列或行是一个向量，描述了给定点对

关系与其他点对关系之间的兼容性^[87]。

我们将列向量 $G_i = (G_{i1}, \dots, G_{ij}, \dots)^T$ 称为点对关系 c_i 的兼容性向量。我们观察到，如果两个点对关系属于同一个实例，它们的兼容性向量具有相似的模式。考虑两个点对关系 $c_i, c_j \in \mathcal{Cs}$ 。对于任何点对关系 $c_k \in \mathcal{Cs}$ ，由于距离不变性，我们有 $G_{ik} \rightarrow 1, G_{jk} \rightarrow 1$ 。对于其他点对关系 $c_k \in \mathcal{C}/\mathcal{Cs}$ ，我们可能有 $G_{ik} \rightarrow 0, G_{jk} \rightarrow 0$ 。换句话说， G_i, G_j 具有相似的 0 – 1 模式。相比之下，如果两个点对关系属于不同的实例，它们的兼容性向量则非常不同。

点对关系的兼容性向量可以被视为该点对关系的特征表示或“特征”。属于同一刚性变换的点对关系具有相似的特征。因此，基于这些兼容性向量，我们可以将点对关系聚类为与来自不同实例的内点相关的不同组。

4.2.3 快速点对关系聚类

我们以自底向上的方式聚类点对关系，这比现有方法采用的谱聚类^{[88][85]}要快得多。一开始，每个点对关系被视为一个独立的组。然后，我们反复合并距离最小的两个组，直到两个组之间的最小距离大于给定值 (min_dist_thresh)。定义组之间距离的方式产生了不同风格的算法。我们遵循^[34]来定义距离。设 $\mathbf{p}_i, \mathbf{p}_j$ 为两个组 i 和 j 的表示向量，组距离定义为

$$d(\mathbf{p}_i, \mathbf{p}_j) = 1 - \frac{\langle \mathbf{p}_i, \mathbf{p}_j \rangle}{\| \mathbf{p}_i \|^2 + \| \mathbf{p}_j \|^2 - \langle \mathbf{p}_i, \mathbf{p}_j \rangle}. \quad (4-5)$$

如果两个组合并，新组的表示向量更新为 $\mathbf{p}_i \leftarrow \min(\mathbf{p}_i, \mathbf{p}_j)$ ，其中 $\min(\cdot)$ 表示取两个向量每个维度的最小值。在聚类开始时，一个组（只包含一个点对关系）的表示向量设置为该点对关系的兼容性向量。

4.2.4 递归簇细化

在凝聚聚类之后，我们通过重复以下步骤来进一步优化结果，直到没有变化发生。

步骤 1. 从点对关系数量大于阈值 α 的簇中估计刚性变换。

步骤 2. 合并相似的变换。这一步将在下一节中解释。

步骤 3. 为每个点对关系重新分配簇标签。将每个点对关系分配给对齐误差最小的变换。如果在所有变换中最小的对齐误差大于 inlier_thresh ，则将点对关系标记

为异常值。

在迭代过程中，点对关系变得越来越集中，因此我们可以在步骤 1 中调整 α 以增加异常值拒绝的强度。我们在每次迭代中更新 α 的策略如下：

$$\alpha \leftarrow \min(\alpha_0 \times \theta^{n-1}, [N/100]), \quad (4-6)$$

其中 n 表示第 n^{th} 次迭代， N 是点对关系的数量， $[.]$ 是四舍五入运算。在我们的实验中，我们设置 $\alpha_0 = 3$ 和 $\theta = 3$ 。细化过程通常在我们的实验中在三次迭代内收敛，因此效率也非常高。

4.2.5 合并重复变换

有时来自不同簇的相似变换会生成，这意味着它们可能属于同一个实例。在这种情况下，我们需要合并它们。给定两个估计的变换 $(\mathbf{R}_1, \mathbf{t}_1)$ 和 $(\mathbf{R}_2, \mathbf{t}_2)$ ，我们计算每个点对关系的对齐误差，即 $e_{ki} = |\mathbf{y}_i - (\mathbf{R}_k \mathbf{x}_i + \mathbf{t}_k)|^2, (k = 1, 2)$ 。接下来，我们设置 $p_{ki} = 1$ 如果 $e_{ki} < \text{inlier_thresh}$ ，否则 $p_{ki} = 0$ 。因此，我们为两个变换获得两个二进制集 P_1, P_2 。合并两个变换的条件是

$$IOU = |P_1 \cap P_2| / |P_1 \cup P_2| \geq 80 \quad (4-7)$$

如果满足这个条件，我们将放弃一个异常值较多的变换 ($p_{ki} = 0$)。然后我们根据在所有变换中对齐误差最小的一个重新为每个点对关系分配簇标签。

4.2.6 从簇中提取变换

在聚类之后，我们需要从这些点对关系簇中提取刚性变换。由于我们不知道目标点云中真实实例的数量，因此我们需要自动选择那些内点簇。我们首先选择内点簇的元素数量大于阈值（在我们的实验中为 10）并从这些簇中估计变换。接下来，我们根据其内点数量按降序对变换进行排序。一个变换拥有的内点越多，它与真实实例相关联的可能性就越高。最后，我们检查内点数量在变换之间的降低比例，以及第一个变换（具有最多内点）之间的比例，通过

$$\gamma_k = \#I_k / \#I_0, \quad k = 1, 2, \dots \quad (4-8)$$

其中 $\#I_k$ 表示第 k^{th} 变换的内点数量。如果 $\gamma_k \leq \gamma_{thresh}$, 我们忽略所有在 k 之后的变换。 γ_{thresh} 可以更改以在召回和精确度之间进行权衡。

4.2.7 处理大量点对关系

当输入点对关系的数量很大时, 计算距离不变矩阵和聚类点对关系可能会变得昂贵。我们通过添加下采样和上采样过程来解决这个问题。在构建距离不变矩阵之前进行下采样过程, 通过随机抽样固定数量的点对关系(在我们的实现中为 1024) 进行进一步处理。在选定点对关系聚类之后进行上采样过程, 将所有点对关系分配给现有簇。分配是通过选择对齐误差最小的变换来完成的, 如第 4.2.4 节中的步骤 3 所述。

4.2.8 训练细节

我们使用 Pytorch^[89] 实现我们的算法。T-linkage 和 Progressive-X 是纯 CPU 算法, 而 CONSAC 是基于 GPU 的学习方法。我们在与 T-linkage 和 Progressive-X 相同的 CPU (Apple M2 Max 32GB) 上运行我们的算法, 并在与 CONSAC 相同的 GPU (RTX 3090 Ti) 上运行。我们的方法有三个参数, 其中在我们的实验中设置为 $min_dist_thresh = 0.2$, $inlier_thresh = 0.3$ and $\gamma_thresh = 0.5$ 。所有点云都在 $0.05m$ 体素大小中进行下采样。我们使用 Predator^[14] 来对点云进行预处理, 提取点云匹配关系。

由于一对配准中使用的指标不能用于多实例设置, 我们从检索任务中采用三个评估指标: MHR (Mean Hit Recall), MHP (Mean Hit Precision), MHF1 (Mean Hit F1)。详情请见第 3.2.3 节。

4.3 基于深度学习的多实例点云配准 DMR

4.3.1 对比学习介绍

对比学习是一种自监督学习方法, 用于为机器学习模型, 特别是深度学习模型, 学习表征。这种学习策略的基本思想是学习嵌入, 使得相似的数据点在嵌入空间中相似, 而不相似的数据点在嵌入空间中不相似。

对比学习的关键思想是在特征空间中把不相似的实例(负例)推开, 把相似的实例(正例)聚合在一起。这是通过创建一种对比损失函数来实现的, 该函数奖励模型正确识别相似和不相似的对。

对比学习的典型设置包括以下步骤：

1. 采样：对数据集进行采样，以创建相似和不相似的实例对。例如，可以通过对同一实例应用不同的数据增强来创建相似对，而可以通过配对来自不同类别的实例来创建不相似对。

2. 编码：对对中的每个实例进行编码（通常通过神经网络），以生成嵌入。

3. 对比损失：定义一种对比损失函数，该函数试图最小化嵌入空间中相似对的距离，同时最大化嵌入空间中不相似对的距离。

例如，损失可能被定义为相似实例的嵌入距离的函数，以及不相似实例的嵌入距离的函数。模型被训练以最小化这种损失。

对比学习已广泛用于机器学习的多个领域，如计算机视觉和自然语言处理，用于在大型无标签数据集上预训练模型。已经证明，对于涉及从数据中学到有意义的表征的任务，对比学习特别有效^[90]。

本文，我们对输入点云的嵌入空间进行学习，通过对对比学习的方式，使得相似的点云在嵌入空间中相似，而不相似的点云在嵌入空间中不相似，得到鲁棒的点云点对关系。

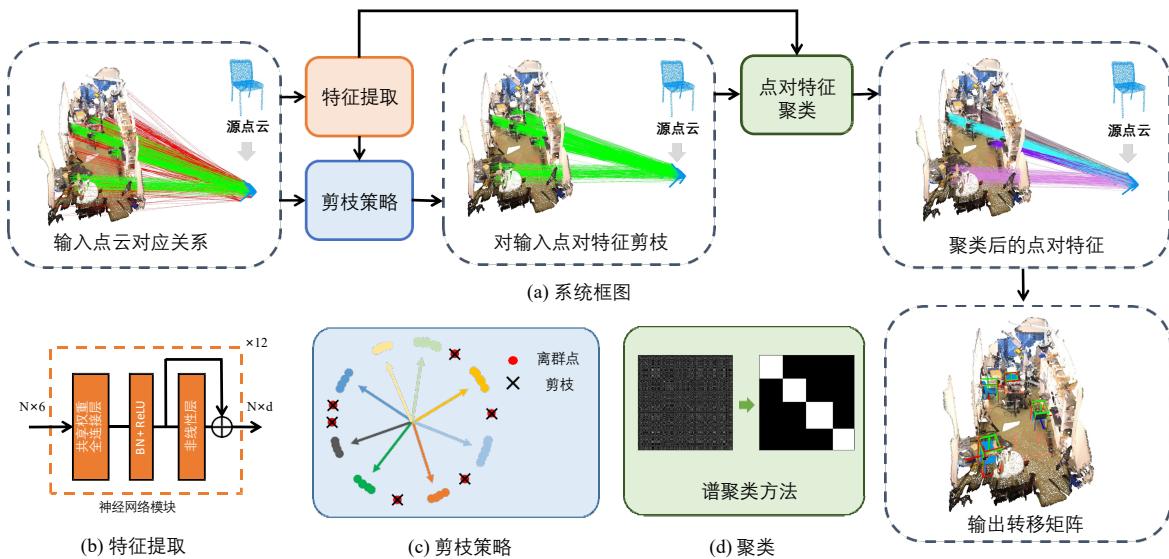


图 4-2 这是我们提出的多实例点云配准框架的流程。它接受假定的点对关系作为输入，并输出 M 个刚性变换。绿线和红线分别代表内点和离群点。在聚类之后，不同簇中的点对关系以不同颜色进行可视化。在输出变换中，绿色的边界框代表目标点云中实例的真实姿态，红色的边界框代表我们的预测。目标点云中的变换点云以蓝色进行可视化。

4.3.2 系统模型

在这一部分中，我们将展示我们的多实例点云配准框架，如图 4-2 所示。我们的框架接受特征匹配生成的假定点对关系作为输入，并首先利用通过对比学习训练的特征提取器来为输入的点对关系提取深度表征（节 4.3.3）。然后，我们根据空间一致性和它们的深度表征的相似性来剪枝点对关系（节 4.3.4）。最后，我们聚类剩余的点对关系，并使用聚类结果估计多个实例的变换（节 4.3.5）。

4.3.3 特征提取器

我们框架的第一阶段将 N 个输入的假设点对关系 $C = \{c_i\}_{i=1}^N$ 嵌入到特征空间中，以获取均匀分布的 d 维表征 $F = \{f_i \in \mathbb{R}^d\}_{i=1}^N$ ，用于后续的剪枝和聚类。这里，我们采用了^[15] 中的 SCNonlocal 模块作为我们的特征提取器，它由 12 个重复块组成。如图 4-2(b) 所示，每个块由一个共享的感知器层、一个带有 ReLU 的 BatchNorm 层和一个 nonlocal 层组成。nonlocal 层整合了刚性变换的空间一致性^[36]。在计算每层的特征之前，首先计算一个空间一致性矩阵 β :

$$\beta_{ij} = \left[1 - \frac{d_{ij}^2}{\sigma_d^2} \right]_+, \quad d_{ij} = \|\|x_i - x_j\| - \|y_i - y_j\|\| \quad (4-9)$$

其中， σ_d 是一个距离参数，用于控制对长度差异的敏感度， $[.]_+$ 表示一个夹紧函数 $\max(x, 0)$ 以使得 $\beta_{ij} \geq 0$ 。nonlocal 层使用 β 聚合中间特征:

$$f_i^{k+1} = f_i^k + MLP \left(\sum_{j=1}^{|C|} \text{soft max}_j(\alpha \beta) g(f_j^k) \right) \quad (4-10)$$

其中， α 表示第 k 块中的中间特征表征 f_i^k 和 f_j^k 之间的嵌入点积相似度， $g(\cdot)$ 表示一个线性投影函数。关于网络的更多细节可以在^[15] 中找到。

我们利用对比学习来训练我们的特征提取器，以获取均匀分布的深度表征 $F = \{f_i \in \mathbb{R}^d\}_{i=1}^N$ 。具体来说，对于一个锚定点对关系 $c_{mi} \in C_m^{gt}$ ，我们将 C_m^{gt} 中的其他点对关系定义为其正样本，并将 $C \setminus C_m^{gt}$ 中的点对关系定义为其负样本。我们将特征空间中到锚定点对关系欧几里得距离最小的负样本定义为最难的负样本。在每次迭代中，我们穷尽所有内点作为锚点，并选择他们最难的负样本来构建最难负样本集合

\mathcal{N} , 并随机选择正样本来构建正样本集合 \mathcal{P} 。我们的对比损失被定义为:

$$\mathcal{L} = \sum_{(i,j) \in \mathcal{P}} [D(f_i, f_j) - m_p]_+^2 / |\mathcal{P}| + \sum_{(i,j) \in \mathcal{N}} [m_n - D(f_i, f_j)]_+^2 / |\mathcal{N}| \quad (4-11)$$

其中 m_p 和 m_n 是正样本和负样本的距离阈值, 它们防止网络过拟合^[91]。由于输入点对关系的内点率通常不高, 我们可以将输入点对关系下采样到合适的规模, 我们对所有负样本的穷举, 以找到每个小批量的最难样本是可行的。我们的消融实验显示, 利用最难的负样本在提出的方法的高性能中起着重要的作用。

通过优化等式 4-11 中的损失, 属于不同实例的点对关系的表征在特征空间中易于分离, 而属于同一实例的点对关系的表征聚集在一起。然后, 我们使用这些均匀分布的表征进行剪枝和聚类。我们的实验表明, 深度表征提供了空间一致性没有的信息, 这使得剪枝和聚类更加有效。

4.3.4 剪枝策略

输入的点对关系往往包含大量的异常值, 这严重破坏了后续的内点点对关系聚类和变换估计。因此, 一种直观的想法是首先剪枝输入的点对关系以去除异常值。过去已经提出了一系列的剪枝策略。例如, 可以基于相似性阈值^[92] 或置信度^[18] 对输入进行剪枝。在本文中, 我们提出了一种基于密度的剪枝策略, 它根据在第 4.3.3 节中学习到的深度表征以及空间一致性约束对输入点对关系进行剪枝。

如第 4.3.3 节所述, 通过对比学习, 满足相同刚性变换的点对关系在特征空间中有相似的表征, 且它们远离其他点对关系。这意味着在内点周围的特征密度高于在异常值周围的特征密度^[93]。基于这个想法, 我们设计了一个基于密度的剪枝策略, 如图 4-2(c) 所示。我们将红色的孤立点视为异常值并移除, 而保留其他颜色的聚类点。我们的剪枝方法的详细步骤如下。

我们首先使用提取的深度表征 $F = \{f_i \in \mathbb{R}^d\}_{i=1}^N$ 来计算点对关系之间的特征相似性矩阵 S^F :

$$S_{ij}^F = \langle f_i, f_j \rangle \quad (4-12)$$

其中 $\langle \cdot, \cdot \rangle$ 表示点积。然后, 我们使用特征相似性矩阵 S^F 和空间一致性矩阵 β 来计算相似性矩阵 S :

$$S = S^F \otimes \beta \quad (4-13)$$

其中 \otimes 表示元素乘积，空间一致性矩阵 β 在之前已经计算过了。

其次，我们设置一个阈值 τ_s 并用它来将相似性矩阵 S 二值化为二进制相似性矩阵 \hat{S} 。当第 i 个点对关系和第 j 个点对关系同时在特征空间中满足足够的空间一致性和相似性时， $\hat{S}_{ij} = 1$ ，否则 $\hat{S}_{ij} = 0$ 。现在，我们可以将输入点对关系视为图的节点， \hat{S} 视为图的邻接矩阵。内点集可以被视为具有一定规模的子图，而异常值则出现为孤立点或小规模子图。

最后，我们对二进制相似性矩阵 \hat{S} 的行进行求和，并选择行和值大于阈值 τ_N 的点对关系。如图 4-2 (a) 所示，我们的剪枝方法可以有效地去除大部分甚至所有的异常值。我们的实验也证明了剪枝步骤对我们方法的重要性。我们还在补充材料中提供了如何选择阈值 τ_s 和 τ_N 以及消融实验的详细信息。

4.3.5 聚类和变换估计

经过剪枝后，我们得到了一个干净的点对关系集合，这个集合几乎没有异常值。下一步是将这些点对关系划分为属于不同实例的多个子集，并估计所有实例的最终刚性变换。点对关系的划分可以被视为一个聚类问题，实例的数量应该等于聚类的数量。在这里，我们使用了一种能够自动确定聚类数量的谱聚类算法^[94-95]。该算法包括以下步骤：

步骤 1：使用剩余点对关系的特征相似性和空间一致性重新计算二进制相似性矩阵 \hat{S}_n ，其中 n 表示剪枝后的点对关系数量；

步骤 2：计算矩阵 \hat{S}_n 的标准化拉普拉斯矩阵 \hat{L}_n ，并计算标准化拉普拉斯矩阵的特征值 $\lambda_1(\hat{L}_n) \leq \lambda_2(\hat{L}_n) \leq \dots \leq \lambda_n(\hat{L}_n)$ ；

步骤 3：通过以下公式确定聚类数量 M ：

$$M = \arg \max_k \left\{ \lambda_{k+1}(\hat{L}_n) - \lambda_k(\hat{L}_n) \right\} \quad (4-14)$$

步骤 4：应用具有 M 个聚类的谱聚类^[94]。

由于深度表征在特征空间中有良好的分布，二进制相似性矩阵 \hat{S}_n 符合^[95] 中定义的理想矩阵。因此，由公式 4-14 确定的聚类数量是可靠的，这在^[95] 中得到了证明。

在以上步骤之后，剩余的点对关系被分配到不同的聚类中，如图 4-2(a) 所示，然后我们可以使用如 RANSAC^[96] 等求解器来估计每个实例的刚性变换。因为我们的剪枝和聚类策略非常有效，每个实例的内点率很高，只需要几十次 RANSAC 迭代就能

达到出色的性能。

4.3.6 训练细节

我们在 Pytorch^[89] 中实现了我们的网络，并使用 scikit-learn^[97] 实现谱聚类。我们随机地将输入点对关系下采样到 1000。对于合成数据集，距离参数 σ_d 被设置为 0.05，对于真实数据集，被设置为 0.1。我们将深度表征的维数 d 设置为 128。阈值 τ_S 被设置为 0.85。对于合成数据集，阈值 τ_N 设置为 10，对于真实数据集，设置为 20。边缘 m_p 和 m_n 分别设置为 0.1 和 1.4。我们的批量大小设置为 16。我们使用 ADAM 优化器优化网络，初始学习率为 0.01，并训练网络 15K 次迭代。所有实验都在 NVIDIA GPU 3090Ti 图形卡上进行。

4.4 小结

本章，我们介绍了提出的两种多实例点云配准流程，高效对应聚类的多实例点云配准和基于深度学习的多实例点云配准，这两种方法均需要通过好的方法来得到好的点云点对关系。下面我们将对这两种方法进行实验验证。

第 5 章 实验结果与分析

本章，我们对使用点云对应聚类的方式和基于深度学习的点云配准方法来进行实验。实验的数据集分为仿真数据集和真实数据集。

5.1 数据集

5.1.1 仿真数据集

如第3.1节所述，我们从 PointNet++^[38] 的预采样 Modelnet40 数据集^[56] 生成一个仿真数据集。我们将每个点云下采样到 256 个点，并随机生成 K 个变换（在我们的测试中最多为 20）来形成一个目标点云。目标点云也与其他物体和随机点混合，以更好地模拟真实世界的情况。

5.1.2 真实数据集

Scan2CAD^[57] 是一个数据集，它将 ShapeNet^[58] CAD 模型与 ScanNet^[62] 点云中的对象实例对齐。部分扫描有多个已对齐的 CAD 模型和标注的姿态。我们选择那些包含多个 CAD 模型的扫描作为目标点云，并从 CAD 模型中采样源点云进行测试。我们为配准测试生成了 173 个样本，其中大多数样本包含 2 ~ 5 个实例。请注意，在每个点云中，Scan2CAD 中仅标注了部分实例。这意味着我们无法使用部分标注的姿态正确评估性能，如准确率和召回率。为解决这个问题，我们仅在目标点云中标注的物体的真实边界框内匹配点以生成对应关系。同样，我们使用 PREDATOR^[14] 和 D3Feat^[22] 进行点匹配，两者都是使用 Scan2CAD 数据集中的 1028 个训练样本和 187 个验证样本进行微调的。

5.2 高效对应聚类的多实例点云配准 ECC

5.2.1 仿真数据集

5.2.1.1 仿真点对关系

在这个测试中，我们直接通过混合真实值和离群值来生成输入对应关系。我们测试了不同的离群比例，10% ~ 50%，50% ~ 70% 和 70% ~ 90%。请注意，对于每个测试样本，离群点是在给定范围内随机抽样的。结果显示在表 5-2 中。随着离群比

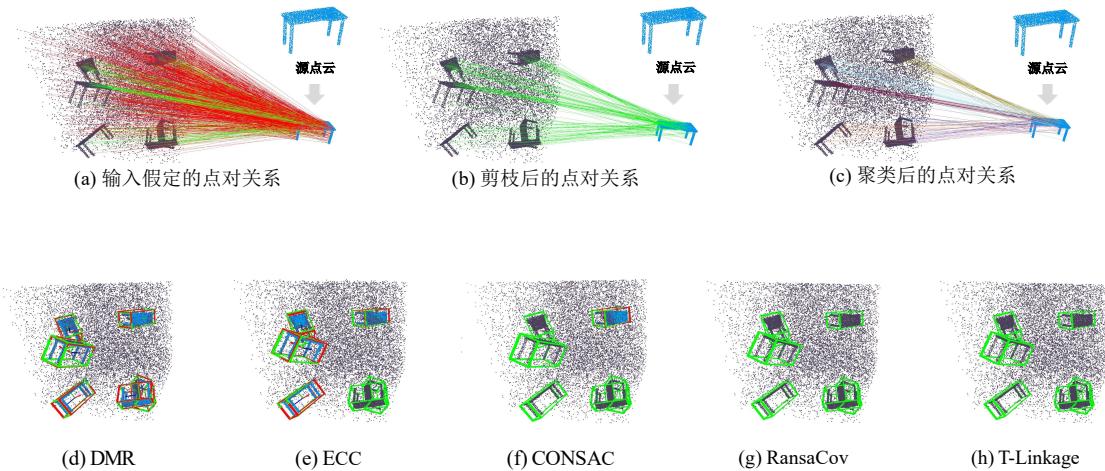


图 5-1 在仿真数据集上的结果。在 (a) 和 (b) 中，绿线和红线分别代表内点对应关系和异常值对应关系。在 (c) 中，聚类后每个簇中的对应关系以不同的颜色可视化。在 (d-f) 中，绿色边界框表示目标点云中实例的真实姿态，红色边界框表示预测。目标点云中的变换点云以蓝色可视化。

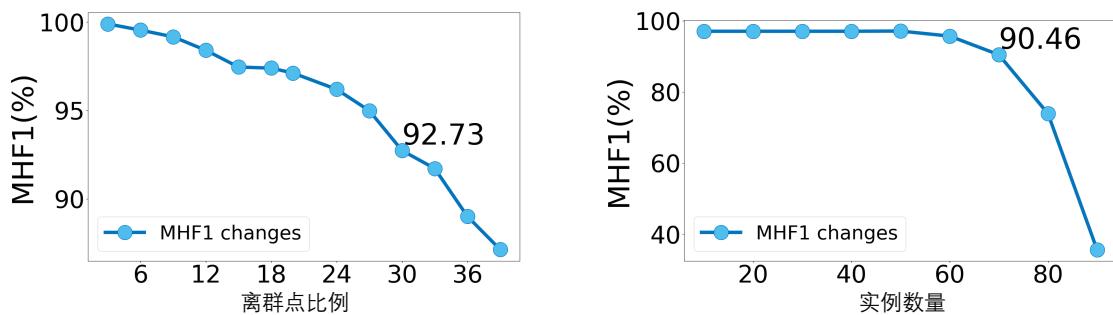


图 5-2 (a) 平均 Hit F1 与离群比例关系。(b) 平均 Hit F1 与实例数量关系 (固定离群比例为 50%)。

例的增加，几乎所有方法的性能都有所下降，但我们的方法下降缓慢，且仍然明显优于其他方法。我们的算法在 CPU 或 GPU 上的速度比现有方法快 10 倍。我们还在图 5-2(a) 中绘制了我们的方法在不同离群比例下的 MHF1(Mean Hit F1) 曲线，其中包括 20 个实例。尽管当离群比例非常大时性能迅速下降，但我们的方法在 70% 的离群比例下仍然能达到 90.46% 的 MHF1。图 5-2(b) 显示了在固定离群比例 50% 的情况下，不同实例数量的 MHF1 曲线。即使存在 30 个实例，我们的方法的 MHF1 也约为 92.73%。

Metric	MHR(%) \uparrow	MHP(%) \uparrow	MHF1(%) \uparrow	Time(s) \downarrow
PREDATOR(estimated outlier ratio : 76.44%)				
T-Linkage	2.46	3.79	2.71	1655.0
Progressive-X	11.58	6.86	7.87	26.32
CONSAC	2.66	0.35	0.62	21.35
ECC	31.63	29.23	27.04	1.46/0.51
D3Feat (estimated outlier ratio : 97.25%)				
T-Linkage	0.04	0.22	0.06	2178.43
Progressive-X	0.67	0.30	0.4	28.48
CONSAC	0	0	0	21.88
ECC	0.29	0.04	0.07	2.13/0.89

表 5-1 在 Scan2CAD 数据集上配准的结果。

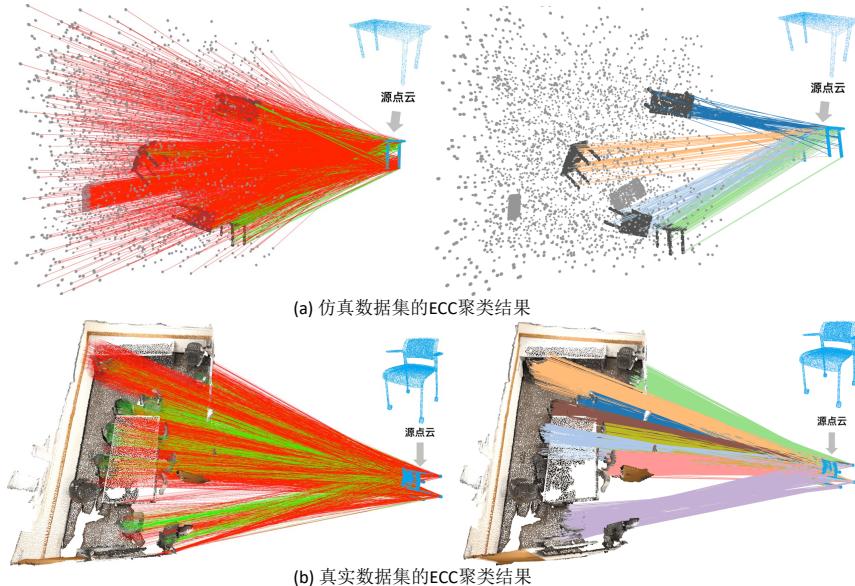


图 5-3 ECC 方法在两个数据集的聚类可视化结果

5.2.1.2 基于特征的点对关系生成方式

在此测试中，我们使用 PREDATOR^[14] 和 D3Feat^[22] 特征匹配来获得点对应关系。这两个特征模型都是在仿真数据上训练的。结果显示在表 5-3 中。值得注意的是，这两个特征都产生了超过 90% 的高离群点比例。在这样一个具有挑战性的情况下，我们的方法在鲁棒性和效率方面仍然表现良好，且明显优于现有方法。使用 D3Feat 的结果要比使用 PREDATOR 的结果差得多。原因不仅是因为更多的离群点，而且我们在检查结果时发现缺少内点。在后面的实验中我们在图 5-3 中展示了一些结果。

方法	MHR(%) ↑	MHP(%) ↑	MHF1(%) ↑	时间 (s) ↓
离群点比重: 10% ~ 50%				
T-Linkage	3.05	14.80	4.65	57.27
Progressive-X	27.91	80.28	41.04	87.25
CONSAC	0.47	0.47	0.47	9.23
ECC	96.08	99.73	97.03	0.62/0.30
离群点比重: 50% ~ 70%				
T-Linkage	1.33	7.00	2.05	56.90
Progressive-X	20.60	75.10	31.70	85.54
CONSAC	0.49	0.49	0.49	9.55
ECC	93.99	99.49	95.51	0.55/0.28
Outlier ratio : 70% ~ 90%				
T-Linkage	0.81	4.42	1.25	56.89
Progressive-X	12.88	62.60	20.73	84.5
CONSAC	0.51	0.51	0.51	7.70
ECC	60.39	94.42	69.36	0.50/0.24
离群点比重: 90% ~ 99%				
T-Linkage	0.28	1.30	0.42	56.69
Progressive-X	7.13	39.19	11.67	84.43
CONSAC	0.51	0.51	0.51	9.57
ECC	14.70	65.20	22.75	0.47/0.21

表 5-2 在不同离群比例的仿真对应关系上的结果。↑表示越大越好，而↓表示相反。我们方法在 CPU/GPU 上的运行时间也展示出来。

5.2.2 真实数据集

在真实数据集中的测试结果如表 5-1 所示。我们的方法在使用 PREDATOR 时明显优于现有方法。请注意，当使用 D3Feat 时，所有方法的性能都很差。在仔细检查结果后，我们发现原因不仅是高异常值比率（约 97.25%），而且在使用 D3Feat 时，即使特征匹配仅限于目标点云中的真实边界框内，内点也不足。一些结果如图 5-4(e) 所示。

5.3 基于深度学习的多实例点云配准 DMR

5.3.1 仿真数据集

我们首先将基于深度学习的方法与仿真数据集上的其他竞争者进行比较，结果如表 5-4 所示。由于内点比例极低，基于采样的方法如 T-Linkage、RansaCov 和 CONSAC 的性能不佳。由于空间一致性的优势，ECC 表现出有效性。然而，我们的 DMR 在所有评价指标上均大幅度超越了第二好的方法 ECC。

为了定性评估我们的 DMR 并将其与其他竞争者进行比较，我们在图 5-1 中提

Metric	MHR(%) ↑	MHP(%) ↑	MHF1(%) ↑	Time(s) ↓
PREDATOR (estimated outlier ratio : 94.32%)				
T-Linkage	0.19	0.54	0.27	43.46
Progressive-X	15.90	31.01	18.98	86.39
CONSAC	0.1	0.07	0.08	7.65
ECC	53.39	61.44	51.80	1.28/0.48
D3Feat (estimated outlier ratio : 99.30%)				
T-Linkage	0.07	0.29	0.1	56.37
Progressive-X	4.29	15.28	5.94	87.22
CONSAC	0.13	0.04	0.05	9.53
ECC	16.98	27.05	17.91	0.68/0.30

表 5-3 使用特征匹配在仿真数据上生成对应点的结果。部分结果可在图 5-3 中进行可视化。

表 5-4 在仿真数据集上的配准结果

	MHR(%)	MHP(%)	MHF1(%)	Time(s)
T-linkage	0.61	1.48	0.87	3.89
RansaCov	0.73	5.33	1.29	0.14
CONSAC	1.00	7.45	1.77	0.61
ECC	82.90	92.92	87.63	3.56
DMR	92.60	99.69	96.01	0.06

供了一组可视化。图 5-1 的第一行显示了输入对应关系，以及我们的剪枝和聚类结果。图 5-1(b) 显示，我们的 DMR 惊人地去除了所有的离群点，剩余的对应关系在图 5-1(c) 中被很好地聚类。

图 5-1 的第二行显示了我们提出的方法和竞争者的配准结果。可以看到，T-Linkage 和 RansaCov 都无法配准任何实例。对于目标点云中的六个实例，CONSAC 只成功配准了一个实例。值得注意的是，尽管 ECC 成功配准了四个实例，但它未能配准右下角的两张桌子。这是因为这两张桌子混在一起，空间一致性不足以区分它们。然而，我们的方法成功配准了所有实例。

5.3.2 真实数据集

如表 5-5 所示，我们的 DMR 在所有三个评估指标，MHR, MHP 和 MHF 上均优于所有竞争者，速度也很有竞争力。ECC 和 DMR 的性能低于仿真数据集，而其他方法的性能高于仿真数据集。这是由于实例内点比率分布的变化。

我们还提供了一组可视化，以定性评估我们的 DMR 并与其他竞争者进行比较。图 5-4的第一行显示了输入对应关系，以及我们的剪枝和聚类结果。图 5-4(b) 表明，

表 5-5 在真实数据集上的配准结果

	MHR(%)	MHP(%)	MHF1(%)	Time(s)
T-linkage	34.99	46.86	40.07	6.64
RansaCov	60.50	33.28	42.94	0.07
CONSAC	55.48	53.34	54.39	0.39
ECC	64.66	69.73	67.10	1.84
DMR	78.10	70.64	74.18	0.10

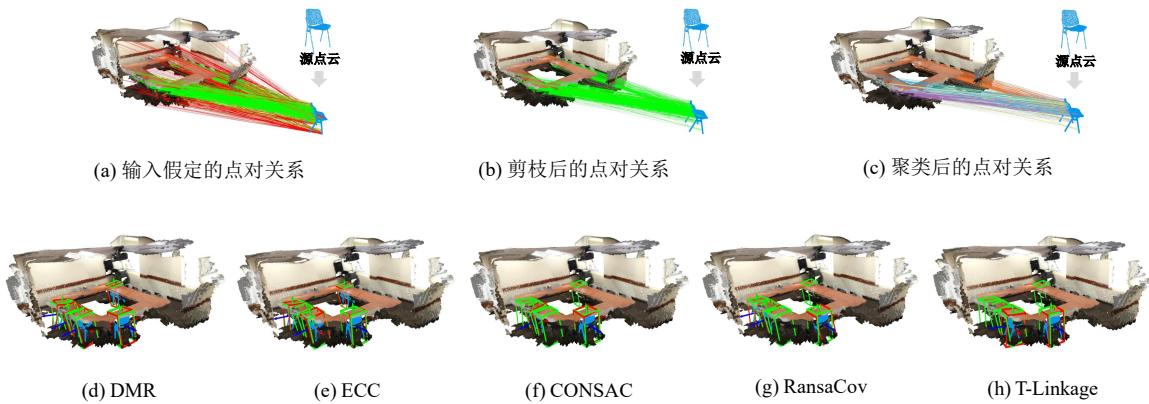


图 5-4 在真实数据集上的结果。在 (a) 和 (b) 中，绿线和红线分别代表内点对应关系和异常值对应关系。在 (c) 中，聚类后每个簇中的对应关系以不同的颜色可视化。在 (d-f) 中，绿色边界框表示目标点云中实例的真实姿态，红色边界框表示预测。目标点云中的变换点云以蓝色可视化。

我们的方法几乎去除了所有的异常值，确保了如图 5-4(c) 所示的后续聚类高效执行。对于包含在目标点云中的五个实例，T-Linkage 和 CONSAC 成功配准了两个实例，但 T-Linkage 的一个预测有大的错误。RansaCov 成功配准了三个实例。由于目标点云中的三把椅子靠得很近，ECC 并没有成功配准所有这些实例。我们的方法成功完成了所有实例的配准。

5.3.3 消融实验

5.3.3.1 深度表达的消融研究

为了研究我们采用的深度表达的有效性，我们比较了我们的框架在有和无深度表达下的性能。没有深度表达的版本删除了特征提取器并设置 $S = \beta$ ，只依赖空间一致性进行剪枝和聚类。比较结果如表 5-6 所示。可以看到，使用深度表达后，准确性和速度指标都有所提高。没有深度表达，剪枝的对应关系少，需要聚类的对应关系多，这增加了我们方法的运行时间。尽管没有深度表达的框架在 MHR、MHP 和

表 5-6 深度表达和剪枝的消融研究结果。

深度表达	剪枝	MHR(%)	MHP(%)	MHF1(%)	Time(s)
	✓	76.61	65.05	70.36	0.17
✓		62.23	32.77	42.93	1.24
✓	✓	78.10	70.64	74.18	0.10

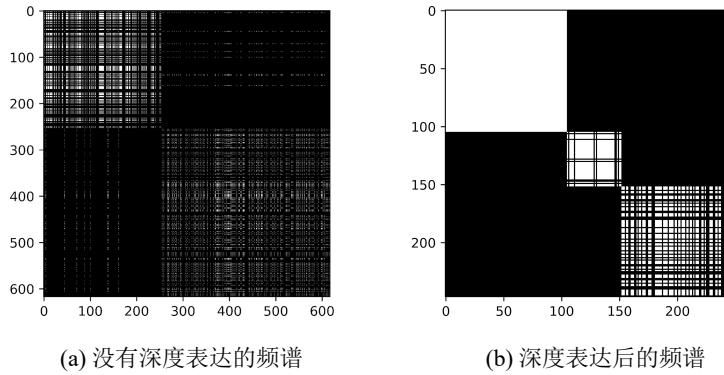


图 5-5 无深度表达和有深度表达的聚类结果可视化。

MHF1 上的性能较低，但它仍然是一个有竞争力的基线，适合没有训练数据的情况。这表明了本文提出的剪枝和聚类策略的有效性。

此外，我们在图 5-5 中可视化了有无深度表达的聚类结果。我们选择了一个例子，其目标点云包含三个实例。我们首先使用一个 3 维的 one-hot 向量来表示一个对应关系属于哪个实例。然后我们使用这些向量来计算相似性矩阵，并使用聚类的结果对这些矩阵进行排列。可以看到，通过深度表达排列的相似性矩阵比没有深度表达的框架排列的相似性矩阵小得多，因为在前者的情况下，剪枝过程中去除了更多的异常值。更重要的是，图 5-5(b) 中的矩阵显示了三个明显的簇，这些簇对应于三个实例。相反，图 5-5(a) 中的矩阵显示了两个块，其中右下角的块实际上对应于两个实例，如果不使用提出的深度表达，无法成功区分。

5.3.3.2 对剪枝的消融研究

为了定量研究我们的剪枝策略的有效性，我们简单地在我们的方法中去除剪枝策略，然后比较去除前后的性能。表 5-6 显示，如果去除剪枝步骤，我们的方法的性能会大幅下降，这是由于噪声二进制相似性矩阵不符合^[95] 定义的理想矩阵，光谱聚类无法正确地对应关系进行分组。这个结果表明，剪枝对后续的聚类至关重要。

表 5-7 RANSAC 迭代次数的影响

RANSAC 次数	MHR(%)	MHP(%)	MHF1(%)
5	77.39	67.76	72.26
50	78.10	70.64	74.18
500	79.33	70.64	74.73

表 5-8 用最难的负样本进行训练的消融实验结果。

	MHR(%)	MHP(%)	MHF1(%)	Positive(%)	Top-1(%)	Top-10(%)
随机	76.85	67.50	71.87	95.43	96.91	91.37
最难	78.10	70.64	74.18	83.96	61.32	49.65

5.3.3.3 对 RANSAC 迭代次数的消融研究

此外，我们还测试了我们的模型在不同的 RANSAC 迭代次数下的性能，结果如表 5-7 所示。可以看到，只用五次迭代就可以得到相当好的结果，而 50 次和 500 次迭代的性能非常接近。这些结果表明，剪枝和聚类步骤极大地提高了每个实例的内点比例，因此，只使用少量的迭代次数就可以估计出可靠的结果。

5.3.3.4 关于最难负样本消融研究

为每个小批量找到最困难的负对会产生更大的计算开销，但我们的消融研究显示这是值得的。我们将其与使用随机选择的负对训练的相同模型进行比较。使用最困难的负对（如本方法所述）和使用随机选择的负对的结果如表 5-8 所示。所有的准确性指标都通过在对比学习中使用最困难的负对得到了提升。通过比较表 5-8 和表 5-6，我们可以发现，在对比表示学习中使用随机负对得到的结果优于不使用深度表达，但差距很小。此外，我们在表 5-8 中收集了特征空间中正对和 top-K 最困难的负对的平均余弦相似性。结果显示，使用最困难的负对训练的表示在特征空间中更好地分离，更具区分性。

5.4 小结

本章，我们对提出的两种多实例点云配准模型进行了广泛的实验评估，包括定量的实验评估和可视化的实验评估，并且对基于深度学习的方法进行了广泛的消融实验。其中基于深度学习的方法效果超过了单纯的聚类算法，证明了深度学习方法在多实例点云配准中的有效性，学习鲁棒的点对特征能够大大增大配准的效果。

结 论

本文对三维点云配准进行了详尽的调研，从点云配准的数学方法入手，引入了基于深度学习的点云配准和经典的点云处理和配准的模型，并且介绍了多实例点云配准的新任务。

在本文研究中，我们关注于多实例点云配准的新任务。首先根据距离不变矩阵的列向量编码了大量关于对应关系相关实例的信息。然后通过聚类算法有效地将对应关系聚集到不同的组中，并通过多次迭代改进结果。无论是在仿真数据集还是真实数据集的结果都表明，这个基于聚类的方法在稳健性、准确性和效率方面都明显优于现有的方法。

但是这个解决方案在所讨论的环境下还没有达到完美，所以我们提出了一个新的基于深度学习框架来解决多实例点云配准的问题。我们利用对比学习来学习输入对应关系的分布良好的深度表示。在此基础上，我们在聚类之前使用剪枝策略，以有效地删除异常值对应并将剩余的内部对应关系分配给正确的实例。然后，可以很容易地估计从源点云到每个实例的转换。在仿真数据集和真实世界数据集的广泛实验证实了我们的框架的有效性和其优于现有解决方案的优势。所提出的表示学习和异常值修剪策略有可能应用于成对点云配准。

本文对基于深度学习的多实例点云配准算法研究取得了一定的结果，但仍有可进一步探索的空间，在以后的研究中，希望能够提出一个全真的大场景数据集，比如城市场景的车辆点云配准任务。并且在多实例点云配准任务中，我们可以尝试更多的聚类算法，比如谱聚类、层次聚类等，来提高多实例点云配准的效果。在以后的研究中，希望能够借助类似 DETR^[98] 的结构来实现完全的端对端算法，能够使用转移矩阵筛选的功能来达到多实例点云配准的效果。

参考文献

- [1] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.
- [2] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. ArXiv preprint arXiv:1409.1556, 2014.
- [3] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
- [4] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[J]. ArXiv preprint arXiv:2010.11929, 2020.
- [5] Qi C R, Su H, Mo K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 652-660.
- [6] Huang X, Mei G, Zhang J, et al. A comprehensive survey on point cloud registration[J]. ArXiv preprint arXiv:2103.02690, 2021.
- [7] Besl P J, McKay N D. Method for registration of 3-D shapes[C]//Sensor fusion IV: control paradigms and data structures: vol. 1611. 1992: 586-606.
- [8] 周慧子, 胡学敏, 陈龙, 等. 面向自动驾驶的动态路径规划避障算法[J]. 计算机应用, 2017, 37(883-888).
- [9] Deng H, Birdal T, Ilic S. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 602-618.
- [10] Deng H, Birdal T, Ilic S. Ppfnet: Global context aware local features for robust 3d point matching[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 195-205.
- [11] Qin Z, Yu H, Wang C, et al. Geometric transformer for fast and robust point cloud registration[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 11143-11152.
- [12] Stückler J, Behnke S. Robust real-time registration of RGB-D images using multi-resolution surfel representations[C]//ROBOTIK 2012; 7th German Conference on Robotics. 2012: 1-4.
- [13] Pomerleau F, Colas F, Siegwart R, et al. A review of point cloud registration algorithms for mobile robotics[J]. Foundations and Trends® in Robotics, 2015, 4(1): 1-104.
- [14] Huang S, Gojcic Z, Usvyatsov M, et al. Predator: Registration of 3d point clouds with low overlap[C]//Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. 2021: 4267-4276.
- [15] Bai X, Luo Z, Zhou L, et al. Pointdsc: Robust point cloud registration using deep spatial consistency[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 15859-15869.
- [16] Wang H, Liu Y, Dong Z, et al. You only hypothesize once: Point cloud registration with rotation-equivariant descriptors[C]//Proceedings of the 30th ACM International Conference on Multimedia. 2022: 1630-1641.
- [17] Barath D, Rozumny D, Eichhardt I, et al. Progressive-x+: Clustering in the consensus space[J]. ArXiv preprint arXiv:2103.13875, 2021.

- [18] Zhao C, Ge Y, Zhu F, et al. Progressive Correspondence Pruning by Consensus Learning[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 6464-6473.
- [19] Barath D, Matas J. Graph-cut RANSAC[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 6733-6741.
- [20] Kluger F, Brachmann E, Ackermann H, et al. Consac: Robust multi-model fitting by conditional sample consensus[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 4634-4643.
- [21] Yang H, Shi J, Carlone L. Teaser: Fast and certifiable point cloud registration[J]. IEEE Transactions on Robotics, 2020, 37(2): 314-333.
- [22] Bai X, Luo Z, Zhou L, et al. D3feat: Joint learning of dense detection and description of 3d local features[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 6359-6367.
- [23] Yi K M, Trulls E, Ono Y, et al. Learning to find good correspondences[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 2666-2674.
- [24] Qi C R, Litany O, He K, et al. Deep hough voting for 3d object detection in point clouds[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 9277-9286.
- [25] Wang W, Yu R, Huang Q, et al. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 2569-2578.
- [26] Han L, Zheng T, Xu L, et al. Occuseg: Occupancy-aware 3d instance segmentation[C]// Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 2940-2949.
- [27] Barath D, Matas J. Multi-class model fitting by energy minimization and mode-seeking[C]// Proceedings of the European Conference on Computer Vision (ECCV). 2018: 221-236.
- [28] Hartley R I. In defense of the eight-point algorithm[J]. IEEE Transactions on pattern analysis and machine intelligence, 1997, 19(6): 580-593.
- [29] Tang W, Zou D. Multi-instance point cloud registration by efficient correspondence clustering[C]// Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 6667-6676.
- [30] Barath D, Matas J. Progressive-x: Efficient, anytime, multi-model fitting algorithm[C]// Proceedings of the IEEE/CVF international conference on computer vision. 2019: 3780-3788.
- [31] Kanazawa Y, Kawakami H. Detection of planar regions with uncalibrated stereo using distributions of feature points.[C]//BMVC. 2004: 1-10.
- [32] Toldo R, Fusiello A. Robust multiple structures estimation with j-linkage[C]//Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part I 10. 2008: 537-547.
- [33] Magri L, Fusiello A. Multiple model fitting as a set coverage problem[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 3318-3326.
- [34] Magri L, Fusiello A. T-linkage: A continuous relaxation of j-linkage for multi-model fitting[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 3954-3961.
- [35] Magri L, Andrea F, et al. Robust multiple model fitting with preference analysis and low-rank approximation[C]//Procedings of the British Machine Vision Conference 2015. 2015: 20-1.
- [36] Leordeanu M, Hebert M. A spectral technique for correspondence problems using pairwise

- constraints[C]//Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1: vol. 2. 2005: 1482-1489.
- [37] Yuan M, Li Z, Jin Q, et al. PointCLM: A Contrastive Learning-based Framework for Multi-instance Point Cloud Registration[C]//Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX. 2022: 595-611.
- [38] Qi C R, Yi L, Su H, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space[J]. Advances in neural information processing systems, 2017, 30.
- [39] Leberl F, Irschka A, Pock T, et al. Point clouds[J]. Photogrammetric Engineering & Remote Sensing, 2010, 76(10): 1123-1134.
- [40] Jiang C, Sud A, Makadia A, et al. Local implicit grid representations for 3d scenes[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 6001-6010.
- [41] Guan B, Lin S, Wang R, et al. Voxel-based quadrilateral mesh generation from point cloud[J]. Multimedia Tools and Applications, 2020, 79: 20561-20578.
- [42] Cruz L, Lucio D, Velho L. Kinect and rgbd images: Challenges and applications[C]//2012 25th SIBGRAPI conference on graphics, patterns and images tutorials. 2012: 36-49.
- [43] Johnson A E. Spin-images: a representation for 3-D surface matching[J], 1997.
- [44] Steder B, Rusu R B, Konolige K, et al. NARF: 3D range image features for object recognition[C]//Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS): vol. 44. 2010: 2.
- [45] Rusu R B, Blodow N, Beetz M. Fast point feature histograms (FPFH) for 3D registration[C]//2009 IEEE international conference on robotics and automation. 2009: 3212-3217.
- [46] Salti S, Tombari F, Di Stefano L. SHOT: Unique signatures of histograms for surface and texture description[J]. Computer Vision and Image Understanding, 2014, 125: 251-264.
- [47] Pio R. Euler angle transformations[J]. IEEE Transactions on automatic control, 1966, 11(4): 707-715.
- [48] Shoemake K. Animating rotation with quaternion curves[C]//Proceedings of the 12th annual conference on Computer graphics and interactive techniques. 1985: 245-254.
- [49] Horn A. Doubly stochastic matrices and the diagonal of a rotation matrix[J]. American Journal of Mathematics, 1954, 76(3): 620-630.
- [50] Diebel J, et al. Representing attitude: Euler angles, unit quaternions, and rotation vectors[J]. Matrix, 2006, 58(15-16): 1-35.
- [51] Šenk M, Cheze L. Rotation sequence as an important factor in shoulder kinematics[J]. Clinical biomechanics, 2006, 21: S3-S8.
- [52] Stuelpnagel J. On the parametrization of the three-dimensional rotation group[J]. SIAM review, 1964, 6(4): 422-430.
- [53] Levinson J, Esteves C, Chen K, et al. An analysis of svd for deep rotation estimation[J]. Advances in Neural Information Processing Systems, 2020, 33: 22554-22565.
- [54] Levenberg K. A method for the solution of certain non-linear problems in least squares[J]. Quarterly of applied mathematics, 1944, 2(2): 164-168.
- [55] 李娇娇, 孙红岩, 董雨, 等. 基于深度学习的 3 维点云处理综述[J]. 计算机研究与发展, 2022, 59(5): 20.
- [56] Sun J, Zhang Q, Kailkhura B, et al. MODELNET40-C: A Robustness BENCHMARK FOR 3D POINT CLOUD RECOGNITION UNDER CORRUPTION[J],

- [57] Avetisyan A, Dahnert M, Dai A, et al. Scan2cad: Learning cad model alignment in rgb-d scans[C]//Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. 2019: 2614-2623.
- [58] Chang A X, Funkhouser T, Guibas L, et al. Shapenet: An information-rich 3d model repository[J]. ArXiv preprint arXiv:1512.03012, 2015.
- [59] Siddiqi K, Zhang J, Macrini D, et al. Retrieving articulated 3-D models using medial surfaces[J]. Machine vision and applications, 2008, 19: 261-275.
- [60] De Deuge M, Quadros A, Hung C, et al. Unsupervised feature learning for classification of outdoor 3d scans[C]//Australasian conference on robotics and automation: vol. 2: 1. 2013.
- [61] Wu Z, Song S, Khosla A, et al. 3d shapenets: A deep representation for volumetric shapes[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1912-1920.
- [62] Dai A, Chang A X, Savva M, et al. Scannet: Richly-annotated 3d reconstructions of indoor scenes[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 5828-5839.
- [63] Uy M A, Pham Q H, Hua B S, et al. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 1588-1597.
- [64] Armeni I, Sener O, Zamir A R, et al. 3d semantic parsing of large-scale indoor spaces[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 1534-1543.
- [65] Shotton J, Glocker B, Zach C, et al. Scene coordinate regression forests for camera relocalization in RGB-D images[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2013: 2930-2937.
- [66] Chang A, Dai A, Funkhouser T, et al. Matterport3d: Learning from rgb-d data in indoor environments[J]. ArXiv preprint arXiv:1709.06158, 2017.
- [67] Choy C, Park J, Koltun V. Fully Convolutional Geometric Features[C]//ICCV. 2019.
- [68] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.
- [69] Wang Y, Sun Y, Liu Z, et al. Dynamic graph cnn for learning on point clouds[J]. Acm Transactions On Graphics (tog), 2019, 38(5): 1-12.
- [70] Choy C, Dong W, Koltun V. Deep global registration[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 2514-2523.
- [71] Huang J, Birdal T, Gojcic Z, et al. Multiway non-rigid point cloud registration via learned functional map synchronization[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.
- [72] Lu W, Wan G, Zhou Y, et al. Deepvcp: An end-to-end deep neural network for point cloud registration[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 12-21.
- [73] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. ArXiv preprint arXiv:1412.6980, 2014.
- [74] He K, Fan H, Wu Y, et al. Momentum contrast for unsupervised visual representation learning[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 9729-9738.
- [75] Abadi M. TensorFlow: learning functions at scale[C]//Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming. 2016: 1-1.
- [76] Paszke A, Gross S, Massa F, et al. Pytorch: An imperative style, high-performance deep learning library[J]. Advances in neural information processing systems, 2019, 32.

- [77] Ma Y, Yu D, Wu T, et al. PaddlePaddle: An open-source deep learning platform from industrial practice[J]. *Frontiers of Data and Computing*, 2019, 1(1): 105-115.
- [78] Chen T, Li M, Li Y, et al. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems[J]. *ArXiv preprint arXiv:1512.01274*, 2015.
- [79] Chua L O, Roska T. The CNN paradigm[J]. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 1993, 40(3): 147-156.
- [80] Albawi S, Mohammed T A, Al-Zawi S. Understanding of a convolutional neural network[C]//2017 international conference on engineering and technology (ICET). 2017: 1-6.
- [81] Sabour S, Frosst N, Hinton G E. Dynamic routing between capsules[J]. *Advances in neural information processing systems*, 2017, 30.
- [82] Sun Y, Cheng C, Zhang Y, et al. Circle loss: A unified perspective of pair similarity optimization[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 6398-6407.
- [83] Li P, Wang R, Wang Y, et al. Evaluation of the ICP algorithm in 3D point cloud registration[J]. *IEEE Access*, 2020, 8: 68030-68048.
- [84] Shi X, Liu T, Han X. Improved Iterative Closest Point (ICP) 3D point cloud registration algorithm based on point cloud filtering and adaptive fireworks for coarse registration[J]. *International Journal of Remote Sensing*, 2020, 41(8): 3197-3220.
- [85] Shi J, Yang H, Carlone L. ROBIN: a graph-theoretic approach to reject outliers in robust estimation using invariants[C]//ICRA. 2021: 13820-13827.
- [86] Buch A G, Yang Y, Krüger N, et al. In Search of Inliers: 3D Correspondence by Local and Global Voting[C]//CVPR. 2014: 2075-2082.
- [87] Yang J, Xian K, Wang P, et al. A Performance Evaluation of Correspondence Grouping Methods for 3D Rigid Data Matching[J]. *IEEE TPAMI*, 2019, 14(8): 1-1.
- [88] Parra Á, Chin T J, Neumann F, et al. A Practical Maximum Clique Algorithm for Matching with Pairwise Constraints[J]. *ArXiv preprint arXiv:1902.01534*, 2019.
- [89] Paszke A, Gross S, Chintala S, et al. Automatic differentiation in PyTorch[J], 2017.
- [90] Tian Y, Sun C, Poole B, et al. What makes for good views for contrastive learning?[J]. *Advances in neural information processing systems*, 2020, 33: 6827-6839.
- [91] Lin J, Morere O, Chandrasekhar V, et al. Deephash: Getting regularization, depth and fine-tuning right[J]. *ArXiv preprint arXiv:1501.04711*, 2015.
- [92] Heckel R, Bölcskei H. Subspace clustering via thresholding and spectral clustering[C]//2013 IEEE International Conference on Acoustics, Speech and Signal Processing. 2013: 3263-3267.
- [93] Torr P H, Nasuto S J, Bishop J M. Napsac: High noise, high dimensional robust estimation-it's in the bag[C]//British Machine Vision Conference (BMVC): vol. 2. 2002: 3.
- [94] Von Luxburg U. A tutorial on spectral clustering[J]. *Statistics and computing*, 2007, 17(4): 395-416.
- [95] Li Z, Liu J, Chen S, et al. Noise robust spectral clustering[C]//2007 IEEE 11th International Conference on Computer Vision. 2007: 1-8.
- [96] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. *Communications of the ACM*, 1981, 24(6): 381-395.
- [97] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in Python[J]. *The Journal of machine Learning research*, 2011, 12: 2825-2830.
- [98] Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers[C]//Computer

北京理工大学本科生毕业设计（论文）

Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16, 2020: 213-229.

致 谢