



单位代码 10006

学 号 21371393

分 类 号 TP391.9

北京航空航天大学
B E I H A N G U N I V E R S I T Y

毕业设计(论文)

基于手持激光扫描仪的地面建筑三维模型构建

学 院 名 称 自动化科学与电气工程学院

专 业 名 称 自动化

学 生 姓 名 汪子涵

指 导 教 师 王江云

2025 年 5 月

北京航空航天大学

本科生毕业设计（论文）任务书

I、毕业设计（论文）题目：

基于手持激光扫描仪的地面建筑三维模型构建

II、毕业设计（论文）使用的原始资料（数据）及设计技术要求：

基于现有的点云预处理和点云分割方法，目标实现点云精细化建模，设计建筑物部件规整和重用方法。

III、毕业设计（论文）工作内容：

手持三维激光扫描仪是无人智能装备场景感知系统设计与验证子系统中必须的设备，本文利用手持式 Slam 扫描仪，生成建筑物的 3D 点云图像，实现真实场景采集系统近景部分精细数据采集任务及场景精细建模任务。在实现过程中建立融合多特征与深度学习的分割模型，提升复杂场景（如植被遮挡、结构异形）下建筑物点云的分割精度与鲁棒性；同时设计几何与语义联合约束的模型重建算法，对于缺失或不完整的外部特征结构进行补充，实现建筑物细节（如墙面、屋顶、门窗）的高保真表达。

IV、主要参考资料：

[1] 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.

[2] Eastman C M. BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors[M]. John Wiley & Sons, 2011.

[3] Grilli E, Menna F, Remondino F. A review of point clouds segmentation and classification algorithms[J]. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2017, 42: 339-344.

自动化科学与电气工程 学院 自动化 专业 210323 班

学生汪子涵

毕业设计（论文）时间： 2025 年 1 月 3 日至 2025 年 6 月 15 日

答辩时间： ____年__月__日

成 绩： _____

指导教师： _____

兼职教师或答疑教师（并指出所负责部分）：

_____系（教研室） 主任（签字）： _____



本人声明

我声明，本论文及其研究工作是由本人在导师指导下独立完成的，在完成论文时所利用的一切资料均已在参考文献中列出。

作者：汪子涵

签字：

时间：2025 年 5 月



基于手持激光扫描仪的地面建筑三维模型构建

学 生：汪子涵

指导教师：王江云

摘 要

在多飞行器智能协同试验平台项目-建模仿真部分的“真实场景采集系统单元”中，手持三维激光扫描仪是无人智能装备场景感知系统设计与验证子系统中必须的设备，其提供的真实场景的数字化建模是算法开发与系统验证的基础。

本论文聚焦于建筑物三维点云的精细化建模需求，通过使用手持式 Slam 扫描仪，生成建筑物的 3D 点云图像，并提出一种基于点云分割的高精度、高效率建筑物精细化建模方法，解决现有方法在复杂场景下分割精度不足、模型几何与语义信息融合困难等问题。在实现过程中建立融合多特征与深度学习的分割模型，提升复杂场景（如植被遮挡、结构异形）下建筑物点云的分割精度与鲁棒性；同时设计几何与语义联合约束的模型重建算法，对于缺失或不完整的外部特征结构进行补充，实现建筑物细节（如墙面、屋顶、门窗）的高保真表达。

本文对建筑物点云精细化建模提供了规整和重用方法，有效地解决了点云的缺失和不完整等结构问题，完成了真实采集系统近景部分精细数据采集及精细建模任务，解决了场景建模粒度低等问题。

关键词：点云分割，点云精细化建模，点云配准



Reconstruction of buildings models using handheld laser scanner

Author: WANG Zihan

Tutor: WANG Jiangyun

Abstract

In the "Real Scene Acquisition System Unit" of the modeling and simulation section within the Multi-Aircraft Intelligent Collaborative Test Platform project, handheld 3D laser scanners serve as essential equipment for the unmanned intelligent equipment scene perception system design and verification subsystem. The digital modeling of real scenes provided by these scanners forms the foundation for algorithm development and system validation.

This paper focuses on the refined 3D point cloud modeling requirements of buildings. By employing handheld Slam scanners to generate 3D point cloud images of buildings, we propose a high-precision and efficient refined modeling method based on point cloud segmentation. This approach addresses existing challenges such as insufficient segmentation accuracy in complex scenarios and difficulties in integrating geometric and semantic information in current methods. During implementation, we establish a segmentation model that combines multi-feature fusion with deep learning to enhance segmentation accuracy and robustness in complex environments (e.g., vegetation occlusion and irregular structures). Simultaneously, we design a model reconstruction algorithm with joint geometric-semantic constraints that supplements missing or incomplete external structural features, achieving high-fidelity representation of architectural details (including walls, roofs, doors, and windows).

This research provides standardized and reusable methods for refined building point cloud modeling, effectively resolving structural issues like missing and incomplete point cloud data, thereby improving the scene perception capabilities of unmanned intelligent equipment systems.

Key words: Intelligent Scene Perception, Point Cloud Segmentation, PointCloud Modeling



目 录

1	绪论	1
1.1	论文背景和意义	1
1.2	国内外研究现状	2
1.2.1	建筑物建模发展概述	2
1.2.2	点云分割发展概述	3
1.2.3	基于深度学习的点云分割技术	4
1.3	研究目标与内容	6
1.4	论文内容结构	7
2	算法基础与技术路线设计	8
2.1	技术路线设计	8
2.2	建筑物点云扫描	8
2.3	点云预处理与特征提取算法	11
2.3.1	点云滤波	12
2.3.2	点云特征提取	15
2.4	滤波算法对比实验	18
2.4.1	点云可视化	19
2.4.2	软件 CCloudCompare 点云滤波实验	19
2.4.3	点云库 PCL 点云滤波实验	20
2.5	点云特征提取	21
2.5.1	提取点云法向量	22
2.5.2	计算点云曲率	22
2.6	本章小结	23
3	基于点云特征的分割算法	24
3.1	点云分割	24
3.1.1	局部几何编码	24
3.1.2	全局上下文聚合	25
3.1.3	注意力加权	26
3.1.4	多尺度特征融合	26



3.1.5	边界优化.....	27
3.2	实验验证.....	27
3.2.1	点云标注.....	27
3.2.2	基于 PointNet++的语义分割.....	29
3.3	本章小结.....	32
4	建筑物点云精细化建模.....	33
4.1	基于聚类的建筑物部件信息提取与分类.....	33
4.1.1	建筑物部件信息定义.....	33
4.1.2	候选对象检测.....	34
4.1.3	K-Means 聚类算法.....	36
4.2	基于点云配准的建筑物部件重用.....	36
4.3	实验验证.....	37
4.3.1	部件信息提取与分类.....	37
4.3.2	部件重用.....	38
4.4	本章小结.....	40
	结论.....	42
	致谢.....	44
	参考文献.....	45



1 绪论

1.1 论文背景和意义

在无人智能装备验证系统中，真实场景的数字化建模是算法开发与系统验证的基础。现有卫星遥感和机载 LiDAR 系统受限于分辨率限制（通常 $>10\text{cm}$ ），难以获取复杂建筑结构、植被分布等细节特征。手持三维激光扫描仪通过搭载高精度惯性测量单元(IMU, Inertial Measurement Unit)和视觉传感器，可实现毫米级点云分辨率采集，特别适用于：建筑物内部结构逆向建模、密集植被覆盖区域三维重建、动态目标运动轨迹追踪和多源数据融合的基准坐标系建立。本研究通过深度融合 Slam、深度学习和计算几何方法，突破传统点云处理流程中人工干预多、计算效率低的瓶颈，为构建"空-地-内"一体化的智能感知系统提供关键技术支撑。预期成果将直接服务于多飞行器协同导航、智能安防等典型应用场景，推动新一代无人系统向高精度、强适应方向演进。

点云重建技术的核心优势在于其高精度和高密度的数据采集能力。相比传统的二维地图或粗糙的三维建模方法，点云数据能够更为精确地重建城市空间的三维结构，涵盖更多的细节信息。点云重建主流的方法可分为 3 类：（1）基于 MVS (Multi-view Stereo)：通过多个视角的激光扫描数据来恢复场景；（2）泊松重建：一种基于隐式表面重建方法，通过求解泊松方程来重建点云的表面；（3）基于深度学习方法：通过 CNN、GAN、GNN 等深度学习方法，可以直接从稀疏或部分点云中生成完整的三维模型。深度学习方法近年来也为点云重建带来了革命性的进展，尤其在自动驾驶和机器人感知等领域有着广泛应用。

现在主要的点云分割技术有三种，基于聚类^[1]的分割方法、基于法向量和曲率的几何分割^{[2][3]}方法和基于深度学习^[4]的点云分割方法。基于法向量和曲率的点云分割方法通过充分利用点云中表面的几何特征来进行高效的分割。法向量提供了每个点的方向信息，而曲率则帮助描述点云表面的局部形状变化，两者的结合可以有效地区分不同的物体和表面。

传统的点云处理技术通过激光雷达、摄影测量或深度相机等技术获取场景丰富的空间信息，然后基于点云的几何空间分布对三维点云进行修复和补充，但是在出现大范围的点云缺失或几何结构较复杂等场景下，存在着较大的局限性^[5]。而基于点云分割的精



细化建模方法可以从三维点云的局部特征出发，对点云的不同区域进行语义标注，实现更加高效、精确的点云精细化建模。克服了传统点云修复方法对于点云模型几何结构和场景的限制，从而拓展了点云修复的适用场景，能够用于更大规模点云数据的建模修复。

当使用传统的点云分割技术如条件随机场时，需要让模型先学习图像特征，再将图像特征转化为深度信息，如点的三维坐标和语义标签，并将这些包含深度信息的二维像素投影为带有语义标签的三维点，实现三维语义分割，但是这种分割方法存在着从二维数据到三维数据的对齐误差，同时对噪声和不完整数据的敏感性较低，从而影响后续的语义匹配。通过使用基于深度学习的三维点云分割技术，能够处理原始的点云数据，提高点云的分割精度；并且此分割方法通过 GPU 加速和并行计算，能够在短时间内处理大规模的点云数据，提升计算效率。

结合了点云分割的点云精细化建模技术可以对三维点云的特征进行精确的分割与提取，并对一些点云特征的缺失进行补充与完善。例如在扫描仪所获取的点云特征中可能存在点云的几何结构或表面特征缺失的情况，此时通过训练深度学习模型如 Pointnet++^[6]完整的建筑物特征会被用作训练数据，模型通过学习点的局部和全局特征，能够捕捉建筑物的几何信息，并根据这些信息进行准确的分类和分割；对于不完整的特征，可以基于深度生成模型（GAN 或自编码器 VAE）^[7]或点云补全网络 PCN^[8]进行点云补全，通过将训练好的分割网络和生成网络结合，从而推测出缺失的部分。

本论文聚焦于建筑物三维点云的精细化建模需求，通过使用手持式 Slam 扫描仪，生成建筑物的 3D 点云图像，并提出一种基于点云分割的高精度、高效率建筑物精细化建模方法，解决现有方法在复杂场景下分割精度不足、模型几何与语义信息融合困难等问题。在实现过程中建立融合多特征与深度学习的分割模型，提升复杂场景（如植被遮挡、结构异形）下建筑物点云的分割精度与鲁棒性；同时设计几何与语义联合约束的模型重建算法，对于缺失或不完整的外部特征结构进行补充，实现建筑物细节（如墙面、屋顶、门窗）的高保真表达。

1.2 国内外研究现状

1.2.1 建筑物建模发展概述

建筑物点云建模是计算机视觉、激光（LiDAR）扫描以及三维建模技术的交叉领域，广泛应用于城市规划、建筑遗产保护、建筑设计、施工管理等领域。在 20 世纪 70 年代，



随着激光雷达(LiDAR)技术的出现,可以将环境中的物体表面转化为一系列的点数据。随后,一些学者将计算机视觉技术和激光雷达技术相结合,应用于建筑物测量领域,进行建筑结构的简单描述和记录。在 20 世纪 90 年代,高精度激光扫描设备如激光扫描的问世,其能捕捉到建筑物的外观及其窗户、门、墙面等细节信息,后面对于点云数据处理算法的研究:如何处理点云噪声、如何进行点云配准等问题的研究使得点云数据的质量大幅提高。

进入 2000 年代,随着计算机视觉和图像处理技术的发展,自动化的点云数据处理技术取得了突破。例如,点云分割、边界、面片生成等技术得到迅速发展,为点云数据向建筑物模型的转换提供了强大的支持。其中一项有代表性的技术就是点云到 BIM^[9] (PCTB) 技术,利用自动化建模软件,点云数据能够被有效地转化为 BIM 模型,从而广泛地应用于建筑物的现代建模、结构监测和历史遗迹保护等领域。近些年来,建筑物建模呈现智能化与跨学科融合的趋势,数字孪生技术也在建筑物点云建模中得到了广泛应用,通过实时的点云数据更新和虚拟模型的同步,为建筑物的生命周期管理提供更加精确和实时的决策支持;多传感器的融合应用也提升了点云建模的精度和细节表现能力,从而提升模型的真实性和准确性。

建筑物点云建模从最初的激光扫描技术起步,经历了多次技术革新与发展。从简单的几何测量到 BIM 技术的结合,再到深度学习、人工智能以及云计算等新兴技术的融合,建筑物点云建模已经成为建筑行业中的重要工具。

1.2.2 点云分割发展概述

传统的点云分割方法主要是依赖于几何特征和人工规则,例如 Rabbani 在 2006 年所提出的基于边缘的分割算法^[10],通过边缘检测勾勒出不同区域的边界,再将边界内的点云进行分组,以获得最终分割结果。常用的局部表面属性包括法线、梯度、曲率或高阶导数等^[11]。这些方法虽然可以快速分割,但是点云数据中经常出现的噪声和点云密度不均匀的情况下可能出现不准确的结果。

基于区域生长^[12]的方法以一个或多个具有特定特征的点(种子点)为起点,然后围绕具有相似特征(如表面方向、曲率等)的临近点生长种子点。由于区域生长方法使用了全局信息,因此比基于边缘的方法更能抵御噪声,然而由于这些方法对初始种子区域的位置和区域边界附近点的法线和曲率估计不准确非常敏感。



后续提出的基于模型拟合进行分段的点云分割方法主要包含霍夫变换（HT）和随机抽样一致算法（RANSAC），其中 HT 用于检测平面^[13]、圆柱体和球体^[14]，RANSAC^[15] 通过从原始点云中随机选取一个子集，然后根据样本拟合一个模型，并通过评估剩余点是否符合该模型来判断其一致性，其能在存在大量噪声的情况下，准确地从点云中提取具有一致性的点。模型拟合方法速度快，对异常值具有鲁棒性，能够有效地对圆柱体、球体和立方体等几何参数简单的形状进行三维检测，深入了解点云样本的几何特性。但对于复杂形状（点云包括多个不规则的几何形状或者混合模型）或全自动实施来说，它还不够完善，因此通过局部描述符^[16]利用表面几何的丰富性提供了更好的解决方案。

有些分割算法基于机器学习方法，如 DBSCAN^[17]、K-means^[18]或均值漂移聚类^{[19][20][21]}。DBSCAN 是一种基于密度的聚类算法，通过对点的密度进行评估来确定簇的边界；K-means 聚类方法将数据点分配到 K 个簇中，使得簇内点的均值最小化；均值漂移聚类是一种基于密度的非参数聚类方法，算法的核心思想是通过在数据点附近计算其“质心”并向该质心方向“漂移”来发现数据的密集区域。但是这些聚类方法在高维数据通常使得数据的分布变得稀疏，距离度量失效，导致聚类结果不佳，同时这些算法在处理大规模点云数据时计算复杂度较高，计算效率较低。

1.2.3 基于深度学习的点云分割技术

近年来，随着 3D 数据采集技术和人工智能的发展，基于深度学习的点云处理受到越来越多的关注，包括三维点云分割，三维点云目标检测，三维点云分类，三维点云重建等。其中，Semantic3D (ISPRS'17)，S3DIS (CVPR'17)，ScanNet (CVPR'17)，NPM3D (IJRR'18)，SemanticKITTI (ICCV'19)等点云分割数据集的提出，极大地促进了点云分割领域的发展。目前，点云分割主要分为三类：语义分割，实例分割，部件分割。

基于点的网络直接作用于不规则点云，然而点云是无秩序的、非结构化的，直接应用标准 CNN 是不可行的，为此，Qi C R 等人开创性地提出了 PointNet，通过学习使用共享 MLP 的逐点特征和使用对称池函数（maxpooling）的全局特征来解决点的无序性以及原始点云旋转不变性差的问题。首先输入为一帧的全部点云数据的集合，表示为一个 $n \times 3$ 的 2d tensor（n 为点云数量，3 对应 xyz 坐标）。输入数据先通过和一个 T-Net 学习到的转换矩阵相乘来对齐，保证了模型的对特定空间转换的不变性。然后通过多次 MLP 对各点云数据进行特征提取后，再用一个 T-Net 对特征进行对齐。接着在特征的各个维

度上执行 maxpooling 操作来得到最终的全局特征。最后，对分类任务，将全局特征通过 MLP 来预测最后的分类分数；对分割任务，将全局特征和之前学习到的各点云的局部特征进行串联，再通过 MLP 得到每个数据点的分类结果。该网络采用了原始点云的输入方式，最大限度地保留了点云的空间特征。但是该网络存在过分关注全局特征而忽略局部特征的问题，没有考虑点与点之间的结构信息，也没有充分考虑到点云密度不均匀所造成的不利影响等问题，难以适应复杂环境。为此提出了 PointNet++在 PointNet 的基础上加入了多层次特征提取结构（hierarchical structure），通过先提取点云的局部特征，然后逐步扩大局部范围，最终得到一组全局的特征，进行分类。虽然有效解决了 PointNet 在采样和特征提取方法上存在的问题，但是该网络没有充分考虑如何有效利用点与点之间的结构信息，且所采用的 K 近邻搜索方法可能使得采样点集中在一个方向上，容易出现标签错分。

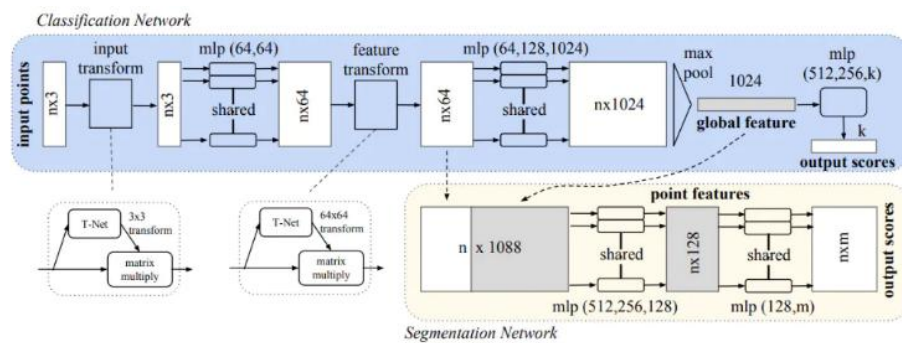


图 1.1 PointNet 网络结构

后来提出了点云的卷积方法,PointCNN^[22]通过引入 X-conv 变换,将邻居点的 feature 矩阵变得与邻居点的顺序无关,从而使得点云变得有序,进行卷积操作。A-CNN^[23]提出了一个环形卷积方法,可以在每个本地环形区域内定义任意大小的卷积核,从而有助于获得对 3D 物体更优的几何表示;同时基于空洞环提出一个新的多级分层结构,从而可以更好地捕捉提取几何细节。PointConv^[24]通过使用 MLP 去学习一个近似的 3D 卷积权重函数,然后去对点云特征进行卷积,最后的实验表明 PointConv 可以取得与图像 CNN 同等水平的学习效果。KPConv^[25]将针对图像的可变形卷积扩展到点云领域,提出了核心点卷积,在以 X 为球心的球体区域内确定若干个核心点,每个核心点带一个权重矩阵,对于落在球体范围内的任意点,用核函数,计算出该点的权重矩阵,用该矩阵对这个点的 feature 进行变换,最后对所有球体内的点进行上述方法,所有 feature 累加起来作为点 X 的 feature。



还有一些基于图像的分割方法，DGCNN^[26]提出了一个新的操作 EdgeConv，可以直接在点云上操作，从而能在保证置换不变性的同时捕获局部几何信息。SPG^[27]不是逐点进行分类，而是将多个点组成的点看作一个完整的整体，对每个点集再进行分类，并且可以描述相邻物体之间的关系，非常适用于上下文分类。GACNet^[28]通过将点云进行图结构转换，用于记录邻域关系，实现点云特征的下采样，局部点云特征提取，以及点云特征的上采样，实现一个类似 Encoder-Decoder 的网络结构；同时通过 Graph Attention Convolution 建立每个点与周围点的图结构，并通过引入注意力机制计算中心点与每个邻接点的边缘权重，从而使得网络能在分割的边缘部分取得更好的结果。

还有一些方法是基于 RNN 网络。如 3D-RNN^[29]构建了一个高效的金字塔池化模型（pyramid pooling module）来提取 3D 点云的局部信息，再通过一个双向的 RNN 提取空间的点云全局依赖性。两个 RNNs 通过不同的方向扫描 3D 空间提取信息，最终达到良好的 3D 语义分割的效果。

本论文拟采用一种混合型网络架构，结合局部几何特征、全局语义上下文与注意力机制，重点优化稀疏区域的分割边界精度。

1.3 研究目标与内容

本论文聚焦于建筑物三维点云的精细化建模需求，通过使用手持式 Slam 扫描仪，生成建筑物的 3D 点云图像，并提出一种基于点云分割的高精度、高效率建筑物精细化建模方法，解决现有方法在复杂场景下分割精度不足、模型几何与语义信息融合困难等问题。在实现过程中建立融合多特征与深度学习的分割模型，提升复杂场景（如植被遮挡、结构异形）下建筑物点云的分割精度与鲁棒性；同时设计几何与语义联合约束的模型重建算法，对于缺失或不完整的外部特征结构进行补充，实现建筑物细节（如墙面、屋顶、门窗）的高保真表达。

主要的研究内容如下：

- 点云预处理与特征提取：针对 Slam 激光扫描仪扫描出来的点云数据进行降噪、稀疏点云补充、坐标系统标准化等操作，构建标准化预处理流程；接着提取点云的几何特征（如曲率、法向量）、语义特征（材质、反射强度）与上下文特征（局部领域关系），为分割提供多维度输入。
- 点云分割算法研究：探究混合型网络架构，融合局部几何特征与全局语义上下



文，提升复杂场景下的分割边界准确性。同时引入注意力机制优化分割结果，针对建筑物的主要特征结构如墙壁、屋顶等进行权重增强，从而解决点云稀疏区域的误分割问题。

- 点云精细化建模：对于分割后的点云子集，深度学习方法可能会在图像识别时出现部件边界不稳定的情况，其次，当场景中存在植物或其他建筑物的遮挡，会导致部分部件无法成功识别和分割，最后的分割结果会出现特征缺失或不完整等情况。此时需要结合先验知识库与模版匹配方法，提出基于聚类 and 线性回归的建筑物部件规整化以及基于语法规则的建筑物部件重用方法，实现语义辅助的局部精细化重建。
- 实验验证与算法优化：通过公开数据集（如 WHU-Urban3D Dataset、Semantic 3D、Paris-Lille-3D、SemanticKITTI 等），与实地采集数据验证系统性能；对比传统的机器学习方法（RanSAC、随机森林）与主流的深度学习方法（PointNet、PointCNN）的性能差异；通过对不同角度的有物体遮挡的局部点云进行分割，不断优化局部精细化建模的方法。

1.4 论文内容结构

本文分为绪论、设计路线与算法基础、点云分割算法研究、精细化建模方法、总结五个章节，内容安排大致如下：

第一章，绪论。阐述了点云精细化建模算法研究的背景与意义，对目前点云分割算法进行了分析，简要介绍了本文的研究内容及结构安排。

第二章，算法基础与技术路线设计。总结了点云滤波与特征提取的算法基础，使用 PCL 库和现有成熟的商业软件 CloudCompare 进行对比，提出了本设计的技术路线。

第三章，点云分割算法研究。通过对预处理后的点云进行语义分割操作，提取建筑物点云的局部部件，建立部件语义库，为后续的精细化建模做出基础。

第四章，建筑物精细化建模算法研究。通过部件规整和重用操作，从而实现复杂场景下的局部点云修复。

第五章，结论。对本设计的内容进行了总结，对未来的相关工作进行了展望。



2 算法基础与技术路线设计

2.1 技术路线设计

基于以上算法基础及实验，对基于点云精细化建模的建筑三维模型重建进行技术路线设计，技术路线如图 2.1 所示，主要包括 3 个部分，分别为点云采集与预处理、点云分割与特征提取和建筑物点云精细化建模。

点云预处理与特征提取：针对 Slam 激光扫描仪扫描出来的点云数据进行降噪、稀疏点云补充、坐标系统标准化等操作，构建标准化预处理流程；接着提取点云的几何特征（如曲率、法向量）、语义特征（材质、反射强度）与上下文特征（局部领域关系），为分割提供多维度输入。

点云分割算法研究：探究混合型网络架构，融合局部几何特征与全局语义上下文，提升复杂场景下的分割边界准确性。同时引入注意力机制优化分割结果，针对建筑物的主要特征结构如墙壁、屋顶等进行权重增强，从而解决点云稀疏区域的误分割问题。

点云精细化建模：对于分割后的点云子集，深度学习方法可能会在图像识别时出现部件边界不稳定的情况，其次，当场景中存在植物或其他建筑物的遮挡，会导致部分部件无法成功识别和分割，最后的分割结果会出现特征缺失或不完整等情况。此时需要结合先验知识库与模版匹配方法，提出基于聚类 and 线性回归的建筑物部件规整化以及基于语法规则的建筑物部件重用方法，实现语义辅助的局部精细化重建。

2.2 建筑物点云扫描

手持式三维激光扫描仪 SLAM-K120 产品采用 Slam 技术（Simultaneous Localization and Mapping 即实时定位与建图技术），可在室内外未知环境使用。通过手持产品在走动中进行自身定位及增量式三维建图，从而获取周边环境的空间三维信息。扫描仪如图 2.2 所示。

扫描仪的相关参数如下：系统精度为 1cm，测量距离为 0.05m~120m，水平视场角为 360°。在进行数据采集时需要提前做好路径规划，确定扫描区域路线和记录起始位置点，且开始和结束位置的轨迹一圈需要形成闭合回环，从而提高后续数据处理的精度。

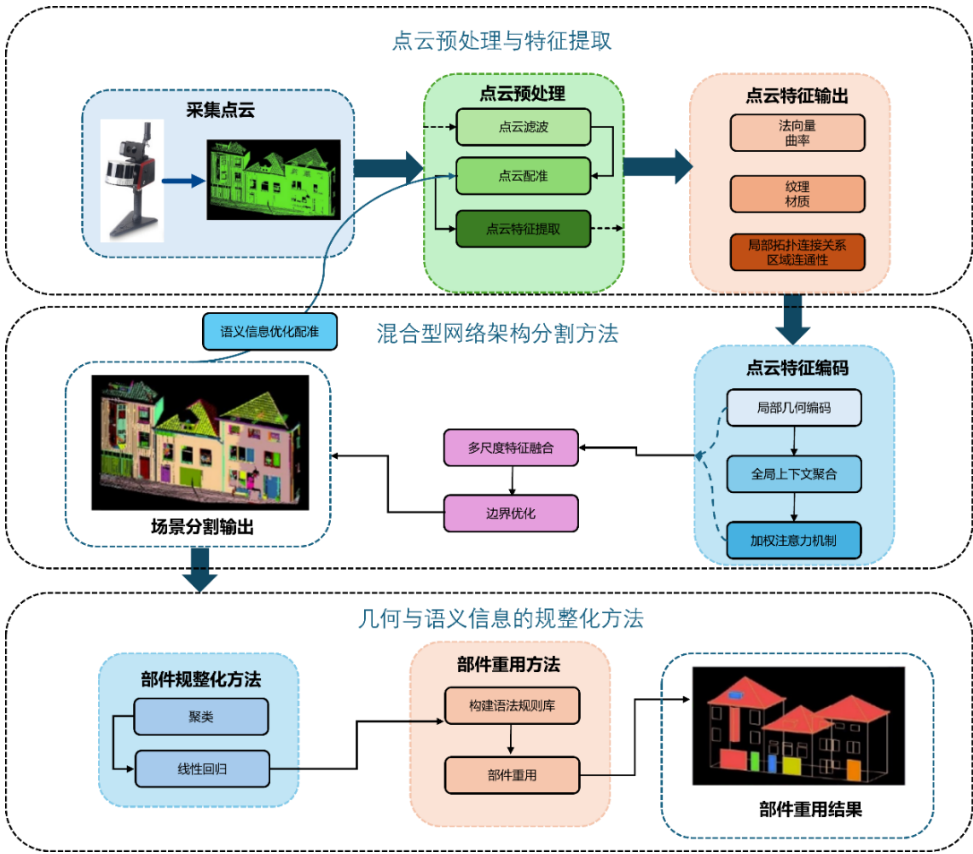


图 2.1 技术路线



图 2.2 手持式三维激光扫描仪 SLAM-K120 实物图

采集完点云数据后，点云可视化的工作在配套软件 KolidaSLAM 中进行。首先进行 Slam 解算，对采集的点云进行生成与处理，设置选项最大/最小距离为参与解算的激光数据的距离阈值，开始消息序号—Slam 解算将从用户设置的位置开始。解算完成后，可以通过调节显示点云帧数大小来浏览点云整体效果。

完成 Slam 解算/优化轨迹后，即可导出 LAS 格式的点云数据，导出时可选择生成彩色点云、精细化处理和进行重采样（抽稀点云数据）三种方式。



同时也可以进行滤波操作，点击滤波设置可以进行滤波参数的调节，如下表 2.1 所示：

表 2.1 滤波参数

滤波参数	参数含义
重采样间距	<ul style="list-style-type: none">- 默认为 0.015m。表示 0.015 立方米有 1 个点云。- 参数调大，点云更稀疏；参数调小，点云更稠密。
多样式阶数	<ul style="list-style-type: none">- 默认为 0.015m。表示 0.015 立方米有 1 个点云。- 参数调大，点云更稀疏；参数调小，点云更稠密。
内插搜索距离	<ul style="list-style-type: none">- 默认为 0.03m（原默认 0.05m，室外可设为 0.05m）。- 表示平滑处理时使用的搜索半径。- 参数调大，点云越平滑；参数调小，平滑效果减弱。
SOR 邻域搜索点数	<ul style="list-style-type: none">- 默认为 100。计算均值和标准差时考虑的邻近点数。- 参数调大，结果更稳定（减少随机噪声误差）、更平滑，但计算时间长；参数调小，更敏感且计算时间快。
SOR 标准差阈值	<ul style="list-style-type: none">- 默认为 0.5。判定噪声的阈值。- 参数调大，保留的点越多（可能剔除不足）；参数调小，剔除的点越多（可能过度剔除）。- 若点云缺失过多，可设置为 1。
半径内插搜索距离	<ul style="list-style-type: none">- 可不勾选，默认为 0.02m。
半径最小临近距离	<ul style="list-style-type: none">- 可不勾选，默认为 2。- 表示若点云在 0.02m 半径内少于 2 个点，则滤除该点。

随后对生成的点云进行可视化预览，软件可采用高程、强度、时间、类别、RGB、回波、文件、高程+强度、分类+强度、时间+强度等多种方式进行赋色，下方图 2.3 为根据类别对点云进行赋色，图 2.4 为根据高程对点云进行赋色。

其中，按高程进行赋色的原理如下：将点云中每个点的高程值(z 坐标)映射到一个连续的颜色带上，首先确定整个点云或感兴趣区域的高程范围（最小值 Z_{\min} 和最大值

Z_{max}), 然后选择一个颜色带, 例如从蓝色 (低海拔) 渐变到绿色、黄色, 最后到红色 (代表高海拔), 最后软件根据每个点的 z 值在其高程范围内的相对位置, 从颜色带中差值计算出对应的颜色 (RGB 值), 并将该颜色赋给这个点。

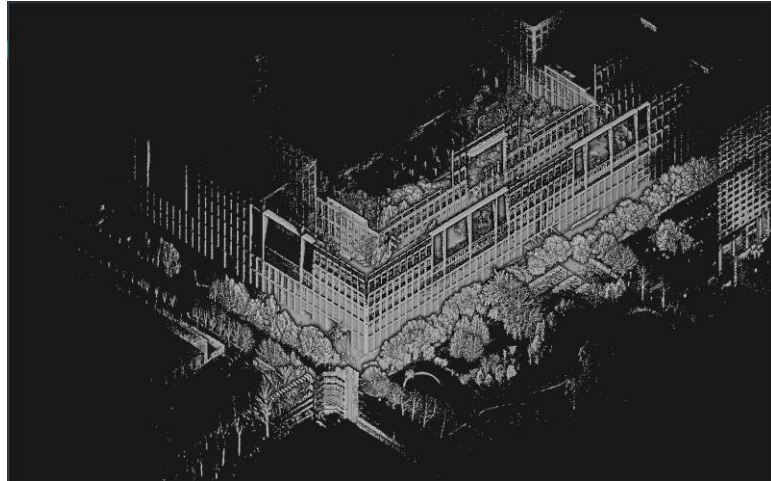


图 2.3 点云按类别进行赋色

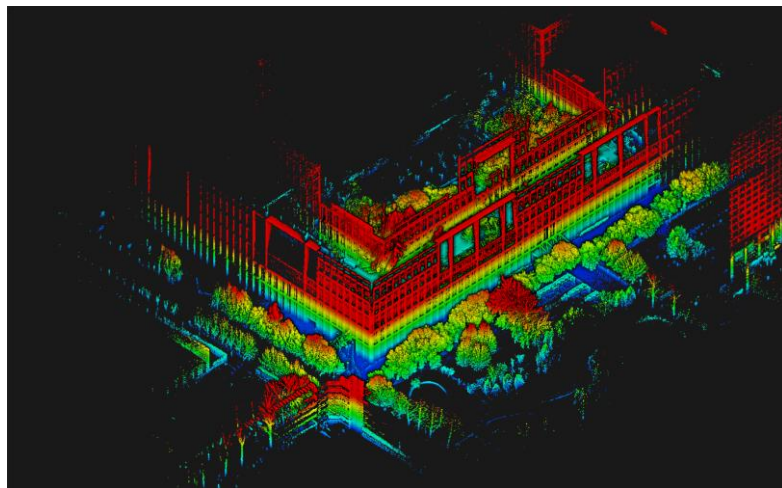


图 2.4 点云按高程进行赋色

2.3 点云预处理与特征提取算法

为了满足真实场景采集系统近景部分精细数据采集任务及场景精细建模任务, 解决场景建模粒度低的问题, 需要保证建筑物点云的质量与精度。点云滤波在此过程中扮演关键角色, 用于去除噪声和离群点, 优化数据, 从而支撑高精度的模型重建。点云预处理与特征提取的整体研究路线如图 2.5 所示。

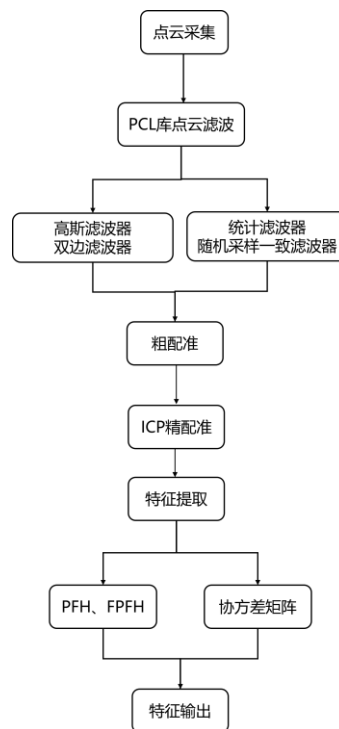


图 2.5 点云预处理研究路线

首先使用 PCL 点云处理库对手持 Slam 扫描仪获取的点云数据进行滤波，利用数据的低频特性剔除离群数据，并进行数据平滑。通常体素滤波器用于对密集点云进行下采样减少数据量，直通和条件滤波器用于处理最前端提取出感兴趣区域，其他滤波器用于平滑点云和去除离散点，因此针对不同的噪声情况需要选择不同的滤波器去处理。点云配准通过 ICP（Iterative Closest Point）算法对齐多视角 Slam 点云，构建全局一致坐标系。

接着在点云多维度特征提取阶段，除了点云的位置坐标，还需要生成一些额外的参数如几何特征—法向量、曲率，语义特征—纹理特征、材质，上下文特征—局部拓扑连接关系、区域连通性等，由此生成带有多维特征标签的点云数据，从而满足分割网络的输入。

2.3.1 点云滤波

对于扫描出来的 3D 点云数据，主要有以下几种情况进行滤波处理：点云数据密度不规则需要平滑、因遮挡等问题噪声的离群点需要去除、数据冗余需要下采样、噪声数据需要去除。对于点之间的距离使用高斯分布进行建模

统计滤波器（Statistical Outlier Removal Filter）^[30]主要用于去除明显的离群点或者测量误差导致的粗差点，滤波的思想为对每个点云计算它到最近的 k 个点的平均距离，通

过假设点云的距离分布是具有均值和标准差的高斯分布，根据给定均值与方差，平均距离在标准范围之外的点，可以被定义为离群点并从数据中去除。算法如下：

- i. 对于每个点 i ，找到它所有的邻居点 j
- ii. 计算所有邻居点到它的距离 $d_{ij}, i = [1, \dots, m], j = [1, \dots, k]$
- iii. 对于点之间的距离使用高斯分布 $d \sim N(\mu, \sigma)$ 进行建模（ μ 表示点距离的平均值 σ 表示点距离的标准差）

$$\mu = \frac{1}{nk} \sum_{i=1}^m \sum_{j=1}^k d_{ij}, \sigma = \sqrt{\frac{1}{nk} \sum_{i=1}^m \sum_{j=1}^k (d_{ij} - \mu)^2} \quad (2.1)$$

- iv. 对于每个点，计算它到它的邻居点的平均距离
- v. 当平均距离位于高斯分布的置信区间以外时，即可移除这些点

$$\sum_{j=1}^k d_{ij} > \mu + 3\sigma \text{ or } \sum_{j=1}^k d_{ij} < \mu - 3\sigma \quad (2.2)$$

高斯滤波（Gaussian Filter）^[31]主要目的是平滑点云数据以去除噪声，基本思想为将每个点周围的点用高斯函数进行加权平均，经过傅里叶变换后仍有高斯函数的特性，令指定区域的权重为高斯分布，从而将高频的噪声点滤除。算法如下：

- i. 每个点的邻域被加权平均，以产生平滑的点云，计算权重的高斯分布函数在三维空间的表示形式如下：

$$G = e^{-\frac{d^2}{2\sigma^2}} \quad (2.3)$$

其中， d 为点云中每个点到邻域点的距离， σ 为高斯函数的标准差。

- ii. 对于每个点，高斯滤波器将其邻域内的所有点乘以一个对应的高斯权重然后将其加权平均。过程公式为：

$$F(x, y, z) = \frac{\sum_{i=1}^N p_i G(|p_i - p_o|)}{\sum_{i=1}^N G(|p_i - p_o|)} \quad (2.4)$$

式中， p_o 为当前点的坐标值； p_i 为邻域内第 i 个点的坐标值； N 为邻域内的点数； $G(p_i - p_o)$ 为第 i 个点的高斯权重，它与第 i 个点与当前点的距离 $p_i - p_o$ 有关； $F(x, y, z)$ 为滤波后点的坐标值。

双边滤波（Bilateral Filter）^[32]主要用于平滑点云数据从而去除噪声，同时保留点云的边缘和细节信息。其主要思想是将每个点周围的点用高斯函数和距离函数进行加权平均，从而达到平滑点云的效果。算法如下：

- i. 对于一个双边滤波器 $Bilateral(p, r, \sigma_d, \sigma_n)$ ，输入参数： p 表示点云中的一个点

$(p \in P)$, r 表示半径, 在此半径内的邻居将被考虑, σ_d, σ_n 表示两个高斯权重, 分别用于空间距离和法线距离的权重; 输出参数: p 表示经过双边滤波处理后的去噪点。

ii. 首先在半径 r 内通过 KdTree 或八叉树加速邻域搜索找到点 p 的邻居, 记为 $N_r(p)$ 。

iii. 接着计算邻居 $N_r(p)$ 的回归平面的局部法向量 n_p , 用于描述邻居的空间排列, 其中拟合平面的选取通常使用主成分分析 (PCA) 或协方差矩阵分解, 选取最小特征值对应的特征向量作为法线。

iv. 初始化变量: 设置累加所有邻域点的总权重 $sum_p = 0$, 累加邻域点沿法线方向的加权位移 $\delta_p = 0$, 这些变量用于累积计算去噪后的点。

v. 遍历所有邻居 q ($q \in N_r(p)$), 执行如下操作:

a) 计算空间距离 d_d , 即点 p 和点 q 的欧几里得距离

$$d_d \leftarrow \|q - p\| \quad (2.5)$$

b) 计算法线投影距离 d_n , 即点 q 到点 p 的向量在法线 n_p 方向上的投影:

$$d_n \leftarrow \langle q - p, n_p \rangle \quad (2.6)$$

c) 计算综合权重 w , 其中 σ_d 是空间距离的标准差:

$$w = \exp - \frac{d_d^2}{2\sigma_d^2} \exp - \frac{d_n^2}{2\sigma_n^2} \quad (2.7)$$

其中空间权重为 $\left(\exp\left(-\frac{d_d^2}{2\sigma_d^2}\right)\right)$, 即距离 p 越近的点权重越大; 值域权重为 $\left(\exp\left(-\frac{d_n^2}{2\sigma_n^2}\right)\right)$, 即法线投影差异越小的点权重越大; 通过双重约束, 仅对空间邻近且法线方向一致的邻域点赋予高权重, 避免夸边缘平滑。

d) 更新位移与权重累加器:

$$\delta_p = \delta_p + w d_n \quad (2.8)$$

$$sum_w = sum_w + w \quad (2.9)$$

vi. 最后通过结合原始点 p 和加权后的值, 计算最终的去噪点 p 。

$$p' \leftarrow p + \frac{\delta_p}{sum_w} n_p \quad (2.10)$$

随机采样一致滤波 (Random Sample Consensus Filter) ^[33]主要目的是去除点云数据中的离群点以及保留点云数据的形状和结构信息。其主要思想是随机选择一组点作为模



型，计算其他点到该模型的距离，从而找到最佳的模型，并将该模型上的点作为局内点（inliers），将其他点作为离群点（outliers）进行删除。算法如下：

i. 参数初始化：

模型类型一定义需拟合的几何模型（如平面、球体等），最小样本数—拟合模型所需的最小数据点数（如平面需 3 个点），迭代次数— N （动态调整或预设固定值），距离阈值— ϵ ，判定点是否为内点的最大允许误差，内点比例阈值— T ，当内点比例超过此值时提前终止迭代。

ii. 随机采样与模型拟合：

假设拟合平面，从点云中随机选择 3 个点，计算平面方程：

$$ax + by + cz + d = 0 \quad (2.11)$$

iii. 内点验证：

对于每个点 $p_i(x_i, y_i, z_i)$ ，计算其到平面的距离：若距离小于阈值 ϵ ，则判定为内点。

$$\text{Distance} = \frac{|ax_i + by_i + cz_i + d|}{\sqrt{a^2 + b^2 + c^2}} \quad (2.12)$$

iv. 评估模型质量：

统计当前模型支持的内点数目 M ，若 M 大于历史最大值，则保存当前模型的内点集合。

$$N = \frac{\log(1-P)}{\log(1-(1-e)^S)} \quad (2.13)$$

v. 动态调整迭代次数：

假设外点比例为 e ，则每次采样均成功找到最优模型的概率为：

$$P = 1 - (1 - (1 - e)^S)^N \quad (2.14)$$

选择内点最多的模型，并输出对应的内点集合（滤波后的点云）

在本文的点云滤波过程中，在主要使用统计滤波器和随机采样滤波器去除点云中的离群点，使用高斯滤波器和双边滤波器平滑点云从而去除噪声。

2.3.2 点云特征提取

常用的点云特征描述与提取算法有：基于协方差矩阵的法线和曲率计算、PFH、FPFH 等。

对于点云数据的几何特征提取，通常对法向量和曲率两个特征进行提取，法向量的提取过程^[34]如下：

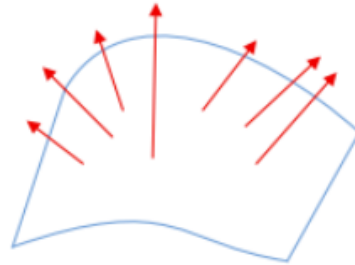
- i. 特征点检测：若某点所在邻域内法向量变化较大（或某一点的法线与其周围邻域点法线的夹角大于给定阈值），则可设该为特征点；
- ii. 给定一点 $p(x_i, y_i, z_i)^T$ ，计算其半径为 r 的球内的邻域点 $p_{i1}, p_{i2}, \dots, p_{ik}$ ；
- iii. 利用式 2.16 和式 2.17 计算该点的单位法向量 n_i ，其中 \bar{p}_i 为邻域点集的三维质心， C 为 p_i 的协方差矩阵；通过特征值分解（EVD）解算出协方差矩阵 C 的三个特征向量和对应的三个特征值，最小的特征值对应的特征向量即为点 p_i 的法向量；计算出的法向量 n_i 规定法向量的方向一致朝向视点方向，可以使用式 2.18 来确定 n_i 的方向。同理可以计算出该点的法向量 $n_{i1}, n_{i2}, \dots, n_{ik}$ ；

$$\bar{p}_i = \frac{1}{k} \sum_{j=1}^k p_{ij} \quad (2.15)$$

$$C = \frac{1}{k} \sum_{j=1}^k (p_{ij} - \bar{p}_i)(p_{ij} - \bar{p}_i)^T \quad (2.16)$$

$$n_i = \begin{cases} n_i, & n_i \cdot (v_p - p_i) > 0 \\ -n_i, & n_i \cdot (v_p - p_i) < 0 \end{cases} \quad (2.17)$$

- iv. 特征点判断：给定阈值 T ，如果所有 $n_{i1}, n_{i2}, \dots, n_{ik}$ 的内积的绝对值都大于 T ，则认为 p_i 不是特征点，反之则为特征点。



（如图为每个点半径为 r 的邻域点的法向量示意图）

图 2.6 法向量示意图

点云曲率的计算可采用协方差矩阵的方法，首先对协方差矩阵 C 进行特征值分解：

$$C = V\Lambda V^T \quad (2.18)$$

（ $\Lambda = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$ ，其中 $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ ）

曲率定义为最小特征值与特征值之和的比值：

$$\sigma = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \quad (2.19)$$

该公式反映了局部表面的平坦程度： $\sigma \approx 0$ ：表面平坦； $\sigma \gg 0$ ：表面弯曲。

PFH^[35]和 FPFH^[36]是点云处理中经典的局部特征描述算法，广泛用于三维配准、目

标识和分割任务。两者通过编码点邻域的几何关系生成直方图特征，但 FPFH 通过简化计算显著提升了效率。

PFH 通过统计点邻域内所有点对的几何关系（法向量角度差异、距离等），构建多维直方图，描述局部表面形状。算法如下：

- i. 邻域选择：对每个查询点 p_q ，选择其邻域 $N(p_q)$
- ii. 点对几何关系计算：遍历邻域内所有点对 (p_i, p_j) ，计算以下四个参数
 - a) 法向量夹角： $\alpha = \arccos(n_j \cdot (u))$ 其中 $u = \frac{p_j - p_i}{\|p_j - p_i\|}$ ， n_j 是点 p_j 的法向量。
 - b) 法向量投影夹角： $\phi = \arccos(n_i \cdot u)$
 - c) 法向量间夹角： $\theta = \arctan\left(\frac{n_j \cdot w}{n_j \cdot v}\right)$ ， 其中 $v = n_i, w = v \times u$
 - d) 距离参数： $d = \|p_j - p_i\|$

iii. 直方图构建：将四个参数 $(\alpha, \phi, \theta, d)$ 分箱（将参数的取值范围划分为多个离散区间），统计联合直方图。从而统计邻域内点对的微分几何属性：

$$\text{PFH}(p_q) = \text{Histogram}\left(\bigcup_{p_i, p_j \in N(p_q)} \{\alpha, \phi, \theta, d\}\right) \quad (2.20)$$

FPFH 通过简化点对关系计算和减少冗余统计，将复杂度从 $O(nk^2)$ 降至 $O(nk)$ ，同时保留主要几何信息，算法如下：

- i. 简化点对计算（SPFH）：
 - a) 仅计算查询点 p_q 与其邻域点 p_j 的几何关系（而非所有点对）。
 - b) 保留三个角度参数 (α, ϕ, θ) ，忽略距离 d 。
 - c) 每个点的 SPFH 特征维度为 $3 \times b$ （如每参数 11 bins，总维度 33）。
- ii. 邻域特征加权聚合：

对查询点 p_q ，将其 SPFH 特征与邻域点的 SPFH 特征按距离权重叠加：

$$\text{FPFH}(p_q) = \text{SPFH}(p_q) + \frac{1}{k} \sum_{p_j \in N(p_q)} \frac{1}{\|p_j - p_q\|} \cdot \text{SPFH}(p_j) \quad (2.21)$$

其中，权重 $\frac{1}{\|p_j - p_q\|}$ 强调近距离点的贡献。

iii. 直方图归一化：

将最终直方图归一化至 $[0, 1]$ 区间，消除尺度差异。最终特征为：



$$\text{FPFH}(p_q) = \text{Histogram}\left(\text{SPFH}(p_q) + \text{Weighted}_{\text{sum}}\left(\text{SPFH}(p_j)\right)\right) \quad (2.22)$$

特征提取通过挖掘点云的几何、语义及上下文信息，为分割网络提供高区分度的输入，显著提升对复杂场景的理解能力与分割精度。

2.4 滤波算法对比实验

这两个库和软件处理的点云格式均为 PCD 格式，手持激光扫描仪采集的数据格式为 LAS 格式，因此需要编写脚本对点云格式进行转化。在脚本中调用 Python 中的点云处理库：Laspy 和 Open3d，首先通过 laspy 读取 LAS 文件，并将其转换为 Numpy 数组，接着用过 Open3d 库构建 PointCloud 对象，其中 points 用于存储 XYZ 坐标的 $N \times 3$ 矩阵，colors 用于存储 RGB 颜色的 $N \times 3$ 矩阵（范围[0,1]），normals 用于存储法线向量，最后输出的 s 点云格式即为 PCD 格式。

使用市面上现有的点云处理软件进行分析，经了解，CloudCompare^[37]是一款开源、跨平台的点云处理软件，以其轻量化、高精度的特性成为建筑精细化建模领域的首选工具。该软件最初由法国 EDF 研发部门开发，专为处理三维点云和网格数据设计，尤其擅长应对手持激光扫描仪生成的海量、高密度点云数据。在建筑建模场景中，CloudCompare 的核心优势体现在其高效的交互式处理能力与多模态数据兼容性。

首先，软件提供实时可视化编辑功能，用户可通过多视口同步显示点云的剖面视图，动态调整迭代最近点（ICP）配准参数（如最大对应距离、采样率等），并实时观察配准效果，实测数据显示，在处理高达 2.4 亿点云数据时，粗配准仅需 2.3 分钟，配准后的均方根误差（RMSE）可控制在 3.2 毫米以内；其次，软件内置丰富的预处理算法，包括基于统计分析的离群点滤波（处理 2.1 亿点云耗时 23 秒，精度损失仅 0.02 毫米）、自适应体素下采样（0.005 米体素尺寸下处理 3.4 亿点云需 41 秒）以及基于邻域半径的法向量估算（支持多线程加速，1.8 亿点计算耗时 2.1 分钟），这些功能可快速消除扫描噪声并优化数据规模。此外，CloudCompare 支持 E57、LAS、PLY 等 20 余种工程常用格式，实测 E57 格式转换误差小于 0.003 毫米，RGB 色彩还原色差 $\Delta E \leq 1.2$ ，确保多源数据融合的准确性。在建模阶段，其 Poisson 表面重建插件（重建深度 12 级时生成 580 万三角面片耗时 9 分钟）与 qHoleFill 孔洞填补工具（修复直径 0.5 米孔洞的曲率连续性误差 < 1 毫米/米）可高效生成结构完整的建筑表面模型。然而，软件在处理超 5 亿点云时存在内存瓶颈（64GB 内存设备实测峰值占用 48GB），通常需通过分块加载或结合 PCL 库进行

Octree 分块处理以降低内存压力。同时，针对算法透明度不足的-5 问题（如法向量估算参数固化），可通过导出数据至 PCL 库^[38]进行二次计算实现优化。通过与 PCL 的协同，CloudCompare 在建筑建模中形成了“交互式预处理-算法深度优化-可视化验证”的全流程解决方案，成为高精度建筑数字化重建的核心技术支撑。

2.4.1 点云可视化

选择手持激光扫描仪获得的建筑物点云数据进行点云预处理，首先选择文件选项卡，然后点击打开点云文件，选择.las 格式，添加要可视化的点云图像，添加的过程中，可以选择点云渲染类型、渲染颜色、点云大小、线条粗细等主要参数，经过对比，发现渲染类型和渲染颜色分别选择强度（intensity）和灰色（grey）显示效果较好，效果如下图

2.7 所示：

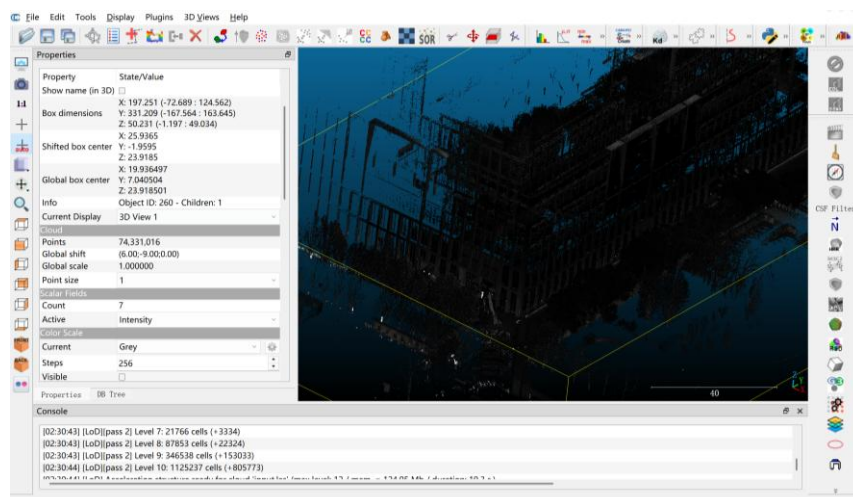


图 2.7 点云可视化

2.4.2 软件 CCloudCompare 点云滤波实验

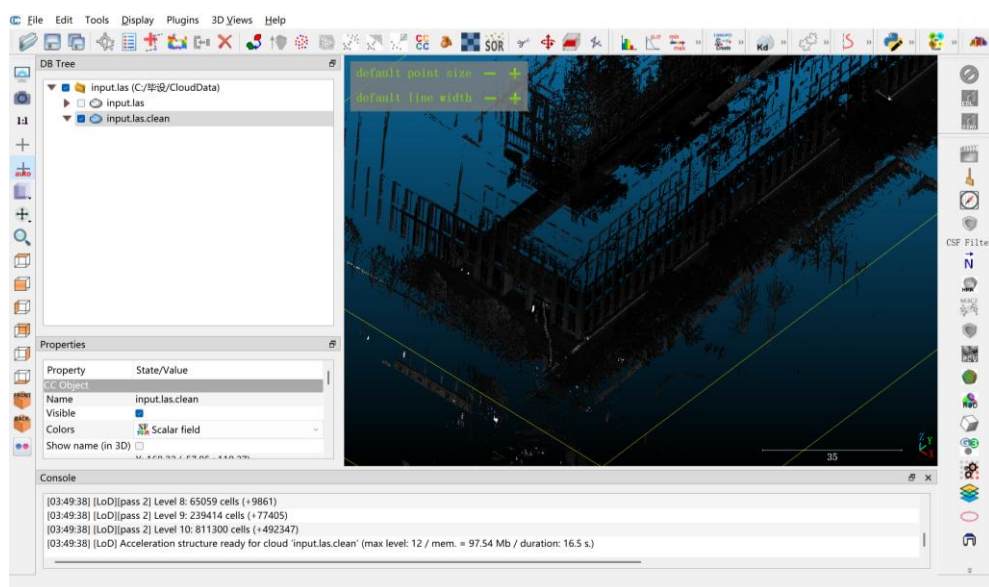
在 CloudCompare 上方 SOR 选项内提供了统计滤波器的接口，点击 SOR 选项卡需要设置近邻点个数和标准差倍数可以选择去除离群点和噪声点两个选项。

参数解释如下：

k_neighbors: k 邻居数，用于计算每个点的平均邻居距离（average distance）的邻居数量；

nSigma: 标准差乘数，用于判断离群点的标准差乘数，离群点定义为距离均值偏离超过一定倍数的点。**SOR** 的实现原理：对于点云中的每个点，找到其 **k_neighbors** 个最近邻点，并计算与这些邻居点的平均距离；对所有点的平均距离计算全局均值和标准差

(std.dev); 如果某点的平均距离超出最大距离（计算公式如下： $\text{max_distance} = \text{average distance} + n\text{Sigma} * \text{std.dev}$ ），则该点被视为离群点，移除这些离群点。对于参数的选择，有如下衡量策略：设定较小的 $n\text{Sigma}$ （比如 1）时，允许的邻居距离变化较小，意味着滤波器会更加严格，会移除更多的点；设定较大的 $n\text{Sigma}$ （比如 2 或 3）时，允许的变化较大，滤波器会更加宽松，只有距离非常大的点才会被移除。通过这种方式，可以有效去除那些不符合局部分布的异常点，从而提高点云的质量。因此对于这种大规模的建筑物点云设定 $k_neighbors$ 和 $m\text{Sigma}$ 分别为 50 和 1.0，有着较好的滤波效果，如下图 2.8 所示，可以较好地滤除噪声点和离群点。



（图 2.7 中滤波前的点云数量为 74331016,；滤波后的点云数量为 67921156）

图 2.8 滤波后建筑物点云

2.4.3 点云库 PCL 点云滤波实验

本文使用 ubuntu 22.04 系统上的 PCL 库进行点云的预处理，数据集为经典数据集 table_scene_lms400，是隶属于斯坦福 3D 扫描仓库的官方数据集采集自一个室内桌面场景，包含桌面上的多个物体（如盒子、杯子等），通过激光雷达（LMS400）扫描生成。主要用于点云去噪、离群点检测等算法的测试。通过调用 PCL 库的 StatisticalOutlierRemoval 滤波器对点云 table_scene_lms400.pcd 进行离群点去除处理，首先读取原始点云并创建滤波器对象，通过 setMeanK(50)设定每个点的邻域分析范围为 50 个最近邻点，利用 setStddevMulThresh(1.0)将离群点判定阈值设为平均邻域距离的 1 倍标准差，执行滤波后保留内点 (inliers) 并保存为 table_scene_lms400_inliers.pcd，随后通

过 `setNegative(true)` 切换滤波模式，提取被剔除的离群点（outliers）并保存为 `table_scene_lms400_outliers.pcd`，从而实现对场景点云中噪声和异常点的高效分离与可视化。滤波结果如下图 2.9 所示：

```
Cloud before filtering:
header: seq: 0 stamp: 0 frame_id:

points[]: 460400
width: 460400
height: 1
is_dense: 1
sensor origin (xyz): [0, 0, 0] / orientation (xyzw): [0, 0, 0, 1]

Cloud after filtering:
header: seq: 0 stamp: 0 frame_id:

points[]: 451410
width: 451410
height: 1
is_dense: 1
sensor origin (xyz): [0, 0, 0] / orientation (xyzw): [0, 0, 0, 1]
```

（从终端中可以显示滤波前的点云数量为 460400，滤波后的点云数量为 451410）

图 2.9 点云滤波效果图

滤波结果可视化如下图 2.10 所示（其中第一张图为原始点云，第二张图为滤出的噪声点，第三张图为滤波后的点云，可以看出 PCL 的 SOR 滤波器对离群点有着较好的去除效果）：



图 2.10 PCL 滤波效果

经过对比，CloudCompare 在对本文采集的 70000000 个点云数据进行滤波操作，处理时间为 10 分钟左右，但是当使用终端直接调用 PCL 库时，对于此数量级的点云数据无法直接处理，因此 CloudCompare 处理大规模点云的精准度和响应速度相较于使用 PCL 库更好，因此在后续的建筑物点云滤波操作中均使用 CloudCompare 进行处理。

2.5 点云特征提取

点云处理软件 CloudCompare 提供了常用的点云特征提取功能，下面使用其对建筑物点云的法向量、曲率进行提取，并进行可视化。

2.5.1 提取点云法向量

通过使用点云处理软件 CloudCompare 对采集到的三维点云数据进行法向量估计。CloudCompare 提供了基于主成分分析 (PCA) 的邻域分析方法来计算每个点的法向量。在提取时设置以下几个参数：

Local surface model: 选择使用最小二乘拟合平面、二次曲面拟合或三角网。

Neighbors->Octree->radius: 局部球体邻域的半径。如果设置的太小(即没有足够的点来计算局部模型),那么法线值将默认为(0,0,1)。如果设置的太大,计算的过程可能会很长,结果会很平滑。

Orientation: 指定一个简单的启发式来设置法线方向(最好是在表面之外)。

Orientation->Use preferred orientation: 使用如下方式设置点云法线的朝向 (+Z 与 Z 轴正方向同向)。

对于点云来说,法线向后的点呈现黑色,法线向前的点呈现白色,如图 2.11 所示。

2.5.2 计算点云曲率

CloudCompare 主要通过分析每个点及其邻域点的局部几何结构来估计曲率。其核心方法通常基于主成分分析 (PCA),并结合邻域点的分布特征来计算不同的曲率度量。

在软件中选择计算表面变化率 (Normal change rate),即最小特征值与特征值之和的比值,进行曲率的计算,可视化的曲率分布如下图 2.12 所示。

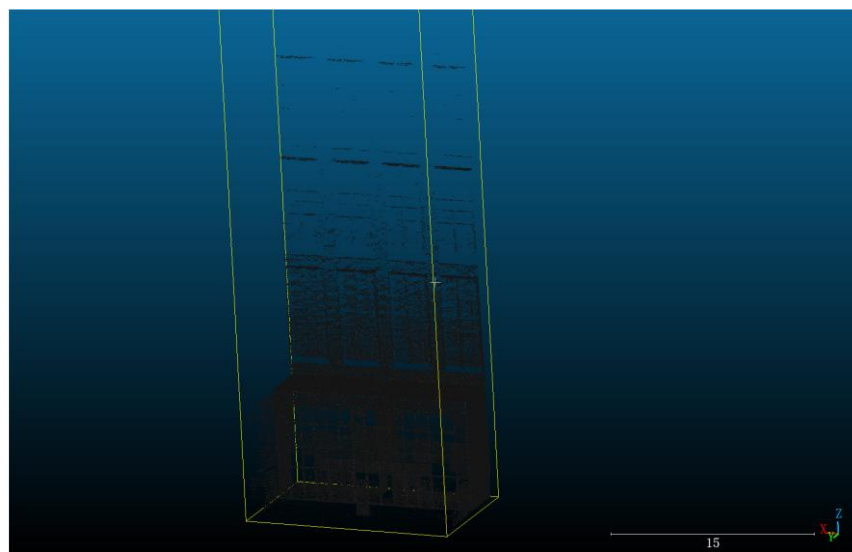


图 2.11 点云法向量计算

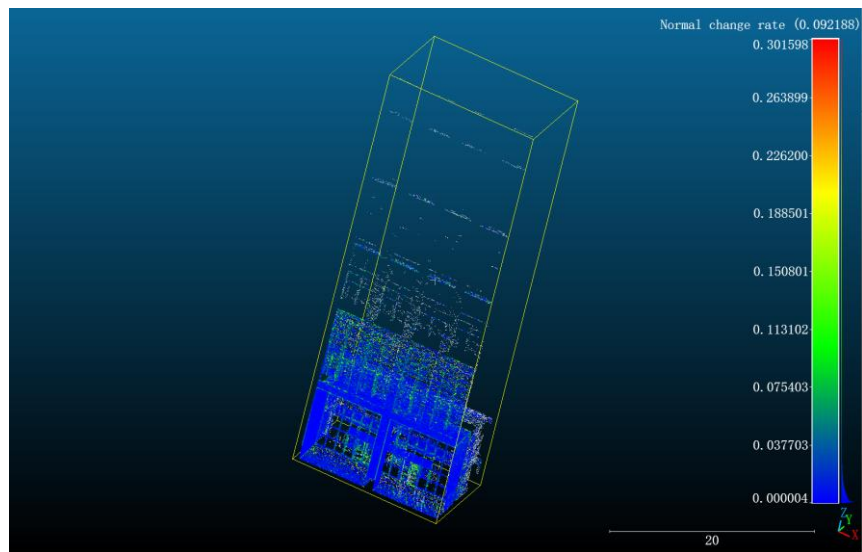


图 2.12 点云曲率计算

2.6 本章小结

本章总结了基本的点云滤波和特征提取算法，通过使用统计滤波器和双边滤波器对点云中的离群点和噪声点进行滤波去除，使得建筑物部件的结构更加清晰、层次分明；通过提取点云法向量、计算点云曲率等特征提取过程，获得点云的几何、语义及上下文信息，为分割网络提供高区分度的输入，显著提升对复杂场景的理解能力与分割精度。实验表明，直接使用 PCL 库对于小批量的点云处理速度较快，易于使用，但是针对本文中扫描的大规模场景点云，由于软件 CCloudCompare 可以使用 GPU 进行加速，因此处理过程较为迅速，在本文后续的点云处理中均使用 CCloudCompare 进行操作。

3 基于点云特征的分割算法

三维点云作为一种重要的几何数据表示形式，能够精确地捕捉物体的三维形状和空间结构，在自动驾驶、机器人导航、文物数字化、城市建模等众多领域展现出巨大的应用潜力。点云分割作为点云处理的关键任务之一，旨在将点云中的每个点划分到具有语义或功能意义的不同类别中，为后续的精细化分析和理解奠定基础。

3.1 点云分割

针对建筑物等复杂场景的点云分割需求，本文提出一种混合型网络架构，结合局部几何特征、全局语义上下文与注意力机制，重点优化稀疏区域的分割边界精度。整体技术路线如下：

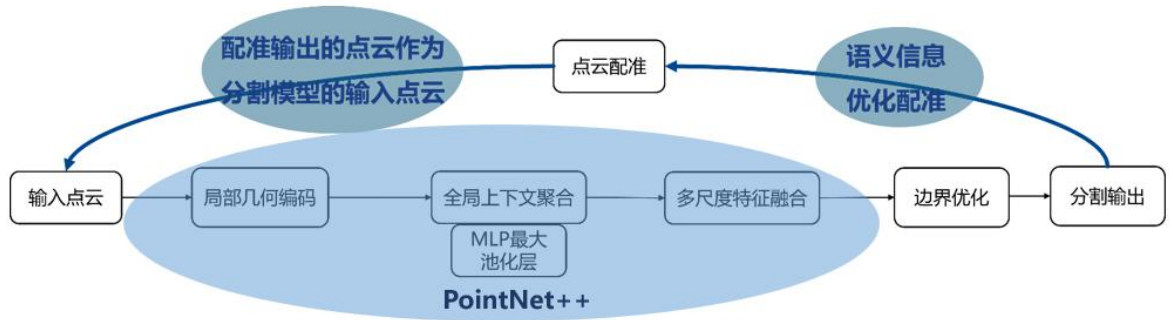


图 3.1 点云分割技术路线

3.1.1 局部几何编码

在本部分将第二章采集的建筑物点云特征数据转化为特征向量的形式，从而可以进行特征的处理和分析。实现方法如下：对上一阶段提取到的局部几何特征进行特征编码，可以采用动态图卷积(DGCNN)[39]描述点云与其近邻点的局部特征关系，其算法如下：

(1) 动态图卷积 (EdgeConv):

原始点云为 $P = \{(x_i, y_i, z_i)\}_{i=1}^N$ ，对于每个点 $P_i = (x_i, y_i, z_i)$ ，选择 K 近邻 $N(p_i)$ ，计算边特征（其中 h_Θ 为共享 MLP）：

$$e_{ij} = h_\Theta([p_i, p_j - p_i]) \quad (3.1)$$

通过最大池化聚合邻域信息：

$$F_i^{\text{local}} = \max_{j \in N(p_i)} e_{ij} \quad (3.2)$$

（2）多层次特征金字塔：

堆叠3层 EdgeConv, 输出特征维度分别为 64、128、256; 每层使用最远点采样(FPS)降低点云分辨率。

3.1.2 全局上下文聚合

本部分需要建立完整的场景语义关系以及建立前后向量权重关系, 通过使用 Point Transformer^[40]网络, 建立点云的长程依赖关系, 捕捉整体语义结构, 算法实现过程如下:

（1）位置编码（Positional Encoding）：

输入为来自局部几何编码的特征 $F_{\text{local}} \in \mathbb{R}^{N \times d}$, 其中 N 为点数, d 为特征维度。将点坐标 $p_i = (x_i, y_i, z_i)$ 映射到高维空间, 与特征拼接, 采用正弦-余弦编码:

$$PE(p_i) = [\sin(p_i W_p), \cos(p_i W_p)] \in \mathbb{R}^d \quad (3.3)$$

最终输入为:

$$F_{\text{in}} = F_{\text{local}} + PE(P) \in \mathbb{R}^{N \times d} \quad (3.4)$$

（2）自注意力机制:

查询、键、值生成, 其中 $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$ 为可学习权重矩阵:

$$Q = F_{\text{in}} W_Q, K = F_{\text{in}} W_K, V = F_{\text{in}} W_V \quad (3.5)$$

注意力权重计算, 其中 Softmax 按行归一化, 表示每个点对其他点的关注程度:

$$A = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \in \mathbb{R}^{N \times N} \quad (3.6)$$

特征聚合:

$$F_{\text{att}} = AV \in \mathbb{R}^{N \times d} \quad (3.7)$$

（3）多头注意力机制:

通过使用多头注意力机制（Multi-head Attention）捕捉不同类型的依赖关系。将 Q, K, V 分割成 h 个头, 每个头独立计算注意力后拼接:

$$\text{MHA}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W_O \quad (3.8)$$

其中 $\text{head}_i = \text{Attention}(QW_Q^i, KW_K^i, VW_V^i), W_O \in \mathbb{R}^{d \times d}$ 为输出投影矩阵。

（4）残差连接与层归一化:

残差连接: 缓解梯度消失, 保留原始特征信息。

$$F_{\text{out}} = F_{\text{in}} + \text{MHA}(Q, K, V) \quad (3.9)$$

层归一化（LayerNorm）：稳定训练过程。

$$F_{\text{norm}} = \text{LayerNorm}(F_{\text{out}}) \quad (3.10)$$

（5）前馈网络（FFN）：

通过 FFN 可以增强非线性表达能力：

$$F_{\text{ffn}} = \text{ReLU}(F_{\text{norm}}W_1 + b_1)W_2 + b_2 \quad (3.11)$$

其中 $W_1 \in \mathbb{R}^{d \times 4d}$, $W_2 \in \mathbb{R}^{4d \times d}$ 为全连接层。

再次残差连接,可以得到整体的点云特征输出：

$$F_{\text{global}} = F_{\text{norm}} + F_{\text{ffn}} \quad (3.12)$$

3.1.3 注意力加权

通过使用通道-空间双注意力(PAConv)^[41]机制动态调整特征通道与空间位置的权重,从而增强关键结构（如墙壁、屋顶）的特征响应,抑制无关噪声的干扰,算法具体步骤如下：

（1）通道注意力（Channel Attention）：

使用 SENet^[42]结构,学习通道加权,其加权特征为：

$$F_{\text{CA}} = w_c \cdot F_{\text{global}} \quad (3.13)$$

（2）空间注意力（Spatial Attention）：

根据曲率 σ_i 和局部密度 D_i ,生成空间权重图,其加权特征如下：

$$F_{\text{SA}} = w_s \cdot F_{\text{CA}} \quad (3.14)$$

3.1.4 多尺度特征融合

点云中的物体具有显著的多尺度性,例如建筑物的屋顶（大尺度结构）与窗户（小尺度细节）。多尺度特征融合的目标是通过整合不同层次的特征,兼顾局部细节与全局语义,提升模型对不同尺度物体的分割能力。具体算法如下 KPConv^[25]：

（1）特征上采样：通过插值或反卷积将高层特征恢复到原始分辨率。

（2）跳跃连接：将不同层级的特征（EdgeConv 输出与 Transformer 输出）拼接。

（3）特征压缩：使用 1×1 卷积降低通道维度。

$$F_{\text{fused}} = \text{MLP}(\text{Concat}(F_{\text{local}}, F_{\text{SA}})) \quad (3.15)$$



3.1.5 边界优化

点云分割的边界区域（如建筑物屋顶边缘、窗户轮廓）常因点云稀疏性或噪声导致分割模糊。传统方法仅依赖语义特征，难以精准定位几何边界。边界优化模块 BADet^[43]通过联合几何边缘检测与语义规则预测，显著提升分割边界精度，尤其在复杂结构（如格栅屋顶、雕花窗框）上效果显著。具体算法如下：

（1）几何边缘检测：

基于第一部分特征提取中获得的曲率 σ_i 和法向量差异 $\|n_i - n_j\|$ 生成边缘掩码 M_{edge} 。

（2）语义-几何联合优化：

定义联合损失函数，使用条件随机场优化输出：

$$P_{\text{final}} = \text{MLP}([F_{\text{fused}}, M_{\text{edge}}]) \quad (3.16)$$

最后输出点云为：

（1）输出头：全连接层+Softmax，生成每个点的类别概率 $P \in \mathbb{R}^{N \times C}$ 。

（2）后处理：通过非极大值抑制（NMS）或形态学操作平滑结果。

3.2 实验验证

3.2.1 点云标注

在软件 CCloudCompare 中使用分割功能对扫描的建筑物点云提取部分结构（如部分门窗等），如下图 3.2 所示。

接着在软件内对提取出来的点云结构进行标注，定义如下标签规则：柱子定义为 0，门定义为 1，落地窗定义为 2，悬空窗户定义为 3，墙体定义为 4。最后将标注的点云数据转换为文本格式（.txt），输出文件如下图 3.3 所示，其四列分别为点云 X，Y，Z 坐标值以及点云的标签类别值。

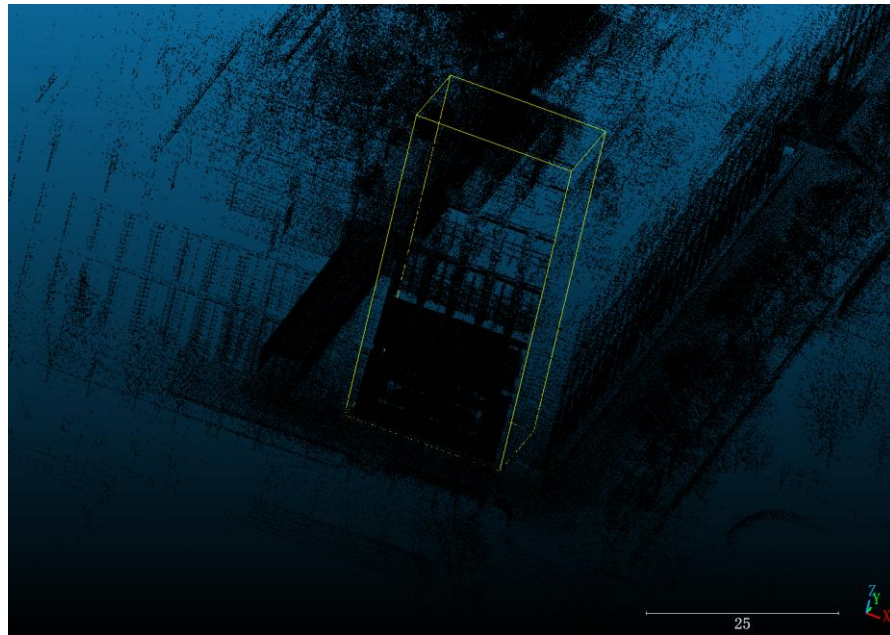
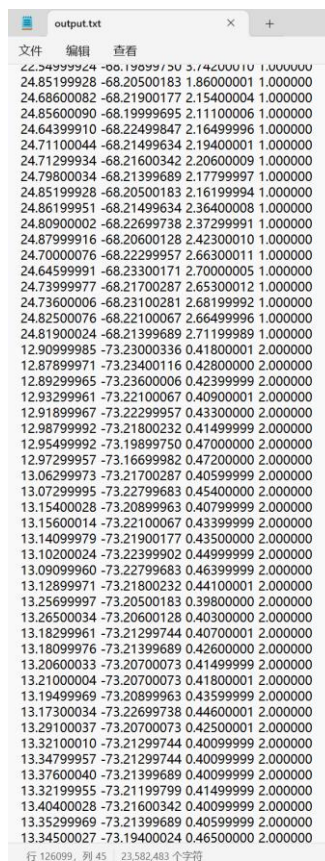


图 3.2 建筑物点云部件提取



（图中四列数据从左到右分别为点云 X, Y, Z 坐标值以及点云的标签类别值）

图 3.3 点云标注 txt 文件



3.2.2 基于 PointNet++的语义分割

PointNet++ 是斯坦福大学研究人员于 2017 年提出的深度学习模型，旨在改进其前身 PointNet，以更高效地处理三维点云数据。点云是一种无序、不规则且密度不均匀的三维数据表示形式（如激光雷达扫描数据），PointNet++ 通过层次化特征学习和局部-全局特征融合，显著提升了点云分类、分割等任务的性能。PointNet++ 的网络结构采用层次化设计，逐层抽象点云特征，每个层级包含三个关键步骤：首先，采样层通过最远点采样（FPS）从输入点云中选取一组均匀分布的关键点（Centroids），作为局部区域的中心；接着，分组层围绕每个关键点构建局部邻域，通过球查询（Ball Query）在固定半径内搜索邻近点，或采用 K 近邻（K-NN）选取最近的 K 个点，并根据点云密度动态调整搜索范围（例如密集区域缩小半径以捕捉细节，稀疏区域扩大半径以覆盖更大范围）；最后，特征提取层利用迷你 PointNet（即共享权重的多层感知机 MLP）对每个局部点集进行编码，并通过最大池化聚合局部特征，从而逐步融合局部结构信息并传递至更高层级，最终形成全局特征表示。这种分层处理机制有效模拟了传统 CNN 的多尺度感受野，显著提升了模型对复杂点云结构的建模能力。

PointNet++在语义分割、点云分类、目标检测等任务中表现优异，本文中测试了 PointNet++在语义分割数据集 S3DIS 进行室内场景的分割的性能。S3DIS(Stanford Large-Scale 3D Indoor Spaces Dataset) 是由斯坦福大学于 2016 年发布的大规模室内三维场景数据集，主要用于三维点云语义分割、场景理解等任务。它是三维计算机视觉领域的重要基准数据集之一。S3DIS 包含 6 个大型室内区域（Area 1~6），覆盖总计约 6,000 平方米的室内空间，涉及办公室、会议室、走廊、休息室等多种场景。其以.txt 格式存储，每行包括 X,Y,Z,R,G,B 及语义标签—每个点被标注为 13 个语义类别（如天花板、地板、墙、桌子、椅子等）。

首先在 ubuntu 上对训练集（Area 1~5）进行训练约 22 小时后得到分割后的数据集，然后在测试集（Area 6）上训练约 10 小时得到最终的评价指标和点云可视化文件.obj，在软件 MeshLab 上对其中的会议室场景可视化后如图 3.4 所示：

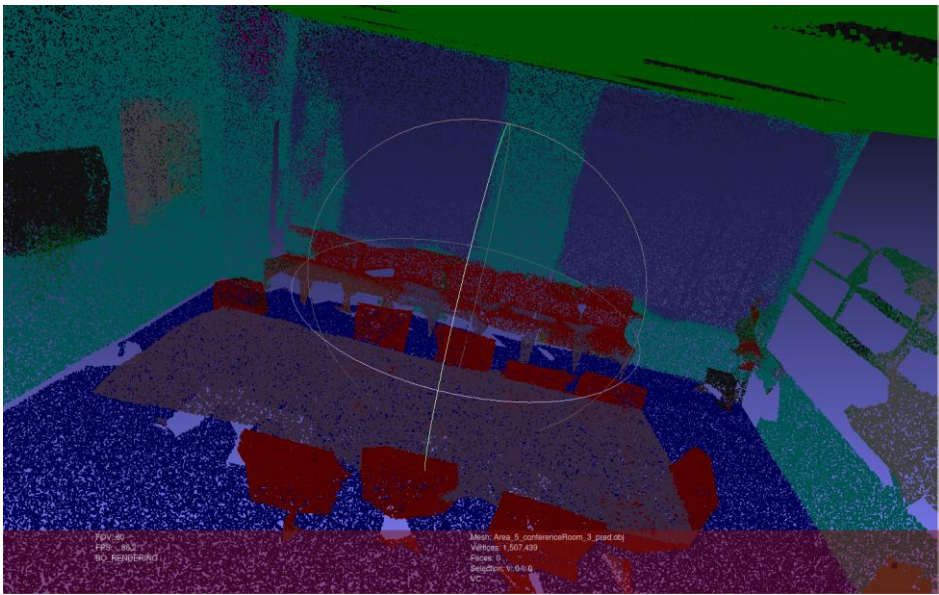


图 3.4 S3DIS 分割效果图

最终的分割评价指标如下：平均类别 IoU（mIoU）— 所有语义类别的平均交并比，反映整体分割精度；平均类别准确率（mAcc）— 表示各类别分类准确率的平均值；整体准确率（OA）— 表示所有点的分类正确率。

表 3.1 类别级 IoU

类别名称	IoU
Floor	0.972
Ceiling	0.890
Wall	0.754
Chair	0.689
Table	0.675
Bookcase	0.627
Window	0.618
Sofa	0.496
Clutter	0.419
Door	0.274
Column	0.046
Beam	0.000



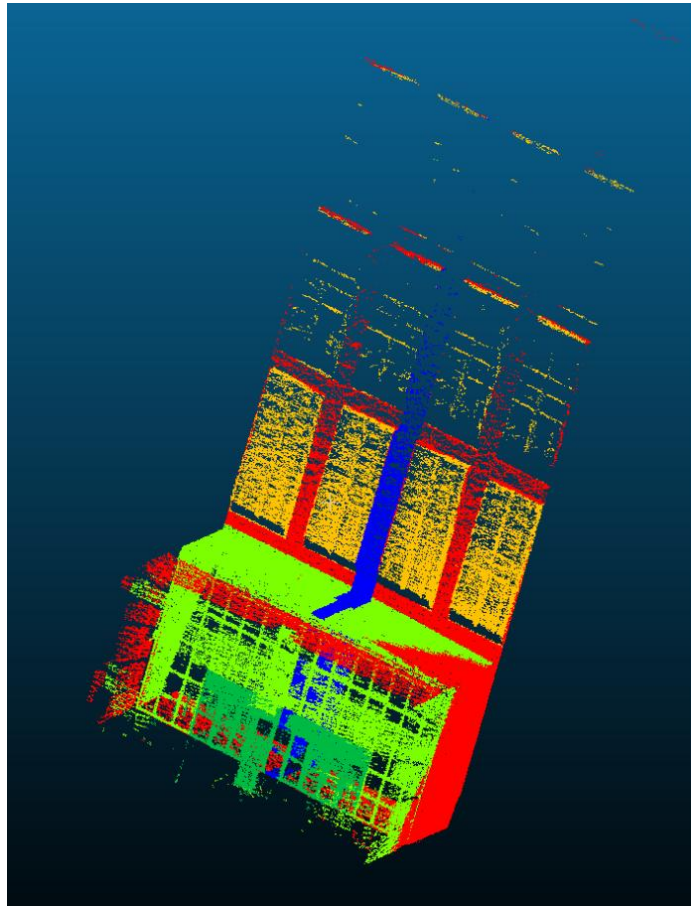
表 3.2 典型场景性能

场景名称	Mean IoU	分析
Area_5_hallway_11	0.873	走廊结构简单，分割效果最佳。
Area_5_office_5	0.807	办公室布局清晰，物体分布均匀。
Area_5_storage_1	0.321	储藏室杂物多、遮挡严重，性能较差。
Area_5_hallway_1	0.359	可能因点云稀疏或复杂布局导致低精度。

接着对 3.2.1 获取的建筑物部分点云进行点云分割实验，修改代码中的训练参数、标签种类数以适配本文的数据，在笔记本 Nvidia 3060 显卡上对训练集训练 20 小时，并在测试集上训练 10 小时后得到的点云分割结果可视化后如下图 3.5 所示，同时输出的点云分割评价指标如表 3.3 所示，可以看出平均交并比（mIoU）为 0.540，表明能够识别出大部分的建筑物部件，但是分割的边缘可能不够精细；总体精度（OA）为 0.836，意味着所有点被正确分类的比例较高，模型能够正确地识别场景中绝大部分的点的类别。针对本文数据较低的 mIoU，主要的原因可能是在场景较高处的建筑物部件点云有所缺失，使得部件的边界不准确，从而导致类型的误判。因此，在第四章的建筑物点云精细化建模部分将进行针对点云的缺失和不完整等问题进行优化，从而辅助更高精度的点云分割。

表 3.3 场景级整体指标

指标名称	值
mIoU	0.540
mAcc	0.639
OA	0.836



（图中浅绿色部分为落地窗，深绿色部分为门，蓝色部分为柱子，黄色部分为悬空窗，红色部分为墙体）

图 3.5 建筑物部件分割效果图

3.3 本章小结

本章给出了基于多头注意力机制的点云分割算法，通过对各个关键建筑物部件进行特征加权，从而提高分割的准确率和精度。在实验部分，通过使用 PointNet++ 对标准数据集和采集的建筑物部件点云进行分割操作，并进行分割指标的评判，可以看到能够较好地根据点云语义标签对建筑物场景进行语义分割，并提取出供后续精细化建模的部件。

4 建筑物点云精细化建模

深度学习算法在进行图像识别和分割部件时会出现部件边界不稳定的情况,导致被分割的同类型部件尺寸和布局不均匀,从而降低了模型的精度;其次,由于深度学习算法存在漏检且场景中存在植物或其他建筑物的遮挡,导致部分部件无法成功识别和分割,从而使得模型完整性不足。这些都会影响建筑物的最终建模效果,因此本部分主要研究建筑物部件语义辅助的局部精细化建模,提出了基于聚类的建筑物部件信息提取与分类以及基于点云配准建筑物部件重用方法。算法流程设计如图 4.1 所示。

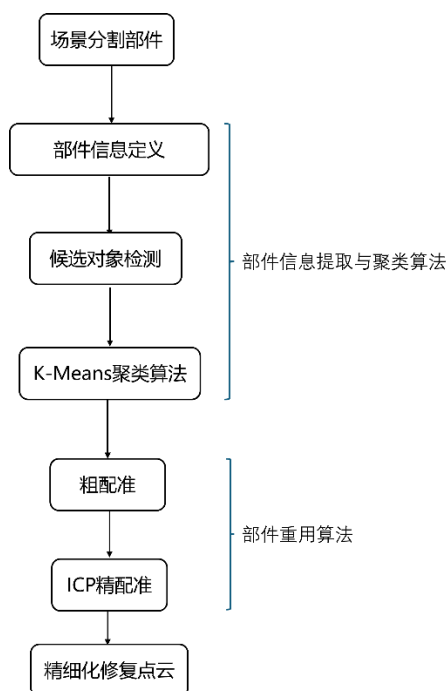


图 4.1 建筑物点云精细化算法流程

4.1 基于聚类的建筑物部件信息提取与分类

要进行建筑物部件的规整化和重用,首先要基于部件特征语义库,再基于聚类算法完成建筑物各部件的特征分类,从而为后续的规整化和重用提供基础和依据。

4.1.1 建筑物部件信息定义

建筑物部件信息定义包括四个要素:部件的高、部件的宽、部件的位置(质心位置和相对位置)以及部件的间隔。具体参数定义如表 4.1 所示。

其中质心坐标是该部件点云四个顶点的平均三维坐标,对于门窗这类规整的长方体



结构，其质心定义在几何中心上；相对位置是指两个相邻的同行（同列）的同一类部件质心之间的距离。

4.1.2 候选对象检测

（1）立面/墙体分割：

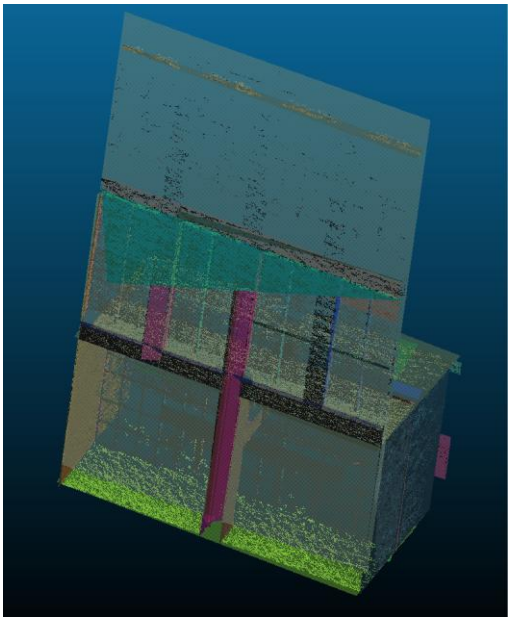
首先使用 CCloudCompare 中的 RANSAC 平面检测拟合并分离出包含门、窗、墙面的建筑物立面或墙体点云，使用方法如下：在软件中使用 RANSAC Shape Detection 插件来自动地检测主要的平面，接着对检测到的平面进行筛选和合并，从而得到较为规整的立面分割图。立面分割效果如图 4.2 所示。

（2）候选部件特征提取：

在上述分割的立面上使用 CloudCompare 的选点工具计算点云坐标和距离，从而提取出部件的特征（H、W、P），提取出来的特征信息如表 4.2 所示。

表 4.1 部件信息定义

部件信息	符号	定义
高	H	部件在垂直方向的上下顶点距离
宽	W	部件在水平方向上的左右顶点距离
位置	P	部件质心坐标
间隔	S	相邻同类部件的质心间距



（如图所示可以分割出落地窗、门、悬空窗、立柱和墙体所在平面）

图 4.2 立面分割效果

表 4.2 部件特征信息

特征 部件	位置(P)/m		
	高(H)/m	宽(W)/m	
落地窗	8.37	15.19	(20.375, -69.365, 4.290)
悬空窗 1	7.50	2.80	(26.684, -78.603, 13.255)
悬空窗 2	7.47	2.78	(22.650, -76.680, 13.225)
悬空窗 3	7.54	2.82	(18.585, -76.813, 13.235)
悬空窗 4	7.48	2.79	(14.518, -76.920, 13.293)
悬空窗 5	7.49	2.83	(26.760, -76.683, 21.640)
悬空窗 6	7.46	2.76	(22.713, -76.678, 21.682)
悬空窗 7	7.51	2.81	(18.618, -76.873, 21.683)
悬空窗 8	7.49	2.78	(14.620, -76.985, 21.680)
门	3.36	9.22	(20.426, -72.575, 1.883)
柱子	8.26	0.95	(20.345, -69.540, 4.290)



4.1.3 K-Means 聚类算法

根据表 4.1 的部件信息定义可知，部件尺寸以及部件位置，仅与部件自身有关，而部件间隔还与部件种类和部件位置有关。因此在本部分对部件信息做分类操作时，将具有相似高度和相似宽度的部件视为同类部件，并将具有相同 X 位置坐标值的部件视为同行部件，将具有相同 Y 位置坐标值的部件视为同列部件。

由于在本文中不同部件的高和宽有着显著差别，因此可将部件的分类问题转化为一个二维 K-Means 聚类^[44]过程，其中 K 代表所需要聚类部件的种类数，点的 X , Y 坐标分别表示部件的 H 和 W 值。首先将上述采集的部件(H, W)信息以 Numpy 二维数组的形式储存，接着使用 K-Means 训练脚本对信息进行聚类，输出结果为 K 个簇的中心坐标与簇标签，此坐标即代表每个部件的特征均值。

对于后续分割出来的部件(H, W)坐标，将其放入已经训练好的 K-Means 模型的 `predict()`函数中，从而计算新数据点到每个已经完成训练的聚类中心的欧氏距离，将该新数据点分配给距离最近的聚类中心所在的簇，同时将簇的标签赋值给新部件，从而完成部件的分类操作。算法步骤如算法 1 所示。

4.2 基于点云配准的建筑物部件重用

在本文激光雷达的扫描过程中，由于会出现树木或其他墙体的遮挡以及超出扫描仪扫描范围等问题，单次采集到的点云只能覆盖目标物表面一部分，会出现点云缺失和不完整的问题，因此为了得到完整的目标点云信息，就需要对缺失部件点云进行多次扫描，并将得到的局部点云数据进行坐标系的刚体变换（旋转矩阵 R 与平移向量 t ），将局部点云统一到全局坐标系，从而实现建筑物部件的重用。

本文将采用精配准算法 ICP（Iterative Closest Point）^[45]对点云进行精配准，其算法核心步骤如下：

- i. 对应点搜索：对源点云中的每个点，在目标点云中找到最近邻点（KD 树加速）。
- ii. 变换估计：基于对应点对，通过最小二乘法计算最优刚体变换（旋转矩阵 R 与平移向量 t ）。
- iii. 迭代更新：应用变换并重复上述过程，直至误差收敛或达到最大迭代次数。

Algorithm 1 基于 K-Means 的建筑物部件信息提取与分类

Input: 建筑物立面或墙体的点云数据 \mathcal{P} , 聚类数量 K (期望的部件种类数)**Output:** 部件分类结果 \mathcal{C} , 每个部件带有类别标签.

- 1: 使用 CloudCompare 中的 RANSAC 平面检测分离出立面/墙体点云 \mathcal{P}_{facade} ;
 - 2: 使用 CloudCompare 选点工具从 \mathcal{P}_{facade} 中提取候选部件的特征 (高度 H , 宽度 W , 位置 P);
 - 3: 将提取的 N 个部件的 (H, W) 信息存储为 NumPy 数组 $\mathbf{D} \in \mathbb{R}^{N \times 2}$, 其中 $\mathbf{D}_{i,:} = [H_i, W_i]$;
 - 4: 初始化 K-Means 模型, 设置聚类数量为 K ;
 - 5: 使用部件特征数据 \mathbf{D} 训练 K-Means 模型;
 - 6: 获取训练后模型的聚类中心 $\mathbf{C} \in \mathbb{R}^{K \times 2}$;
 - 7: 获取每个训练数据点的簇标签 $\mathbf{L} \in \{1, 2, \dots, K\}^N$;
 - 8: 初始化部件分类结果 $\mathcal{C} = \emptyset$;
 - 9: **for** $i = 1$ to N **do**
 - 10: 将第 i 个部件及其对应的簇标签 L_i 添加到分类结果 \mathcal{C} ;
 - 11: **end for**
 - 12: **while** 检测到新的部件 **do**
 - 13: 从新的立面分割中提取新部件的特征 (高度 H_{new} , 宽度 W_{new} , 位置 P_{new});
 - 14: 将新部件的特征 $[H_{new}, W_{new}]$ 输入到已训练的 K-Means 模型的 `predict()` 函数中;
 - 15: 获取新部件的预测簇标签 L_{new} ;
 - 16: 将新部件及其预测的簇标签 L_{new} 添加到分类结果 \mathcal{C} ;
 - 17: **end while**
 - 18: **return** \mathcal{C} ;
-

4.3 实验验证

4.3.1 部件信息提取与分类

本节通过使用 K-Means 算法来对部件信息进行聚类与类别预测, 首先将表 4.2 中部件的 H 、 W 值存储到一个二维 Numpy 数组中, 接着设置聚类的数量为 4, 从而将初始部件按簇的序号从 0 到 3 分为四个类, 接着输入后续分割出的新部件的 (H, W) 坐标, 并进行类别预测, 可以看到能够较为准确地分类出新部件所属的簇, 具体输出结果如下图 4.3 所示。

```
--- 初始聚类结果 ---

簇 0:
  初始部件2, 特征: [7.5 2.8]
  初始部件3, 特征: [7.47 2.78]
  初始部件4, 特征: [7.54 2.82]
  初始部件5, 特征: [7.48 2.79]
  初始部件6, 特征: [7.49 2.83]
  初始部件7, 特征: [7.46 2.76]
  初始部件8, 特征: [7.51 2.81]
  初始部件9, 特征: [7.49 2.78]
簇 1:
  初始部件1, 特征: [8.37 15.19]
簇 2:
  初始部件10, 特征: [3.36 9.22]
簇 3:
  初始部件11, 特征: [8.26 0.95]

--- 新部件预测 ---

后续部件的预测结果:
新部件 新部件1 的特征: 高度=3.5, 宽度=9.1, 所属簇: 2
新部件 新部件2 的特征: 高度=7.8, 宽度=3.0, 所属簇: 0
新部件 新部件3 的特征: 高度=8.0, 宽度=0.9, 所属簇: 3

--- 更新聚类结果 ---

包含所有部件的最终聚类结果:
簇 0:
  初始部件2, 特征: [7.50, 2.80]
  初始部件3, 特征: [7.47, 2.78]
  初始部件4, 特征: [7.54, 2.82]
  初始部件5, 特征: [7.48, 2.79]
  初始部件6, 特征: [7.49, 2.83]
  初始部件7, 特征: [7.46, 2.76]
  初始部件8, 特征: [7.51, 2.81]
  初始部件9, 特征: [7.49, 2.78]
  新部件2, 特征: [7.80, 3.00]
簇 1:
  初始部件1, 特征: [8.37, 15.19]
簇 2:
  初始部件10, 特征: [3.36, 9.22]
  新部件1, 特征: [3.50, 9.10]
簇 3:
  初始部件11, 特征: [8.26, 0.95]
  新部件3, 特征: [8.00, 0.90]
```

图 4.3 部件聚类与新部件预测结果

4.3.2 部件重用

本部分将研究如何修复超出扫描范围或被遮挡的部件，首先在 CloudCompare 中分割出完整的标准部件，接着使用点云对齐功能（Align）来手动选取匹配点对，此工具可以在两个实体中拾取至少三个等效点来对齐两个实体，如图 4.4 所示。将两个点云部件对齐、合并后，可以实现点云的粗配准，如下图 4.5 所示，可以实现第二层落地窗部件的点云补全。

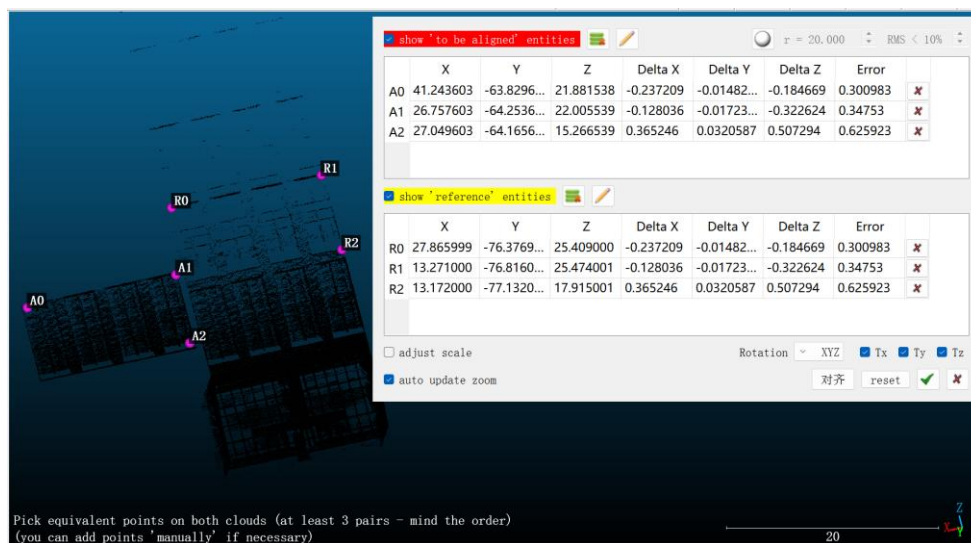


图 4.4 点云对齐

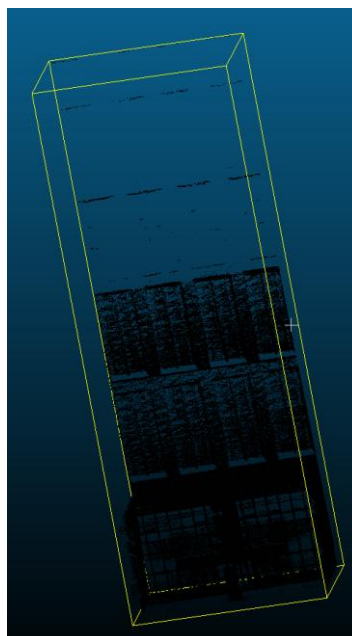


图 4.5 点云粗配准效果图

软件中的点云精配准使用的是 ICP 算法，其主要参数解释如下：Number of iterations—ICP 计算迭代次数；RMS difference—两次迭代误差；Final overlap—设置待配准点云之间的重叠度。点云配准参数设置如下图 4.6 所示。

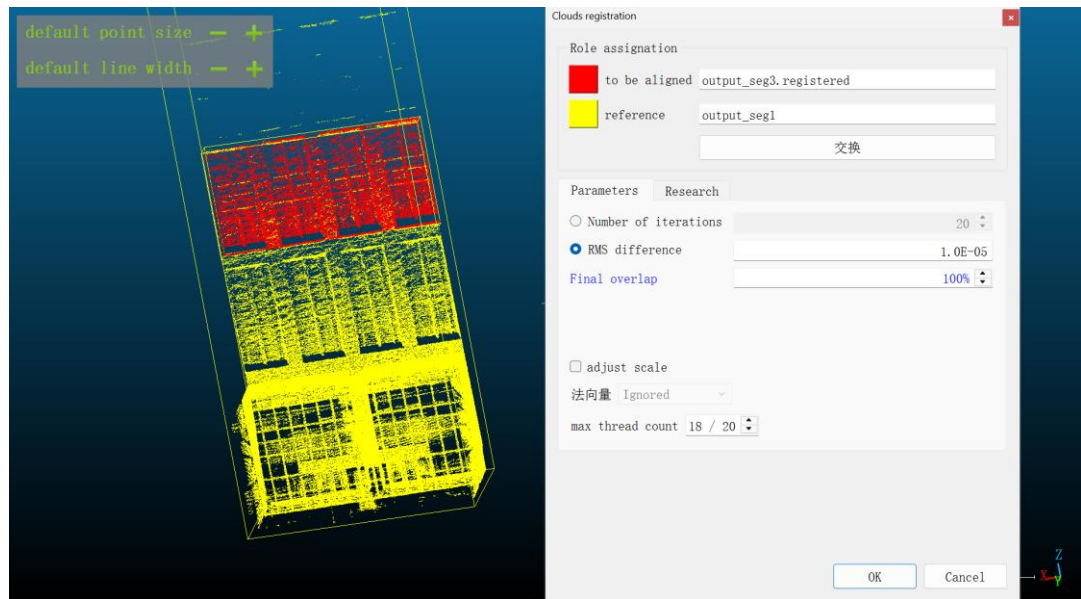


图 4.6 点云精配准参数设置

多次重复上述粗配准、精配准操作，可以为楼层上多层窗户进行补全操作，实现大部分部件的重用操作，接着对点云进行合并，去除重叠点等操作，得到的最终重用点云如下图 4.7 所示。

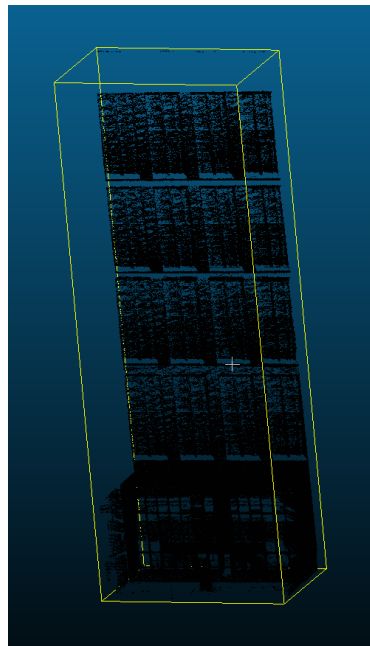


图 4.7 点云精配准

4.4 本章小结

本章基于 K-Means 聚类算法实现建筑物部件信息的提取与分类，并设计了预测新分割部件的算法，能够准确地对新部件进行分类，简化了后续部件重用过程中的部件识别



任务。最后，本文借助 CloudCompare 软件中的点云配准功能，针对场景点云中高楼层窗户的点云缺失问题，较好地实现了点云补全，初步完成建筑物点云精细化建模。



结论

本文实现了基于点云分割的建筑物精细化建模算法的研究，通过对点云进行语义分割，从而获取较为精细的建筑物部件结构与局部特征，接着对建筑物部件进行信息提取与分类，最后对部件进行重用从而补全部件点云，实现真实场景的数字化建模中的建筑物精细化建模。

完成的内容包括：

（1）点云数据采集与预处理

首先本部分通过使用手持激光扫描仪对教学楼的建筑物进行局部扫描，得到原始点云数据；在点云预处理部分，使用基于统计滤波器的点云滤波方法对点云的噪声和离群点进行去除，接着对点云计算法向量和曲率，从而辅助后续的点云分割。

（2）基于混合型网络架构的点云分割算法实现：

本部分给出了基于多头注意力机制的点云分割算法，通过对各个关键建筑物部件进行特征加权，从而提高分割的准确率和精度。在实验部分，通过 PointNet++ 框架对标准数据集和采集的建筑物部件点云进行分割操作，并进行分割指标的评判，可以看到能够较好地根据点云语义标签对建筑物场景进行语义分割，并提取出供后续精细化建模的部件。

（3）建筑物点云精细化建模方法的实现

本部分基于 K-Means 聚类算法实现建筑物部件信息的提取与分类，并设计了预测新分割部件的算法，能够准确地对新部件进行分类，简化了后续部件重用过程中的部件识别任务。最后，本文借助 CloudCompare 软件中的点云配准功能，针对场景点云中高楼层窗户的点云缺失问题，较好地实现了点云补全，初步完成建筑物点云精细化建模。

在以上的研究内容中，本文的主要创新点如下：设计了一套点云预处理流程，有效地对采集的原始点云数据进行格式转换、滤波和特征提取等一系列的点云预处理操作，使得采集的点云可视化效果更加精确和直观；提出了基于注意力机制的点云分割算法，能够进行多尺度特征融合，优化分割精度；借助 K-Means 聚类算法和 ICP 精配准算法，能够有效地对建筑物缺失部件进行补全，实现建筑物点云精细化建模算法。

未来，针对建筑物重建与精细化建模技术的发展，还可以继续探索如下端到端的建模方式：包括本文在内的很多传统点云处理方法存在着参数设置和累积误差等问题，而



一些端到端的大模型方法可以避免此类情况，通过对大语言模型输入准确、特定的指令，能够精确地对点云场景进行各项操作。但是如何构建点云处理库和大语言模型的接口，并精确地设计提示词让大语言模型无误地完成任务，仍是个待优化和改进的地方。



致谢

书写至此，过去四年学习与生活的点点滴滴都在脑海中涌现，有比赛前夕在实验室与小伙伴们的日夜奋战，也有考试周为庞大的知识体系而绞尽脑汁的思索与推理，还有与室友聚会时的轻松与喜悦……这些一幕幕令人难忘的瞬间都交织在过去四年的时间线里，缤纷多彩而又令人难忘，转眼间，也要为本科的学习生涯画上一个句号，开启下一段的旅程。

首先要感谢我的导师王江云老师，在我的科研学习和生活中给予我帮助与支持，她严谨和细致的指导为我指明了未来科研道路上的方向，感谢王老师半年以来的谆谆教导。

感谢指导我的各位老师，在过去的学习时光里慷慨又耐心地给予我支持与帮助，为我排忧解难，鼓励我前进。

感谢我的父母，在二十二载岁月流转中，您们始终以坚定的支持陪伴我成长，迷茫时的理性指引、取得阶段性成果时的客观提醒，构筑了我探索人生道路的坚实基础。感谢我的朋友们，在学习和生活上并肩同行，一起分享欢愉和困难。

最后，感谢评阅本文的各位专家、教授、老师们，感谢您为评阅本文所付出的精力，感谢您提出的每一条珍贵的意见与建议。

大学生活即将结束，而新的征程即将开启。我将带着这份感恩之心，继续努力学习，不断探索，在未来的道路上砥砺前行，不辜负所有关心和支持我的人。



参考文献

- [1] Feng T, Wang W, Wang X, et al. Clustering based point cloud representation learning for 3d analysis[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 8283-8294.
- [2] M. A. Salama, A. E. Hassanien and A. A. Fahmy, "Feature evaluation based Fuzzy C-Mean classification," 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), Taipei, Taiwan, 2011, pp. 2534-2539.
- [3] M. M. Mesmoudi, E. Danovaro, L. D. Floriani and U. Port, "Surface Segmentation through Concentrated Curvature," 14th International Conference on Image Analysis and Processing (ICIAP 2007), Modena, Italy, 2007, pp. 671-676.
- [4] 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.
- [5] WANG Yixian, HU Yufan, KONG Qingqun, ZENG Hui, ZHANG Lixin, FAN Bin. 3D point cloud semantic segmentation: state of the art and challenges[J]. Chinese Journal of Engineering, 2023, 45(10): 1653-1665.
- [6] 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.
- [7] Achlioptas P, Diamanti O, Mitliagkas I, et al. Learning representations and generative models for 3d point clouds[C]//International conference on machine learning. PMLR, 2018: 40-49.
- [8] Yuan W, Khot T, Held D, et al. Pcn: Point completion network[C]//2018 international conference on 3D vision (3DV). IEEE, 2018: 728-737.
- [9] Eastman C M. BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors[M]. John Wiley & Sons, 2011.
- [10] Rabbani, T., Van Den Heuvel, F., & Vosselmann, G., 2006. Segmentation of point clouds using smoothness constraint. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 36(5), pp. 248-253.



- [11]Grilli E, Menna F, Remondino F. A review of point clouds segmentation and classification algorithms[J]. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2017, 42: 339-344.
- [12]Besl P.J., Jain R.C., 1988. Segmentation through variable order surface fitting. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 10.
- [13]Sithole, G. and Vosselman, G., 2004. Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 59(1), pp.85-101.
- [14]Rabbani, T., Van Den Heuvel, F., & Vosselmann, G., 2006. Segmentation of point clouds using smoothness constraint. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 36(5), pp. 248-253.
- [15]Schnabel R, Wahl R, Klein R. Efficient RANSAC for point - cloud shape detection[C]//Computer graphics forum. Oxford, UK: Blackwell Publishing Ltd, 2007, 26(2): 214-226.
- [16]Poux F., Hallot P., Neuville R., R. Billen, 2016. Smart point cloud: definition and remaining challenge. Proc. 11th 3D Geoinfo Conference, 20–21 October 2016, Athens, Greece.
- [17]Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]//kdd. 1996, 96(34): 226-231.
- [18]Kuçak R A, Özdemir E, Erol S. The segmentation of point clouds with k-means and ANN (artificial neural network)[J]. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2017, 42: 595-598.
- [19]Xiao, W.; Zaforemska, A.; Smigaj, M.; Wang, Y.; Gaulton, R. Mean Shift Segmentation Assessment for Individual Forest Tree Delineation from Airborne Lidar Data. Remote Sens. 2019, 11, 1263.
- [20]Ferraz, António, et al. "3-D mapping of a multi-layered Mediterranean forest using ALS data." Remote Sensing of Environment 121 (2012): 210-223.
- [21]Yan, W.; Guan, H.; Cao, L.; Yu, Y.; Li, C.; Lu, J. A Self-Adaptive Mean Shift Tree-Segmentation Method Using UAV LiDAR Data. Remote Sens. 2020, 12, 515.



- [22] Li Y, Bu R, Sun M, et al. Pointcnn: Convolution on x-transformed points[J]. Advances in neural information processing systems, 2018, 31.
- [23] Komarichev A, Zhong Z, Hua J. A-cnn: Annularly convolutional neural networks on point clouds[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 7421
- [24] Wu W, Qi Z, Fuxin L. Pointconv: Deep convolutional networks on 3d point clouds[C]//Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. 2019: 9621-9630.
- [25] Thomas H, Qi C R, Deschaud J E, et al. Kpconv: Flexible and deformable convolution for point clouds[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 6411-6420.
- [26] Phan A V, Le Nguyen M, Nguyen Y L H, et al. Dgcn: A convolutional neural network over large-scale labeled graphs[J]. Neural Networks, 2018, 108: 533-543.
- [27] Xu Q, Zhou Y, Wang W, et al. Spg: Unsupervised domain adaptation for 3d object detection via semantic point generation[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 15446-15456.
- [28] Wang L, Huang Y, Hou Y, et al. Graph attention convolution for point cloud semantic segmentation[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 10296-10305.
- [29] Ye X, Li J, Huang H, et al. 3d recurrent neural networks with context fusion for point cloud semantic segmentation[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 403-417.
- [30] Balta H, Velagic J, Bosschaerts W, et al. Fast statistical outlier removal based method for large 3D point clouds of outdoor environments[J]. IFAC-PapersOnLine, 2018, 51(22): 348-353.
- [31] Han X F, Jin J S, Wang M J, et al. A review of algorithms for filtering the 3D point cloud[J]. Signal Processing: Image Communication, 2017, 57: 103-112.
- [32] Digne J, De Franchis C. The bilateral filter for point clouds[J]. Image Processing On Line, 2017, 7: 278-287.



- [33]Zheng Z, Wang C, Liu H, et al. Adaptive random sample consensus method for ground filtering of airborne LiDAR[C]//Journal of Physics: Conference Series. IOP Publishing, 2023, 2478(10): 102030.
- [34]Mérigot Q, Ovsjanikov M, Guibas L J. Voronoi-based curvature and feature estimation from point clouds[J]. IEEE Transactions on Visualization and Computer Graphics, 2010, 17(6): 743-756.
- [35]Rusu R B, Marton Z C, Blodow N, et al. Persistent point feature histograms for 3D point clouds[M]//Intelligent Autonomous Systems 10. IOS Press, 2008: 119-128.
- [36]Rusu R B, Blodow N, Beetz M. Fast point feature histograms (FPFH) for 3D registration[C]//2009 IEEE international conference on robotics and automation. IEEE, 2009: 3212-3217.
- [37]Girardeau-Montaut D. CloudCompare[J]. France: EDF R&D Telecom ParisTech, 2016, 11(5): 2016.
- [38]Aldoma A, Marton Z C, Tombari F, et al. Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation[J]. IEEE Robotics & Automation Magazine, 2012, 19(3): 80-91.
- [39]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.
- [40]Zhao H, Jiang L, Jia J, et al. Point transformer[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 16259-16268.
- [41]Xu M, Ding R, Zhao H, et al. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 3173-3182.
- [42]Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.
- [43]Qian R, Lai X, Li X. BADet: Boundary-aware 3D object detection from point clouds[J]. Pattern Recognition, 2022, 125: 108524.
- [44]Likas A, Vlassis N, Verbeek J J. The global k-means clustering algorithm[J]. Pattern recognition, 2003, 36(2): 451-461.



-
- [45]Chetverikov D, Svirko D, Stepanov D, et al. The trimmed iterative closest point algorithm[C]//2002 International Conference on Pattern Recognition. IEEE, 2002, 3: 545-548.