

专 业 学 位 硕 士 论 文



论文题目： 面向工业零部件的点云压缩方法研究

专业学位类别

类型

专业（领域）

电子信息硕士

全日制

计算机技术

摘要

随着三维建模与激光雷达技术的持续发展，高质量的点云数据变得更加容易获取。点云表达了物体在三维空间的分布以及各项属性，为工业器件提供了直观且准确的数据表达方式。在工业数据处理与数据传输过程中，使用压缩后的工业器件点云能够极大地提高传输、存储与计算的效率，为后续点云处理工作提供便利。目前，对工业点云的压缩往往采用传统的压缩算法实现，然而使用此类方法会导致工业点云存在特征点大量丢失的问题，并不能对工业点云达到较好的压缩率与重建质量。因此，设计一种适用于工业器件的点云压缩方法颇为重要。于是，本文针对现有方法的不足之处，展开了一些针对性的研究，具体工作如下：

（1）本文提出了一种基于缺陷点分离、面向复杂表面工业产品的点云压缩方法，旨在实现对工业点云的高质量压缩。该方法首先对工业点云使用索引提取法进行缺陷点提取，将原始点云分割为缺陷点云与非缺陷点云。其次通过计算高斯曲率的方式将缺陷点分为三个等级，对缺陷点云使用满足特定规则的曲率分级压缩方法。接着使用改进的跨点前进法对非缺陷点云进行压缩。最后将两部分压缩点云组合获得完整的压缩后点云，从而完成整个点云压缩工作。

（2）本文提出了一种基于优化残差块与逐通道卷积的点云压缩框架，对基于深度学习的点云压缩领域做出研究补充，并在工业点云压缩上进行了应用。该方法设计了一种多尺度端到端的编解码压缩框架，在编码器端使用了优化的下采样模块对当前尺度特征进行初步特征提取，提高了模型的鲁棒性。接着，设计了基于深度可分离卷积的逐通道卷积模块与优化的残差块进行主要的特征提取。然后通过算数编码与八叉树编码结合的形式完成熵编码获取压缩比特流。最后通过多尺度特征重建恢复原始点云。实验结果表明，该方法在公开数据集上相较于MPEG基准算法达到了最高80%的峰值信噪比优化。另外，本文构建了工业零件数据集并使用该框架进行了训练，在压缩工业点云的任务上取得了较好的压缩率、压缩时间与重建质量。

（3）最终，本文构建了一个面向工业点云的压缩系统，并将本文实现的面向复杂表面工业产品的点云压缩方法与经过训练的基于优化残差块与逐通道卷积的点云压缩模型应用于该系统中以验证本文提出方法的有效性与可用性。通过具体的实验操作，证明了本文提出的方法能够有效应用于工业点云的压缩任务。

关键词: 工业点云、点云压缩、高斯曲率、深度学习、卷积神经网络、残差块

Abstract

With the continuous development of 3D modeling and LiDAR technology, high-quality point cloud data has become more easily accessible. Point clouds express the distribution of objects in three-dimensional space and various attributes, providing an intuitive and accurate data representation for industrial devices. In the process of industrial data processing and transmission, using compressed industrial device point clouds can greatly improve transmission efficiency and facilitate subsequent point cloud processing work. Currently, the compression of industrial point clouds often adopts traditional compression algorithms, but using such methods can lead to significant loss of feature points in industrial point clouds and cannot achieve good compression rates and reconstruction quality for industrial point clouds. Therefore, it is quite important to design a point cloud compression method suitable for industrial devices. In response to the shortcomings of existing methods, this article has carried out some targeted research, and the specific work is as follows:

(1) This article proposes a point cloud compression method for complex surface industrial products based on defect point separation, aiming to achieve high-quality compression of industrial point clouds. This method first uses an index extraction method to extract defective points from the industrial point cloud, dividing the original point cloud into defective point cloud and non-defective point cloud. Secondly, the defective points are divided into three levels by calculating Gaussian curvature, and curvature-based hierarchical compression satisfying specific rules is applied to the defective point clouds. Then, an improved cross-point forward method is used to compress the non-defective point cloud. Finally, the two compressed point clouds are combined to obtain the compressed point cloud, thus completing the entire point cloud compression work.

(2) This article proposes a point cloud compression framework based on optimized residual blocks and channel-wise convolutions, making research supplements to the field of point cloud compression based on deep learning and applying it to industrial point cloud compression. This method designs a multi-scale end-to-end codec compression framework. On the encoder side, an optimized down-sampling module is used to initially extract features from the current scale features, improving the robustness of the model. Subsequently, a channel-wise convolution module based on depthwise separable convolution and optimized residual blocks are designed for the main feature extraction. Then, entropy coding is completed by combining arithmetic coding and octree coding to obtain the compressed bit stream. Finally, the original point cloud is recovered through multi-scale feature reconstruction. Experimental results show that this method can achieve up to 80% bitrate savings compared to other benchmark algorithms on public datasets. Additionally, this article constructs an industrial parts dataset and trains it using this framework, achieving good compression rates, compression time, and reconstruction quality for the task of compressing industrial point clouds.

(3) Finally, this article builds a compression system for industrial point clouds and applies the point cloud compression method for complex surface industrial products and the trained point cloud compression model based on optimized residual blocks and channel-wise convolutions proposed in this article to the system to verify the effectiveness and usability of the methods proposed in this article. Through specific experimental operations, it is proved that the methods proposed in this article can be effectively applied to the compression task of industrial point clouds.

Key words: Industrial Point Cloud, Point Cloud Compression, Gaussian Curvature, Deep Learning, Convolutional Neural Network, Residual Block

目录

[第一章 绪论 1](#_Toc166082712)

[1.1 研究背景及意义 1](#_Toc166082713)

[1.2 国内外研究现状 2](#_Toc166082714)

[1.2.1 传统点云压缩方法研究现状 2](#_Toc166082715)

[1.2.2 基于深度学习的点云压缩方法研究现状 4](#_Toc166082716)

[1.3 研究内容 5](#_Toc166082717)

[1.4 本文组织结构 6](#_Toc166082718)

[第二章 相关背景知识介绍 7](#_Toc166082719)

[2.1 点云压缩分类与指标 7](#_Toc166082720)

[2.2 点云压缩基准算法 8](#_Toc166082721)

[2.2.1 PCL点云压缩模型 8](#_Toc166082722)

[2.2.2 MPEG-GPCC模型 10](#_Toc166082723)

[2.2.3 MPEG-VPCC模型 11](#_Toc166082724)

[2.3 三维卷积神经网络 12](#_Toc166082725)

[2.4 深度学习常用模块 14](#_Toc166082726)

[2.4.1 Inception模型 14](#_Toc166082727)

[2.4.2 Residual模型 15](#_Toc166082728)

[2.4.3 熵模型 15](#_Toc166082729)

[2.5 本章小结 16](#_Toc166082730)

[第三章 基于缺陷点分离的工业产品点云压缩方法研究 17](#_Toc166082731)

[3.1 引言 17](#_Toc166082732)

[3.2 方案介绍 18](#_Toc166082733)

[3.2.1 基本思路 18](#_Toc166082734)

[3.2.2 曲率分级法压缩缺陷点云 19](#_Toc166082735)

[3.2.3 跨点压缩法压缩非缺陷点云 21](#_Toc166082736)

[3.3 实验 23](#_Toc166082737)

[3.3.1缺陷点云压缩效果 23](#_Toc166082738)

[3.3.2非缺陷点云压缩效果 24](#_Toc166082739)

[3.3.3对比实验 25](#_Toc166082740)

[3.3.4压缩率评价 26](#_Toc166082741)

[3.3.5重建水平视觉评价 26](#_Toc166082742)

[3.3.6表面积评价 27](#_Toc166082743)

[3.3.7重建水平误差评价 28](#_Toc166082744)

[3.4 本章小结 29](#_Toc166082745)

[第四章 基于优化残差块与逐通道卷积的点云压缩框架研究 30](#_Toc166082746)

[4.1 引言 30](#_Toc166082747)

[4.2 方案介绍 31](#_Toc166082748)

[4.2.1 点云压缩框架 31](#_Toc166082749)

[4.2.2 点云的稀疏卷积 32](#_Toc166082750)

[4.2.3 下采样模块与特征提取模块 33](#_Toc166082751)

[4.2.4 量化与熵编码 35](#_Toc166082752)

[4.2.5 率失真损失函数 36](#_Toc166082753)

[4.3 实验 36](#_Toc166082754)

[4.3.1 模型训练 36](#_Toc166082755)

[4.3.2 对比实验 37](#_Toc166082756)

[4.4 工业数据集的应用 40](#_Toc166082757)

[4.4.1 工业数据集的构建 41](#_Toc166082758)

[4.4.2 实验测试 42](#_Toc166082759)

[4.5 本章小结 43](#_Toc166082760)

[第五章 面向工业点云的压缩系统设计与实现 44](#_Toc166082761)

[5.1 需求分析 44](#_Toc166082762)

[5.2 系统设计 45](#_Toc166082763)

[5.2.1 系统架构设计 45](#_Toc166082764)

[5.2.2 系统详细设计 46](#_Toc166082765)

[5.3 系统功能测试 47](#_Toc166082766)

[5.3.1 格式编辑模块测试 48](#_Toc166082767)

[5.3.2 压缩模块测试 48](#_Toc166082768)

[5.3.3 可视化模块测试 49](#_Toc166082769)

[5.4 本章小结 51](#_Toc166082770)

[第六章 总结与展望 52](#_Toc166082771)

[6.1 总结 52](#_Toc166082772)

[6.2 展望 53](#_Toc166082773)

[参考文献 54](#_Toc166082774)

[附录1 攻读硕士学位期间撰写的论文 57](#_Toc166082775)

[附录2 攻读硕士学位期间申请的专利 58](#_Toc166082776)

[附录3 攻读硕士学位期间参加的科研项目 59](#_Toc166082777)

# 绪论

* 1. 研究背景及意义

随着我国工业的持续发展，工业流水线生产量已经不是评价工业产值的唯一标准，企业对于工业生产的任何一个环节都必须进行合理的安排，例如改进设计误差、生产缺陷、运输损耗等对整个生产流程带来的影响。目前，工业生产零件的质量需求日益提高，优质的工业零件是工业制造业的基础，直接关系到产品的性能和可靠性，在诸如汽车制造、航空航天、机械设备等领域，工业零件的精密性、精确度等质量指标至关重要。因此，对于点云数据的精准采集和分析成为确保工业零件质量的关键一环。通过点云技术[1]，可以实现对工业零件形状、尺寸和表面质量的高精度检测，为工业生产提供了可靠的质量保障和质量控制手段。随着工业智能化和自动化水平的不断提升，点云技术在工业零件质量监测和优化方面的应用将会更加深入和广泛。

点云压缩技术[2]能够有效地减少点云数据的体积，同时保持数据的重要信息。通过采用合适的压缩算法，可以将原始的点云数据压缩为更小的数据集，从而降低数据传输和存储的需求。这种方式不仅可以减少网络带宽的占用[3]，提高数据传输效率，还可以节省存储的空间和成本，对于企业的发展起到关键性的作用。点云压缩技术具有多方面的应用，首先，压缩后的点云数据可以更快速地传输到远程服务器进行处理和分析，通过将点云数据压缩后传输到云端服务器，可以实时监测和分析生产过程中的产品状态和产品质量，及时进行故障诊断和优化调整，提高生产效率。其次，点云压缩技术还可以加快点云数据的处理速度，通过对点云数据进行压缩，可以减少数据处理的时间和计算资源的消耗，提高实时性和响应速度，以便进行后续点云操作如点云补全[4]、点云配准[5]、点云重建[6]等任务，这对于提高工业生产效率具有重要意义。此外，点云压缩技术还可以促进点云数据的共享和合作，通过压缩后的点云数据，不同团队和企业之间可以更方便地进行数据交流和共享，加快创新和合作的进程。由此，工业领域点云压缩技术的重要性不可忽视，在于它能够降低数据传输和存储成本，提高数据处理速度，促进数据共享和合作，为工业生产和创新提供强有力的支持。随着技术的不断进步，点云压缩技术将在工业领域发挥越来越重要的作用。

点云数据是通过激光扫描、雷达[7]等先进技术获取的大量三维点信息。这些点云数据包含了物体的形状、位置、尺寸和表面特征等重要信息。然而，工业零件相较于传统点云测试数据存在数据点集规模庞大、构成复杂、传输和存储成本较高且容易丢失局部细节信息的问题。另外，基于传统算法的点云压缩方法往往只能在规模较小、外形平滑的点云实体进行压缩时取得理想的效果，如果应用于可能存在缺陷、表面复杂的工业零件点云时，往往会产生细节特征信息丢失的问题，对于较为精密的工业领域生产传输并不适用，限制了其在工业领域的应用。目前，在深度学习[8]领域内的点云压缩方法在工业领域并没有广泛应用并且缺乏公开的工业点云数据集作为支持，因此，对于深度点云压缩框架的研究也成为了一项挑战。

* 1. 国内外研究现状

得益于激光雷达技术[7]的发展，高精度点云数据的获取变得越来越容易，对于点云压缩的研究也趋于热门。点云通过大量的点集表达解释了物体的形状特征、空间分布、颜色属性等要素信息，在可视化形式下尤为直观。点云压缩的基本思想是通过降低或消除点云在存储中存在的冗余达到坐标集合信息、形状属性、颜色属性等信息的高效简化，使得点云信息的表达更加紧凑，同时忽略掉不必要的视觉冗余信息。点云压缩通常分为基于传统算法的压缩方法[9]与基于深度学习的点云压缩方法[10]。传统的点云压缩方法通常结合点云的局部特征与整体特征进行重要特征的提取保留，通过聚类或特征子提取的形式进行冗余或噪声的精简。另外也可以通过编码的形式对点云的几何或属性信息进行空间位置编码，将特征信息通过某种结构设计的编码方式表达为连接紧凑、低编码冗余的比特流[11]形式。基于深度学习的点云压缩方法通常使用卷积神经网络[12]的形式对点云的特征进行提取，再通过编码器的形式进行特征向量的编码，最后通过解码重建的形式复原原始点云。深度点云压缩通常构建于相关分类的数据集中，通过不断的模型训练对各种分类的点云数据进行全面的特征提取，满足每个分类的点云数据的高效压缩。目前存在应用于多种点云任务的公开数据集，具体包括适用点云分割与分类的ModelNet40[13]、适用于部件分割的ShapeNet[14]、适用于场景分割的ScanNet[15]与Semantic KITTI[16]等。

* + 1. 传统点云压缩方法研究现状

传统的点云压缩方法通常采用设计算法的方式对点云进行压缩，通常基于点云表面特征、全局特征、聚类化[17]、编码去冗余等形式，该类算法相比于基于深度学习的点云压缩方法是一种较为直观化、轻量级的压缩形式。其中，Botsh[18]等人提出了一种基于八叉树编码[19]的点云几何无损压缩方式，通过八叉树的形式记录点的占用情况而不是直接对坐标信息进行编码，对点云几何数据进行了高效的分层表示，使得每个点的存储消耗远远小于2比特。该方法着重于对点云表面数据进行实验，最终被应用于生产高质量、抗锯齿的重建渲染模型。

Chen等人[20]提出了一种基于KD树[21]的无损点云几何压缩方法。该方法无需对点云数据进行初始量化，通过KD树的形式划分点云几何数据层次，使用最小生成树建模和基于启发式的最优字母表划分法两层改进霍夫曼编码，极大地降低了编码间的冗余，达到了良好的无损压缩形式。

危育冰[22]为了解决规模巨大的散乱点云传输效率低的问题，提出了一种散乱点云的无损点云几何压缩方法。该方法先对点坐标进行了两次量化以便于编码，接着以线性八叉树编码为基础建立了由树根到叶子节点的点云体素占用路径，在此之上使用了排序的差分编码方法，存储首个点、该点邻近点与前一个点的相对位置生成Morton码[23]，便能够达到去冗余的效果，然后使用改进的游程编码去除Morton码中冗余的数码“0”，最后通过算数编码的形式压缩为二进制码流并存储为二进制文件。该压缩方法合理地使用了点云索引进行组织编码，结合了多种编码方式的优势，相较于多种传统压缩算法达到了很高压缩率，但是存在压缩框架复杂、压缩时间较长的缺陷。

Han等人[24]提出了一种基于点法向量的保留边界的点云有损压缩算法。该方法首先使用八叉树建立点的空间拓扑关系，然后采用了特殊的识别方法分割并保留边界点，随后对非边界点进行了法向量的测量，通过计算每个点到其邻域点切平面的欧几里得距离平均值来确定法向量值，接着按照一定的距离大小进行非边界点的削减，最后将完整保留的边界点与非边界点结合得到压缩后的点云。边界点通常是点云特征最密集的地方，该压缩方法采用了特征分离的思想着重保留了密集特征区域，最终实现了较高的点压缩率，这对后续进行其他点云任务极大地降低了计算工作量。

王艺楠等人[25]针对点云数据存在大量冗余的问题，设计了一种基于FPFH特征子结合模糊聚类的有损点云几何压缩算法。该方法首先通过构建点云的FPFH特征子直方图观察几何特征的分布差异，接着使用基于C均值的模糊聚类法将原始点云区分为特征密集点集与非特征点集，最后对两个点集使用了不同大小的比例压缩法进行压缩获取压缩后的点云数据。该方法结合了特征子与聚类算法的思想，能够有效地提取特征点，在点云重建质量方面取得了较好的效果。

张旭康等人[26]针对点云压缩流程中细节特征丢失、点云边缘过度压缩难以恢复的问题设计了一种基于向量相似度的点云有损压缩算法。该方法通过遍历全体点云的坐标数据，计算每个点与坐标原点连线的向量，根据向量长度、向量与坐标值的夹角大小等判定条件将向量相似的点进行聚类，然后设置了相应的阈值对每个聚类空间的点进行采样，构成了能够覆盖原始点云空间的高度压缩后点云数据。该方法实验了斯坦福兔子、头骨和维纳斯这三种公开点云数据模型，相比与其他的聚类算法，该方法在达到类似的压缩率的同时能够较好地保留模型表面特征，实现较好地重建视觉效果。

* + 1. 基于深度学习的点云压缩方法研究现状

在点云压缩领域内，使用深度学习设计点云压缩框架能够更好地对点云特征进行提取，保留重要特征点，实现特定类目下的高效压缩。近年来，国内外研究者对深度点云压缩框架的研究日趋丰富并取得了优秀的成果。Quach等人[27]的研究表明，采用卷积神经网络的方式能够有效地提取点云特征，且随着训练数据集的不断扩大，对相应类目的点云模型的压缩效果也越来越好，这展示了深度学习技术在点云压缩领域内的巨大潜力。

Isik等人[28]将点云的属性进行了空间分块，网络的输入包括了空间坐标和分块信息的潜在向量表示，另外使用了MPEG基于几何的点编解码器GPCC[29]对分块占用的信息进行了RAHT变换，该分块系数是高度可压缩的，损失函数方面采用了拉格朗日损失函数来优化速率失真。该点云压缩框架是第一次采用基于局部坐标的神经网络框架，达到了有效的码率优化。

Gao等人[30]为了精确地表示任意形状的3D物体和相关的空间物理场景设计了一种基于神经图采样（NGS）的深度点云压缩框架。该方法具体通过欧几里得距离，使用*K*邻近算法构造每个点的图分布信息，以目标点的点子集作为输入。在框架结构上，该框架使用了基于端到端分析的VAE编码器，在编码器端采用图采样方式提取邻域结构的潜在关键点，通过3个尺度的图采样最终完成特征提取，接着经过熵编码的形式将特征表示为比特流，最后在解码器端通过多尺度分层卷积逐步重建点云特征。实验结果表明，该框架相比于MPEG的GPCC模型能够在PSNR-D1与PSNR-D2指标上达到几个百分点数量的优化。

Nguyen等人[31]提出了一种基于上下文自适应算数编码的点云几何无损压缩方法。该方法设计了一种新的编码方式，通过将八叉树编码与基于体素的编码相混合。在八叉树域利用了其消除点云间冗余的特性，在体素域中，利用几何信息通过卷积神经网络进行显式处理。得益于以上两种特性，该框架使用了带掩码滤波器的深度卷积神经网络学习体素的概率分布，在ModelNet40数据集上达到了与MPEG的GPCC模型相比28个百分点的速率优化。

Liang等人[32]为了解决体素化方法收集上下文会产生信息缺失的问题，设计了一种基于八叉树的上下文学习框架。该框架采用了八叉树结构对点云结构进行了高效的表示，通过收集兄弟和祖先节点的信息以无损的方式编码八叉树符号序列，随后设计了一个具有较大接受域的条件熵模型来模拟点的上下文关系。具体地，在编码端使用了多头注意力机制对输入的八叉树占用码进行数据增强，随后通过多层感知机建模占用码的分布，最终将串行化的占用码通过算数编码的形式压缩为比特流。该框架在Semantic KITTI数据集上实践了空间场景点云压缩任务，在MVUB数据集上实践了实际物体的点云压缩任务，结果表明该框架在比特率与时间性能上均有明显的优势。

* 1. 研究内容

由于目前的点云压缩方法对于工业领域点云的压缩并没有取得较好的压缩率、时间消耗以及重建质量，针对现有研究方法的不足，本文分别基于传统算法与深度学习的方法各设计了一种点云压缩方法，旨在对工业点云进行高水平的压缩与重建，降低后续工作计算量并提高传输效率。具体来说，本文做出了以下几点研究：

（1）针对表面复杂、可能带有缺陷点的工业点云，本文设计了一种缺陷点分离的工业产品点云几何有损压缩方法。

本方法通过对缺陷点与非缺陷点分别采用了不同的压缩方式。首先对工业点云文件的缺陷点通过索引的形式进行近似提取，随后对于缺陷点采用基于高斯曲率分级的压缩方式，对不同曲率的缺陷点进行了三个等级的分类，并通过特定的保留策略进行了缺陷点的精简压缩，接着对非缺陷点部分采用基于道格拉斯普克算法的改进跨点前进法进行压缩，最后将压缩后的缺陷点点云部分与压缩后的非缺陷点点云部分组合获得完整的压缩点云文件。本方法对缺陷点与非缺陷点进行了区分压缩，使得缺陷点数据能够尽可能地保留，对于平滑的非缺陷点数据做到了冗余数据的筛除，最终达到了较高的压缩率、时间效率以及点云重建质量。

（2）针对传统点云压缩算法目标局限性较高的情况，同时为了实现多类目工业零件点云的特征提取以及特征压缩、弥补点云压缩在深度学习领域研究不足的问题，本文设计了一种基于深度学习的多尺度点云压缩框架。

首先，本方法使用了稀疏卷积的形式对点云数据进行低复杂度的特征表示，在编码器端采用了优化的残差下采样模块、基于深度可分离卷积的逐通道稀疏卷积模块和Inception Residual模块进行特征提取，完成了一个尺度的特征提取，随后采用相同的策略逐步进行下采样特征提取获得了多个尺度的特征总和，接着对获取的点云几何坐标信息以及体素占用特征信息分别采用了八叉树编码和算数编码进行熵编码获取比特流数据。在解码器端采用与编码器端对称的多尺度上采样特征恢复，最终获取了重建后的点云数据。在公开数据集上，本方法的压缩效果相比与其他标准模型有明显的提升。另外，本文还收集了工业零件的网格模型，采用网格点采样的方式进行了工业零件数据集的构建，并且使用本文设计的网络框架进行了训练，最终获得了便于压缩多种类型工业点云零件的深度学习模型。

（3）综合以上研究的两种点云压缩算法，设计并构建了一个面向工业点云的点云压缩系统。该系统采用了这两种方法作为标准算法，为工业零件点云提供了点云压缩、点云重建与点云可视化的功能，最终通过实验测试的形式验证了该系统对工业领域点云压缩任务的有效性与可行性。

* 1. 本文组织结构

本文研究内容共有六个章节，每个章节的概述如下：

第一章为绪论。该章节首先介绍了本文研究的背景及意义，接下来介绍了传统点云压缩和深度点云压缩在国内外的研究现状，然后介绍了本文的主要研究内容，最后总结了全文的整体组织结构。

第二章为相关背景知识介绍。首先介绍了点云压缩的分类与指标，接着详细介绍了点云压缩的几种基准算法包括PCL模型、GPCC模型与VPCC模型，然后介绍了三维卷积神经网络的组成结构，最后介绍了几种常用的深度学习模块。

第三章介绍了一种基于缺陷点分离的工业产品点云压缩方法。该章节首先对本章设计的点云压缩算法流程进行了整体描述，随后对设计的缺陷点与非缺陷点的压缩方法细节分别进行了详细的解释，最后通过实验对比的形式与传统点云压缩算法进行了压缩率、压缩时间、重建表面积变化率与重建误差指标的比较。

第四章介绍了一种基于优化残差块与逐通道卷积的点云压缩框架研究内容。该章节首先展示了本文的深度压缩模型的总体结构并具体解释了其中各个模块的设计思路，接着在公开数据集上将本章模型与其他传统算法和深度压缩框架进行了压缩效果对比，然后阐述了包含多种工业零件实体的工业数据集的构建流程，最后通过使用本章构建的工业数据集进行模型训练验证了该方法在工业零件点云压缩的有效性。

第五章是本文提出的两种点云压缩算法在原型系统中的应用。针对本文提出的基于传统算法与基于深度学习的点云压缩方法设计并实现了一个面向工业点云的点云压缩系统，首先进行了系统的需求分析，接着进行了系统的架构设计与细节模块设计，最后展示了系统的格式编辑、点云压缩与点云可视化模块的实现效果与功能测试。

第六章是本文的总结与展望。本章首先总结并回顾了本文提出的两种压缩方法的设计流程、创新之处以及性能优势，随后指出了各项研究中存在的不足之处，构思了可完善的内容，最后展望了未来可能的研究方向。

# 相关背景知识介绍

本章节首先介绍了点云压缩的分类与指标，然后介绍了几种点云压缩的基准算法，最后介绍了点云压缩在深度学习领域的相关技术。

* 1. 点云压缩分类与指标

点云压缩根据不同的性质可以区分为多个种类。根据信息还原质量的高低可以分为无损压缩[33]和有损压缩[34]。数据是否有损是区分压缩质量的一项重要指标，文字、图像、视频等多种领域均可以通过文件是否有损来进行区分。点云领域内也可以根据文件是否有损来区分，点云有损压缩是一种在压缩时允许删除一定的数据（通常是冗余数据或者噪声数据）来达到文件数据压缩的一种技术，这种技术能够在不影响点云视觉与重建质量的情况下降低点云规模以及点云文件的空间消耗以便于降低计算量与提高传输效率，但是在压缩和解压缩的过程中会丢失一定的信息导致无法完全还原原始文件的状态。点云有损压缩是处理点云的最为常见的一种技术。点云无损压缩常用于对文件数据有完整性要求的场合下，经过压缩与解压能够获得与压缩前完全相同的信息数据，能够保证压缩文件的质量。

根据处理方法的不同可分为基于传统算法的点云压缩方法[9]与基于深度学习的点云压缩方法[10]。基于传统算法的点云压缩方法通常聚焦于点云的空间特征，通过消除点云的空间冗余从而达到较高的压缩率。另外也可以通过点云精简的方式进行冗余点以及非特征点的筛选进行压缩。基于深度学习的点云压缩方法通常采用神经网络的方式，通过计算机学习点云数据的特征向量表示来进行比特流编码，通过编解码器的形式来进行点云特征的学习与表示，该类方法能够更好地适应特定领域的点云压缩需求，具有广阔的发展前景。

根据信息类型的差异可分为几何压缩[34]与属性压缩[35]。点云数据的几何数据通常为点云文件的点坐标表示与法向量的具体数值，属性数据通常为点云各个点的RGB颜色数据。对不同的数据类型进行压缩是区分几何与属性压缩的标准。

点云压缩通常根据点云文件的压缩效率和重建质量来进行压缩质量的评价。压缩率是最主要也是最直观的评价指标，其根据压缩后点云数据文件大小与原始点云文件大小进行比较来得出具体的压缩率数据，压缩率数值越小，则表明点云的压缩效果越好。重建质量可以通过具体的公式计算来量化评价或通过软件进行压缩文件可视化来进行评价，也是一种重要的点云评价指标。本文所设计的一个点云压缩算法则重点考虑了以上两种指标。

在深度学习领域，不同的深度学习模型在不同的损失函数下可能有不同的评价指标，但在较多情况下则是根据MPEG项目的pc\_error文件进行点云的PSNR[36]质量比较。本文设计的深度点云压缩框架则是根据该指标进行压缩质量的评价。

* 1. 点云压缩基准算法

ISO/IEC下属的MPEG工作组目前正致力于点云压缩技术的标准化工作。关于点云压缩（Point Cloud Compression）的讨论始于2013年，当时主要集中在沉浸式呈现的处理上。随着时间的推移，更多的数据和实用场景案例出现，点云压缩的必要性变得愈发明显，于是在2015年，MPEG开始了探索性的实验。首先，专家组实验采用了PCL（Point Cloud Library）[37]这一测试工具，它支持静态和动态的点云对象，并能对几何和特征信息进行压缩。

在前期的研究中，MPEG工作组明确了点云压缩技术在消费电子工业中的需求，并据此成立了MPEG-3DG工作组[38]。为了更好地推进标准化进程，MPEG将点云压缩技术划分为两大类别：基于视频的点云压缩（V-PCC）[39]和基于几何的点云压缩（G-PCC）[40]。这两种方法各有特点，旨在应对不同场景和需求下的点云压缩挑战，MPEG分别提供了两种方法的参考模型TMC2与TMC13[41]。通过这一分类，MPEG工作组希望能够更系统地研究和制定点云压缩标准，为未来的点云应用提供更为高效和可靠的压缩方案。

* + 1. PCL点云压缩模型

点云数据通常包含大量的三维点，每个点都带有位置信息以及可能的颜色、法线等属性。点云的数据量庞大，如果直接存储和处理，将占用大量的内存和计算资源。因此，点云压缩算法的目标就是在不损失过多关键信息的前提下，减少数据的存储量和计算复杂度。基于PCL库的点云压缩算法是处理点云数据的关键技术之一，旨在将高分辨率的点云数据有效地压缩成低分辨率的数据，从而显著减少数据存储所需的空间。PCL库是一个跨平台的开源库，专门用于处理二维、三维图像和点云数据，其包含了众多先进的算法，如滤波、特征估计、表面重建、模型拟合和分割等。

PCL库提供了多种点云压缩算法，这些算法可以分为有损压缩和无损压缩两大类。有损压缩允许在压缩过程中丢失一定的信息，以换取更高的压缩比，而无损压缩则力求在压缩和解压缩过程中保持数据的完整性。在实际应用中，点云压缩算法的选择取决于多个因素，包括点云的规模、密度、应用场景以及对数据精度的要求等。对于大规模、高密度的点云数据，通常采用有损压缩算法以提高处理效率；而对于需要保持数据完整性的场景，无损压缩算法则更为适用。PCL库默认的压缩方式采用了基于八叉树的点云无损压缩方式，另外还有基于体素滤波与基于熵编码[42]的压缩模型。在此基础上，PCL库还提供了对这些算法的灵活配置和优化选项，用户可以根据具体需求调整参数，以在压缩比、重建精度和计算效率之间达到最佳平衡。

八叉树（Octree）[43]是一种用于表示和管理三维空间的树状数据结构，在对点云的空间特征表示尤为适用。它的核心原理是将三维空间递归地分割成八个八分体，每个八分体可以继续分割，以实现对三维空间的更精细的划分，划分过程如图2.1所示。八叉树的每个节点表示一个正方体的体积元素，每个节点有八个子节点，将八个子节点所表示的体积元素加在一起就等于父节点的体积。这种数据结构使得八叉树在解决空间搜索和查询问题时能够达到很高的效率。在八叉树中，每个节点要么有八个子节点，要么没有子节点，也就是说子节点的数量只能是0或8，不会有其他数目，这种特性使得八叉树在表示三维空间时具有很高的灵活性。

**8cubes**

图2.1 八叉树结构划分

八叉树的根节点通常是一个包含所有点云数据的立方体包围盒，通过对这个包围盒进行递归的八等分，可以生成一个完整的八叉树结构。在一次压缩过程中，每个节点的占用信息可以用八个比特来表示，称为空间占用码。这些空间占用码以及最终的叶子节点中包含的点数（当点云中包含重复点时）可以通过熵编码进一步压缩，得到最终的输出码流。一个具体的兔子点云的八叉树划分如图2.2所示。

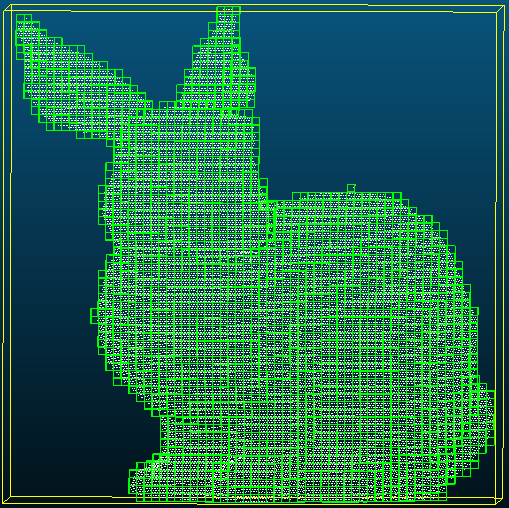
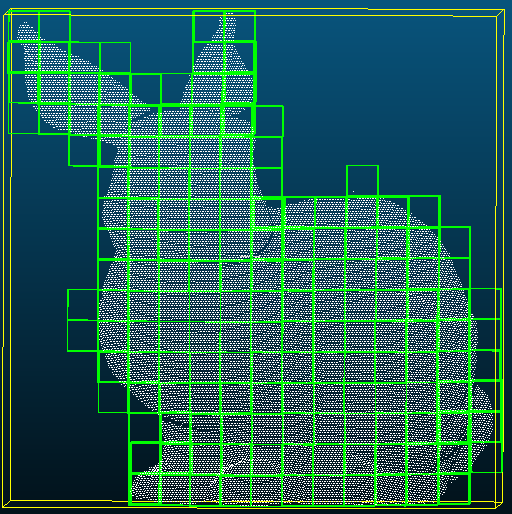
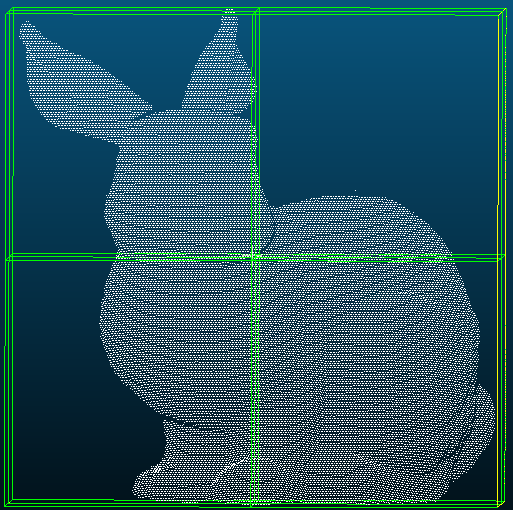


图 2.2 兔子点云的八叉树划分

PCL八叉树压缩法应用非常广泛。八叉树可以有效地实现对点云的空间分区、下采样和搜索操作（如近邻搜索）等。此外，八叉树还可以用于三维场景管理，能够快速地确定物体在三维场景中的位置，检测物体是否与其他物体有碰撞，以及定位物体是否在可视范围内。该方法的优势在于能够更加灵活地处理不同密度的点云，将不同的点云根据需求的深度进行八叉树划分，对子空间进行编码压缩，能够保证较高的压缩比同时减少信息损失。

总的来说，八叉树是一种强大而灵活的数据结构，它在点云空间表示、搜索和查询等方面具有显著的优势，为三维点云压缩提供了有力的工具。

* + 1. MPEG-GPCC模型

GPCC是一种点云压缩模型，由MPEG[29]提供并开发，用于3D点云的压缩。点云是海量点的集合，存储点云不仅会消耗大量的内存，而且不利于传输，因此需要对点云进行压缩。MPEG-GPCC的压缩技术可以广泛应用于机器感知点云和人眼感知点云的各种场景。机器感知点云可以用于自主导航系统、实时巡检系统、地理信息系统、视觉分拣机器人、抢险救灾机器人等场景；而人眼感知点云则可以用于数字文化遗产、自由视点广播、三维沉浸通信、三维沉浸交互等点云应用场景。然而，MPEG-GPCC是一个正在实验的标准化流程，其具体的编解码器设计、测试模型以及性能评估等都在不断发展和完善中。

根据不同的任务需求，MPEG-GPCC能够对点云进行几何压缩与属性压缩。GPCC对点云几何信息压缩通常采用Octree编码或Trisoup编码，对于属性压缩，有区域自适应分层变换（RAHT）[29]、基于插值的分层最近邻预测、基于插值的具有更新或提升变换（Predlift）[44]的分层最近邻预测等方法。GPCC的几何与属性压缩流程如图2.3所示。

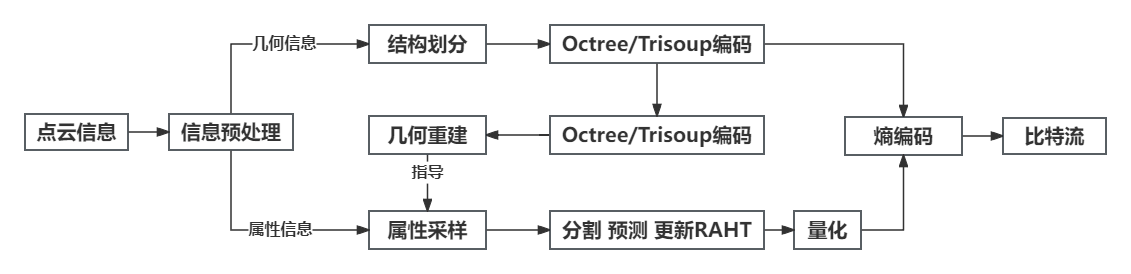


图2.3 GPCC编码流程

在MPEG-GPCC中，点云数据首先进行信息预处理，包括坐标变换与逆变换，为点云数据划分合适的包围盒，将原始空间中的位置坐标转换到包围盒中所在盒子的位置坐标。然后，通过特定的编解码器（Octree或Trisoup）对几何信息进行编码，对属性信息采用RAHT或其他方法进行属性编码。最后将几何编码与量化后的属性编码通过熵编码压缩为比特流。编解码器的具体实现细节可能会根据不同的测试模型和应用场景而有所变化。

官方提供了多种组合的几何——属性点云压缩方式涵盖了Octree几何压缩、Trisoup几何压缩、Predlift属性压缩、RAHT属性压缩等方式，这些压缩组合都有各自的有损与无损实现形式。

* + 1. MPEG-VPCC模型

MPEG工作组的VPCC（Video-Based Point Cloud Compression）[39]压缩模型是一种针对动态点云数据的压缩算法。该模型的核心思想是将3D点云数据通过高效的分割和分方向投影算法，将其投影到2D平面上，进而利用成熟的2D图像压缩工具进行压缩编码。

VPCC模型的主要优势在于其相较于传统的点云压缩算法（如Octree等）在率失真性能上展现出了更好的表现。这主要得益于其结合了视频压缩领域的先进技术，通过投影转换[45]，将3D点云数据的压缩问题转化为更为成熟的2D图像压缩问题，从而实现了更高的压缩效率和更好的重建质量。

在VPCC模型中，点云数据首先被分割成多个部分，每个部分都包含了一定数量的点。然后，这些部分被投影到2D平面上，形成一系列的2D图像。这些2D图像随后被送入成熟的2D图像压缩工具（如HEVC）进行压缩编码。通过这种方式，VPCC模型充分利用了2D图像压缩的成熟技术，实现了对3D点云数据的高效压缩。此外，VPCC模型还考虑了点云数据的动态特性。对于动态点云数据，模型会进一步利用时间上的相关性，通过运动估计和预测等技术，进一步提高压缩效率。

总的来说，MPEG工作组的VPCC压缩模型是一种高效、先进的点云压缩算法，它通过结合视频压缩领域的成熟技术，实现了对3D点云数据的高效压缩和高质量重建，为点云数据的存储和传输提供了有力的支持。

* 1. 三维卷积神经网络

通过广大研究者的实验，卷积神经网络在图像领域取得了优异的成果。三维卷积神经网络（3D Convolutional Neural Networks，3D-CNN）[46]是卷积神经网络（CNN）在三维数据处理领域的一种扩展，相比于原始2D卷积仅仅考虑2D图片空间信息，三维卷积神经网络在输入端多了一个维度，由二维输入的变为了三维的。如图2.4所示，不同于二维卷积，原始三维数据通过维度为3的卷积核进行卷积运算获取特征表示。这种网络结构通过引入三维卷积操作，能够有效地处理如点云或视频数据等三维数据，提取其深层次的空间和时间特征。对于点云压缩领域，三维卷积神经网络能够通过对点云转化而成的规则网格体积模型进行卷积操作，能够达到高效的特征保留。

**3D Data**

**……**

**Kernel**

**……**

**……**

**Convolutional Layer**

**……**

**……**

**……**

图 2.4 三维卷积示意图

三维卷积神经网络通常由多个卷积层和池化层交替组成。此外，网络还可能包含全连接层和激活函数，用于特征融合、分类或回归任务。激活函数的引入增加了模型的非线性，使其能够更好地适应复杂的数据分布。

（1）卷积层

在三维卷积神经网络中，卷积核在三个维度（通常是深度、高度和宽度）上进行滑动，对输入的三维数据块进行卷积操作。在每个位置，卷积核与其覆盖的输入数据块的部分进行元素乘法并求和，生成一个输出值。这个过程在整个输入数据上重复，生成一个新的三维输出数据块。这种操作使得3D卷积神经网络能够捕捉到三维数据中的局部特征比如空间结构。

（2）池化层

三维卷积神经网络的池化层与二维卷积神经网络中的池化层在功能上是相似的，但处理的数据维度更高。池化层在三维卷积神经网络中起到了降低特征图的空间和时间维度、减少参数数量和计算复杂度的作用，同时保留重要的特征信息。具体来说，三维池化层通过在输入特征图上进行滑动窗口操作，对每个窗口内的数据进行下采样。与二维池化层不同，三维池化层的滑动窗口在三个维度上进行移动，通常是空间上的两个维度加上时间维度。这使得三维池化层能够同时考虑空间和时间上的信息，提取出更具代表性的特征。

在池化过程中，常用的操作包括最大池化和平均池化。最大池化选择滑动窗口内的最大值作为输出，而平均池化则计算窗口内的平均值作为输出。这些操作有助于减小特征图的尺寸，降低模型的计算复杂度，并且帮助网络具备一定程度的平移不变性。通过三维池化层的操作，模型可以更加关注于输入数据中的显著特征，忽略一些细节信息，从而提高模型的鲁棒性和泛化能力。

（3）激活层

激活层主要对卷积层的输出结果进行非线性映射以增强网络的表达能力。通过激活函数，激活层能够引入非线性因素，使得网络能够学习并逼近复杂的函数关系，这有助于网络更好地捕捉输入数据中的特征，并提升模型的性能。因此，激活层在三维卷积神经网络中扮演着至关重要的角色。

常见的激活函数有ReLU、Sigmoid、Tanh、GELU等，其中ReLU函数的公式如式2.1所示，该函数提供了一个简单的非线性变换，能够将负数值设置为0从而增强网络的非线性特征。ReLU的函数图像如图2.5所示。

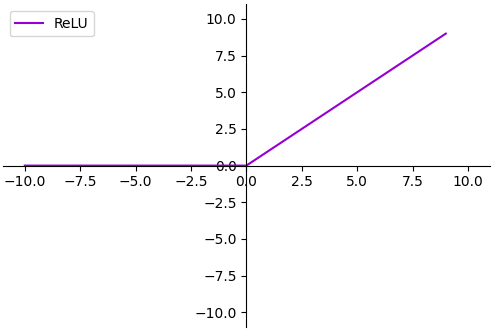


图2.5 ReLU函数图像

（4）全连接层

全连接层主要起到分类或回归的作用。它接收之前层（如卷积层、池化层等）提取的特征，并将这些特征映射到样本标记空间，从而输出最终的预测结果。全连接层中的每个神经元都与前一层的所有神经元相连，通过权重和偏置参数进行线性变换，并经过激活函数进行非线性处理。通过训练过程，全连接层学习并优化这些参数，以最小化预测误差，从而实现对输入数据的准确分类或回归。

* 1. 深度学习常用模块

随着深度学习技术的不断发展，新的模块和算法也在不断涌现，为各种应用场景提供更强大的支持。本节介绍了深度学习在各种领域内常用的几个模块。

2.4.1 Inception模型

Inception模型[47]是一种基于卷积神经网络（CNN）的架构，于2014年的图像识别GoogLeNet网络[48]中首次提出，其设计理念是在不改变原始模型的准确度的同时降低模型的计算量，在图像领域实现了多尺度的特征提取，有效提高了模型对图像特征的学习能力。

其基本框架如图2.6所示。Inception模型通过1\*1的卷积来独立学习通道间的相关性并映射到独立的空间中，接着通过5\*5、3\*3卷积以及最大池化层学习其中所有的相关性，最后将多个卷积核运算结果连接得到最终输出。以此方法能够将通道与空间的相关性进行解耦，充分利用了多尺度的信息。

**Previous Layer**

**1\*1 Conv**

**1\*1 Conv**

**1\*1 Conv**

**MaxPooling**

**5\*5 Conv**

**3\*3 Conv**

**Concatenation**

图2.6 Inception模型

2.4.2 Residual模型

ResNet[49]是Microsoft研究院提出的模型，该模型设计了一种Residual残差块的理念，解决了在训练过程中出现的网络退化现象，大大提高了卷积神经网络的训练深度。Residual模型的示意图如图2.7所示，最终输出结果即为初始值与对应各层卷积结果逐个相加的值，这种连接方式使得F(x)在训练的深度增加时能够无限接近于0，从而让深度网络的精度在最优浅层网络的基础上不会下降，避免了梯度在传播途中的衰减。

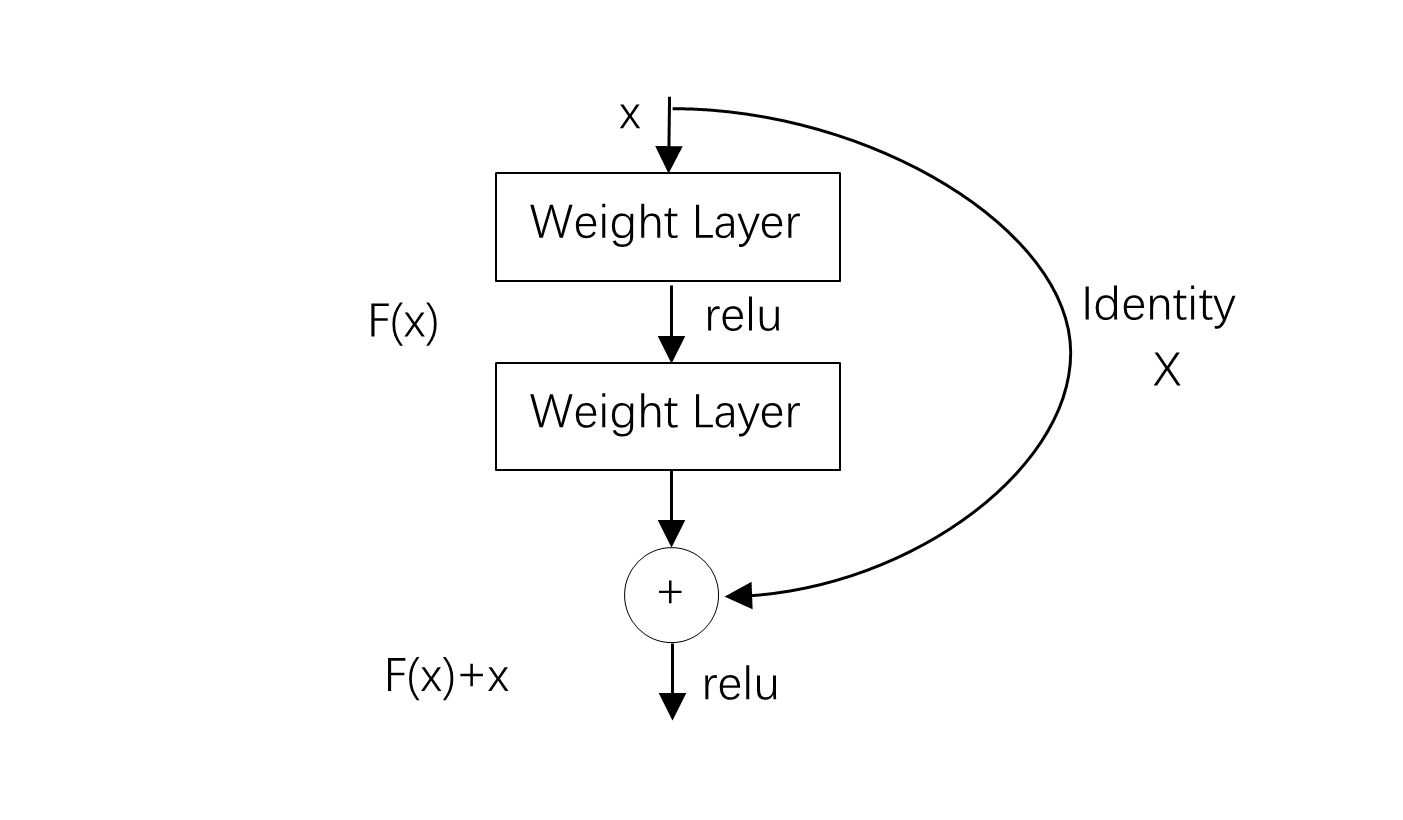


图 2.7 Residual模型

2.4.3 熵模型

无论在传统算法中或是深度学习压缩框架中，熵编码都扮演着及其重要的角色。在点云压缩流程中，通常需要采用熵编码的形式对前置压缩步骤中获得的几何信息与属性信息进行压缩以完成整体流程。熵编码的基本原理是按照熵原理进行编码，确保在编码过程中不丢失任何信息。熵在此处指的是信源的平均信息量，即不确定性的度量。这种编码方式在视频编码、图像编码、点云编码等领域中都有广泛的应用。在视频编码中，熵编码的主要任务是将一系列用来表示视频序列的元素符号转变为一个用于传输或存储的压缩码流。这些元素符号可能包括量化的变换系数、运动矢量信息、预测模式信息等。熵编码能够有效去除这些视频元素符号的统计冗余，是提高视频编码压缩效率的重要工具之一。

常见的熵编码方法包括香农（Shannon）编码[50]、哈夫曼（Huffman）编码[51]、算术编码（arithmetic coding）[52]等。这些编码方法各有特点，但都是基于概率统计原理，通过为不同符号分配不同长度的码字，使得编码后的码流长度最短，从而实现信息的有效压缩。

* 1. 本章小结

本章首先介绍了点云压缩的分类与评价指标，为本文开展实验提供了依据。接着介绍了几种点云压缩基准算法的思想与流程，为后文验证提出方法的有效性提供了参考对象。然后介绍了卷积神经网络的相关概念，为后文基于深度学习的点云压缩框架提供了理论基础。最后介绍了几种常用的深度学习模块，为后文点云压缩模型中模块的设计与改进提供了借鉴思路。

# 基于缺陷点分离的工业产品点云压缩方法研究

* 1. 引言

高密度的点云数据极大地增加了点云数据处理的计算量，这给各种点云操作任务例如点云分割、配准、补全、识别均带来了很大的挑战，于是通过点云压缩的方式减少数据处理的时间和计算资源的消耗，提高实时性和响应速度成了必要。何宽等人[53]设计了一种逐点前进点云几何有损压缩算法，该算法参考了使用于二维空间的道格拉斯普克算法[54]并将该思想应用于三维空间，最终对兔子点云实现了较高的压缩率与较低的运行时间。然而，他们的方法过度考虑了点云的整体特征，该方法采用了等同的策略去处理各个部位的点，因而忽略了许多特征点密集部位的特征提取，最终导致了特征点的大量丢失。在吸取了该研究点的经验后，我们在不破坏全局特征的情况下将视角着重于局部细节的特征提取来提高点云压缩后的重建质量。

裴书玉等人[55]针对激光雷达扫描的古文物模型点云存在大量冗余点的问题，以传统算法的形式设计了一种基于自适应分层的点云压缩方式。该方法根据古文物模型的表面特征合理选取了一种基于倒角距离的分层方式，接着使用弦高差值来归类特征点，最后根据一定的阈值将各层的冗余点进行筛除完成压缩。该方法将注意力集中于古文物特有的细节特征，设计了一种适用于古文物的点云压缩方式，这为本章设计适用工业产品的点云压缩方式提供了一定的借鉴意义。

在工业领域内，将工业产品模型转化为点云的形式，并对点云进行压缩后再进行处理能够极大地降低计算量，同时压缩后的点云数据通过三维重建技术便能够获取较为接近原始产品的模型文件。然而传统的点云压缩方法通常只能处理表面平滑的点云数据，满足不了工业领域表面复杂点云文件的压缩要求。本章节提出了一种适用于具有缺陷点的、表面复杂的工业产品点云压缩方法，该方法通过不同的压缩方式分别压缩缺陷点与非缺陷点部分。我们对于缺陷点设计了一种曲率分级的压缩方法，对于非缺陷部分设计了一种跨点压缩的方法，分别完成了两部分的点云压缩，通过实验对比的方式给出了本文压缩方法在时间、压缩率、表面积变化率与重建误差的结果对比，凸显了本文方法对于工业点云压缩的优势。

* 1. 方案介绍
     1. 基本思路

本章设计的面向复杂表面的工业产品点云压缩方法的基本流程具体有以下几个步骤：

1）对点云数据进行缺陷点提取得到缺陷点点云数据和去除缺陷点的点云数据；

2）对缺陷点点云数据使用基于高斯曲率[56]分级的压缩算法进行压缩获得压缩后的缺陷点点云数据；

3）对去除缺陷点的点云数据使用跨点压缩算法获得压缩后的去除缺陷点的点云数据；

4）将步骤3与步骤4中的点云结合获得完整的工业品压缩点云；

5）对工业品压缩点云进行泊松重建得到Mesh模型；

6）进行压缩率和重建误差的评价。

具体操作流程如图3.1所示：

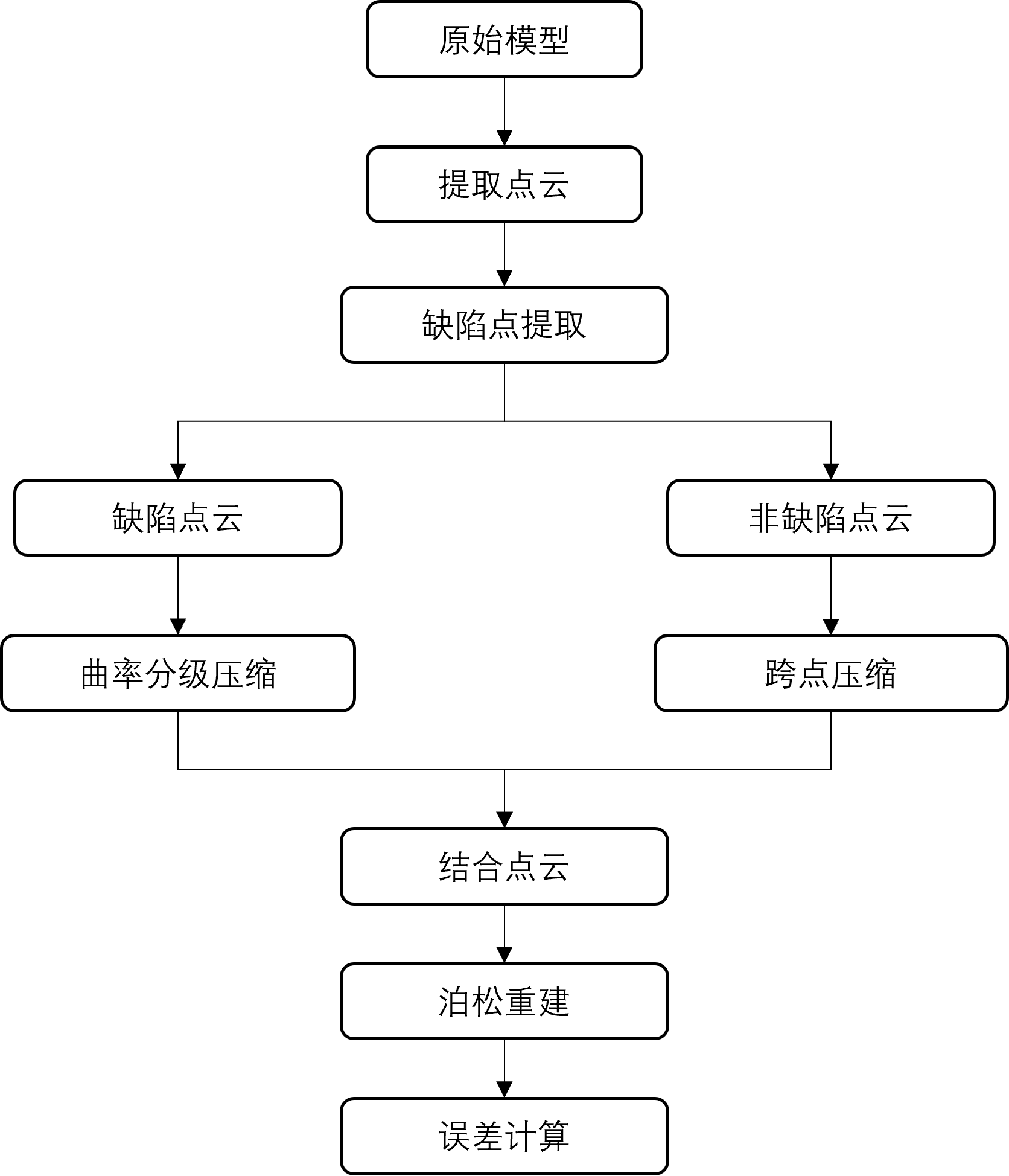


图3.1 总体压缩流程

* + 1. 曲率分级法压缩缺陷点云

缺陷点是本文研究的点云数据中最重要的特征点部分，是原始模型上的凸起、凹陷或者尖锐部分，因此必须将缺陷点的压缩与整体点云数据的压缩进行区分以最大限度地保留特征点。通常对于缺陷点可以不进行压缩，但由于本文采用的缺陷点提取方式是按照索引近似提取，会包含缺陷部位附近的部分平坦区域，所以使用一种有效的压缩方式很有必要。

本章节设计了一种适用于缺陷点的压缩方法，首先通过最小二乘法拟合二次曲面法计算缺陷点云中每个点的高斯曲率，然后将该曲率量化后根据一定公式进行三种分级，第一种分级为曲率较小的平坦部分的点，该种类的点几乎不保留，通过具体的压缩策略以极小概率保留；第二级为普通特征点，通过分级函数的阈值因子控制大小以确定保留程度；第三级为曲率值较大的点，由于曲率变化较大的部位最容易反映特征的变化，所以该类特征点完全保留。最后对分级后的点采用分级压缩策略压缩得到压缩后的缺陷点云。使用该方法压缩缺陷点云能很好地保留缺陷点并且能够略微筛去在缺陷点提取时额外划分在缺陷点的周边轮廓点云数据。具体的压缩步骤如下：

（1）基于最小二乘法拟合二次曲面对缺陷点点云数据进行高斯曲率计算。

散乱点云中取一个数据点，然后以为中心在点云中均匀地取出个点，这个点尽量覆盖整个点云。通过这个点，利用最小二乘法拟合二次曲面公式，解得系数后根据空间曲面的性质计算数据点的高斯曲率[18]。

对于目标点云的任一个数据点的邻域，根据最小二乘原理需要使下式3.1取最小值：

式中，，，是邻域内的点。将上式分别对系数求导，使其为0，得出下式3.2：

由此，解出二次曲面系数。将曲线方程写成曲面参数的形式如式3.3所示：

若曲面上存在一条曲线*r*，则*r*的表达式为。通过对该等式求导，可得式3.4：

如果*s*表示曲线*r*的弧长，则弧长以通过弧长微分公式获得，微分规则如式3.5：

如式3.6，由曲面的第一基本公式可得：

假如*P*是曲线上的一点，则曲线的单位法向量可表示为式3.7：

由曲面的第二基本公式，可得式3.8：

通过引入方程如式3.9：

则曲线法曲率可表示为式3.10：

通过操控变量λ，我们可以得到两个主曲率和。于是，我们可以根据曲率特性求得高斯曲率，如式3.11所示：

（2）对获取的每个点的高斯曲率进行分级获得该点的曲率等级，如式3.12。

该公式将曲率的等级分为3级，其中等级0表示曲率为0的平坦点，等级为1表示曲率较小的点，等级2表示曲率较大的点。*T*为曲率控制因子，改变*T*的大小可以控制等级1和等级2点的比例，*T*值越大等级为2的点就越多，即曲率较大的点越多。*H*代表了第*i*个点的高斯曲率，由于高斯曲率可能存在负值，这边统一将其设置为正数。*Ceiling*函数为向上取整函数。采用*ln*函数取平方值的目的是使得曲率较大的点能够有更高的概率保留为等级2。

（3）遍历曲率等级，将满足一定条件的点滤除，压缩思想类似于跨点前进法。

取出并判断连续的3个点的曲率等级，如果连续3个点的曲率等级均为1则保留两端的点，删去中间的点，以此方式能够有效地保留曲率略大的点数据并且略微筛去一部分点而不会破坏整体的特征；如果连续3个点的曲率等级均为0，则删除两端的点，保留中间的点，这样能够大量删减平坦曲率点，仅仅保留近乎三分之一的平坦点；其余情况下则将3个点全部保留，这样大曲率点能完全保留的同时也能兼顾到大曲率往小曲率点过度的平坦点，达到了良好的压缩性能。由此不断进行3点循环直到遍历完成。采用这种删减方法的目的是保证微小局部的凹凸性，能够在保留部分小曲率点的同时有效保留大曲率点。具体操作流程如下图3.2所示：

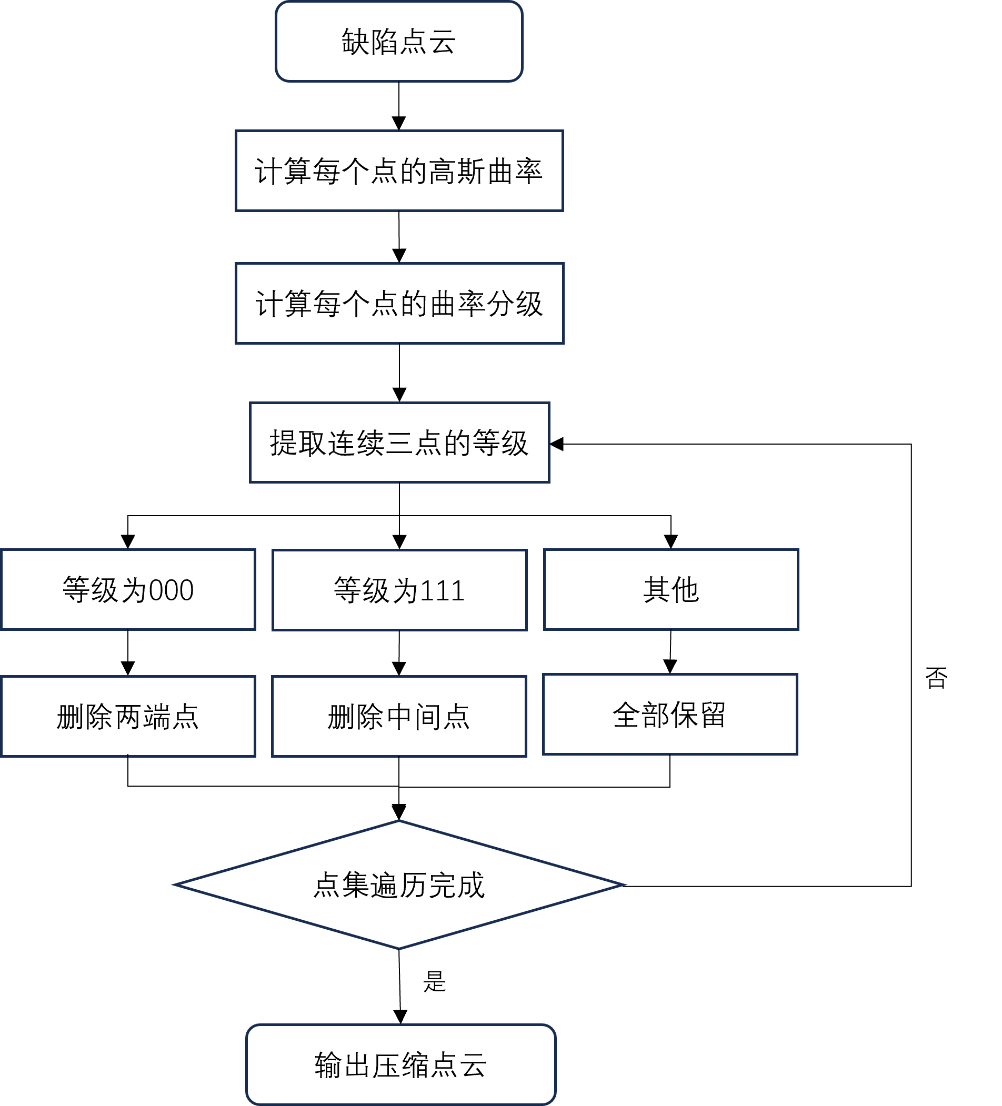


图 3.2 曲率分级法压缩流程

* + 1. 跨点压缩法压缩非缺陷点云

相较于缺陷点点云数据必须大量保留原始点集，对于非缺陷点点云可以采用相对宽松的压缩方法，在保证特征点大量保留的情况下能较好地删减非特征点区域。另外，压缩后的重建工作也是评价压缩质量的关键一环，所以对于压缩方法的设计也要同时考虑到为后续重建工作提供便利。

本文对非缺陷点点云压缩的方式参考了道格拉斯普克算法对于二维点的压缩思想，设计了基于三维点云数据的压缩方法，采用了一种跨点压缩的方法实现非缺陷点点云的压缩。

对于矢量数据的压缩，直到现今已经有很多方法，例如垂距法、光栏法、道格拉斯普克法等。道格拉斯普克算法是一种基于曲线多变形逼近的垂距法的改进方法。该方法是基于二维曲线矢量数据的一种数据压缩方式，其主要思想为：遍历曲线两端点构成的一条直线之间的点，依次判断各个点到直线的距离，选取距离最大的点判断满足一定的阈值，然后将其他点删减仅留下距离最大的点，将保留下来的点作为分界点依次对左右两侧重复执行最大距离判断。该方法实现了一条直线内变化较小的点的压缩，实现了特征点，即距离最大点的保留。对于三维点云数据也可以参考这一设计思想，将二维点的压缩方式在三维场景中实现。

本文参考了以上思想，在三维点集中，可以将完整的点云分为若干个小点集，对于每个小点集，均采取一种合适的特征点保留策略，将整体的压缩转换为点集与点集之间的特征提取从而进行压缩，这样能够实现均匀压缩，不会破坏局部的特征点。于是，我们设计了一种参考道格拉斯普克算法思想的算法，在三维点云中实现了一种跨点压缩的方法。

具体地，通过点云中连续几个点进行特征距离计算，判断当前点与端点所在直线的隔离性来决定是否将该点保留。该方法可参照图3.3的设计思想，先由小范围局部判断点与直线的隔离性，如图中点C与直线AB的距离如果小于一定的阈值，则可认为该点与当前直线冗余度较高，可以列入待删减队列，随后继续向后搜寻两点DE分别构成直线AD与AE，通过继续判断C点与另外两直线构成的点集的隔离性来判断C点是否相较于其他集群具有一定的隔离性，若C与另外两条直线相距较远，则可以认为C为其中一条直线构成的点集的特征点，所以可以将其保留。总的来说，该方法赋予了每个点额外的两次判断机会，加大了特征点保留的概率同时也能够有效删减冗余的平坦点。该方法的具体设计步骤如下：

（1）从点云集合中提取前三个点，记为A（X1，Y1，Z1），B（X2，Y2，Z2），C（X3，Y3，Z3），连接AB两点，计算点C到直线AB的距离*d1*。

（2）判断*d1*是否小于阈值*H0*，如果不小于，则将该点C保留在特征点点云集合中，继续向后搜索，即从当前的点C开始重新执行步骤（1）。

（3）如果*d1*小于阈值*H0*，则继续向后引入点D（X4，Y4，Z4）与点E（X5，Y5，Z5），连接AD与AE，分别计算点B到直线AD与AE的距离，记为*d2*，*d3*。如果*d2*和*d3*均大于等于2倍的*H0*，则给予点C“机会”，将原本应被舍弃的点C加入特征点点云中。否则，点集索引前进，继续步骤（1）。

（4）遍历直到点集全部检索完。

（5）将遍历完成保留下来的点汇总到一个新的点云中，得到压缩后的非缺陷点点云。

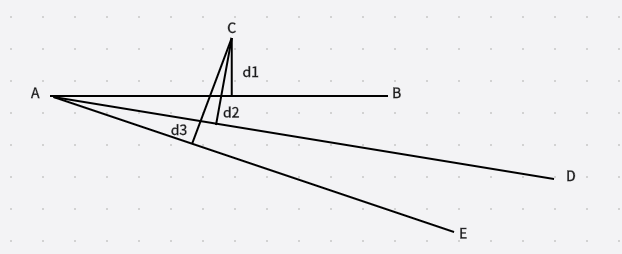


图3.3 跨点压缩思想

对于距离判断阈值*H0*的取值应设置为一个极小数，由于本方法采用了小范围的隔离性验证，所以可能会陷入局部适应的情况，即连续三点构成的小点集中，如果该点C始终小于一个较大的*H0*阈值，则每次判断的点C均会被认定为非特征点删除，这样会导致删减点过多而损失局部的特征，为了保证当前点C能以一定的条件被列入待保留点或者删除，该点必须满足距离三条直线的距离均大于阈值才能将其认定为具有隔离性的特征点，因此将*H0*设置为一个极小的数能够有效满足点集内距离过于相近近似重叠的冗余点的删减和具有隔离性的特征点的保留。通过设置不同的*H0*阈值，依据后文实验可以发现当*H0*从0.05至0.15取值时候，压缩率降低了将近25%，使得过量的点被删除以及点云部分特征丢失，最终导致重建提取点云后的误差偏高。因此选取合理的阈值能够对提高压缩的质量。

* 1. 实验

对于工业点云压缩效果的评价可以从多种方面进行判断，例如压缩率、压缩时间、重建质量等。本文实现的算法综合了以上几种评价标准进行评判，结果表明我们实现了一种优秀的点云压缩策略。

通过对比实验，我们给出了各个压缩方法的压缩效果以及重建后的恢复质量，通过与经典算法进行比较来凸显本文在复杂表面工业品点云压缩方向的优势。本节给出了缺陷点的压缩情况、非缺陷点压缩质量与全局重建的效果。

3.3.1缺陷点云压缩效果

通过Cloud Compare软件[57]观察一个表面复杂的工业品点云文件，其多角度视图如3.4所示。缺陷点部位如图3.5所示，本文实验的点云数据为带有凸起和凹陷的工业零件点云，点数为500062。另外，通过Cloud Compare软件提取缺陷点部位的索引，将点云分为缺陷点部分与非缺陷点部分，缺陷点部分提取点云点数为31301。



图 3.4 点云多角度视图

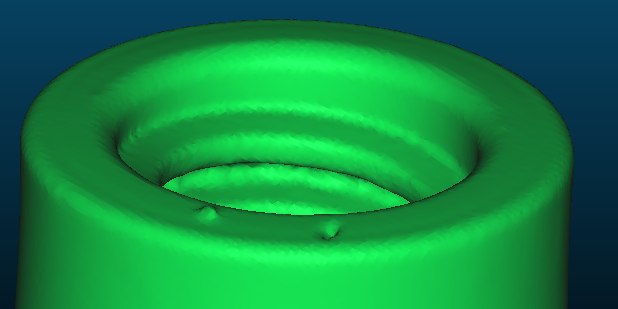


图3.5 缺陷点部位视图

通过3.2节的缺陷点曲率分级压缩方式对缺陷点点云数据进行压缩，我们实验了不同的曲率控制阈值*T*来改变曲率等级的分布，具体的等级分布于压缩情况如下表3.1所示。

表 3.1 不同*T*值时各等级点数（个）与压缩情况

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 阈值*T* | 等级1 | 等级2 | 等级3 | 压缩后 | 压缩率 |
| 20 | 49 | 18733 | 12519 | 20402 | 65.18% |
| 40 | 49 | 16302 | 14950 | 20966 | 66.98% |
| 60 | 49 | 15076 | 16176 | 21203 | 67.74% |
| 80 | 49 | 14337 | 16915 | 21344 | 68.19% |

由表数据可以看出在压缩率差距在3%左右的情况下多保留了近1000个等级为3的点数，说明本文的缺陷点曲率分级压缩法能够较好地保留较大曲率的特征点。

3.3.2非缺陷点云压缩效果

跨点压缩法压缩点云的结果受到距离阈值*H0*的影响，我们实验了不同距离阈值下的跨点压缩法压缩情况，使用的点云数据为前文提取的去除缺陷点点云，点数为468761。在一般情况下为了保证良好的重建效果，*H0*应取一个极小值，并且根据测试的点云会有一定的浮动，需要根据特定点云进行测试，便于通过多次筛选保留下可能成为特征的点。具体实验数据如下表3.2所示。

表 3.2 不同*H0*值的压缩情况

|  |  |  |
| --- | --- | --- |
| 距离阈值*H0* | 压缩后点数/个 | 压缩率/% |
| 0.05 | 349816 | 74.62 |
| 0.1 | 213711 | 45.60 |
| 0.15 | 159953 | 34.12 |

由表中数据可以得出在距离阈值越大，压缩效果越好。为了提高重建的水平，将非缺陷点的压缩率控制在合理的范围内。下图3.6为阈值*H0*为0.1时的点云可视化细节图。

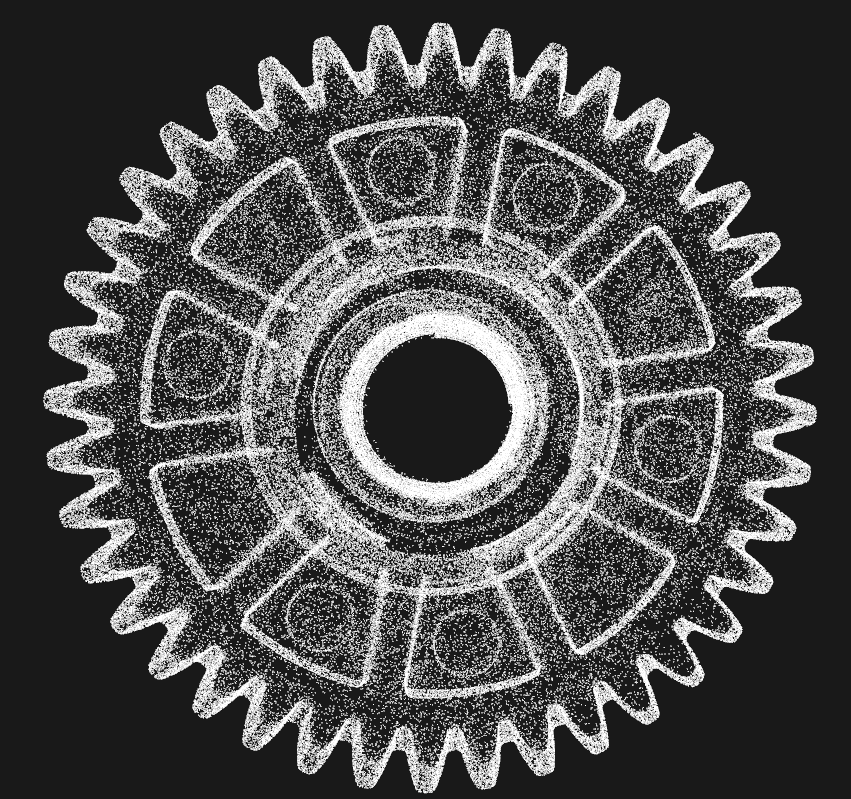


图 3.6 阈值*H0*=0.1时的点云部分细节

3.3.3对比实验

将缺陷点点云与压缩后的非缺陷点点云结合得到完整压缩后点云，通过本文压缩方法与传统PCL点云压缩算法、均匀压缩法与逐点前进法进行对比，以证明本文压缩流程在复杂表面的工业产品点云压缩方面的优越性。

针对静态点云，压缩率、压缩时间、重建误差都是重要的数据评价标准，另外重建水平视觉指标也是一项重要的主观评价指标。本文通过运行不同的压缩算法来对本文的表面复杂的工业产品点云进行压缩，并使用泊松重建法来进行点云重建比较视觉效果。本文压缩方法设置曲率阈值*T*=80，距离阈值*H0*=0.1。

3.3.4压缩率评价

本文使用的点云数据点数为500062，通过经典PCL库压缩算法、比例压缩算法、逐点前进压缩法和本文算法进行比较。由下表3.3数据可见，经典PCL库算法在面对数量及其庞大的点云数据取得了较为不理想的压缩率，几乎没有压缩点集；比例压缩法在设置与本文算法相近的压缩率时在耗时上较为落后，由于比例压缩法需要指定一定的压缩比例，随机性过大所以效率低下；逐点前进压缩法应用于整体压缩时取得了和本文算法相近的压缩率，但是在时间上有所落后。综上可看出本文算法在压缩率和压缩时间均有一定的优势。

表 3.3 不同算法压缩情况对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 压缩方法 | 点云数目 | 压缩点数 | 压缩率/% | 压缩时间/s |
| PCL算法 | 500062 | 499108 | 99.81 | 2.531 |
| 比例压缩 | 500062 | 235029 | 47.00 | 3.093 |
| 逐点前进 | 500062 | 229352 | 45.86 | 2.418 |
| 本文算法 | 500062 | 235055 | 47.01 | 2.382 |

3.3.5重建水平视觉评价

将比例压缩算法、逐点前进压缩法和本文算法分别通过泊松重建获取Mesh模型，通过Cloud Compare软件进行可视化。如下图3.7所示，本文算法在重建的全局视觉效果上能够清晰完整地还原原始点云模型，不存在明显的空洞与撕裂。如图3.8所示，相比与其他两种方法，本文方法能够对缺陷部位进行较好的重建，能够获得棱角更加分明、更加接近原始模型的细节特征。以上视觉效果表明本文的改进具有实际作用。



图3.7 本文方法复原总体视觉效果

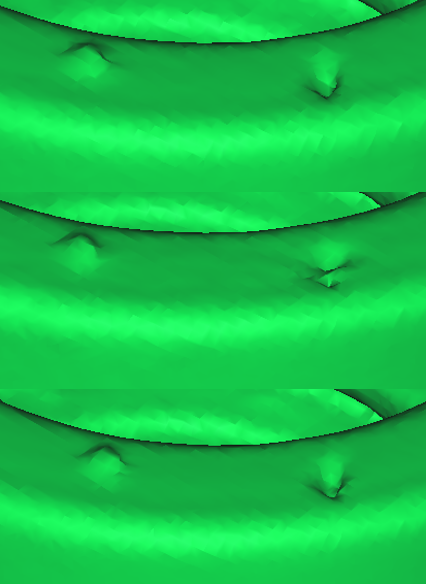


图3.8 缺陷部分视觉效果（上为比例压缩、中为逐点前进算法、下为本文算法）

3.3.6表面积评价

点云表面积可以通过三角网格化点云数据，并通过具体的公式进行计算。具体地，通过不同的压缩方式压缩点云，将压缩后点云重建为三角化Mesh模型并计算点云Mesh模型的表面积。为了比较原始点云与不同压缩方式得出的压缩点云的表面积差异，判断压缩对原始点云物体表面的修改程度来评估压缩质量，我们参考了李金涛等人[58]的论文中的公式来计算表面积，如式3.13：

其中，为三角形顶点坐标，、、分别为该三角形的各条边长度，而*L*则为三边长度之和的二分之一。

计算原始点云模型、比例压缩法点云模型、逐点前进法点云模型、本文算法点云模型的表面积，可以得出如下表3.4数据：

表 3.4 不同算法重建表面积对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 压缩算法 | 压缩率 | 原始表面积 | 压缩表面积 | 表面积变化 |
| 比例压缩 | 47.00 | 5141.56 | 5125.26 | 0.00317% |
| 逐点前进 | 45.86 | 5141.56 | 5122.39 | 0.00373% |
| 本文算法 | 47.01 | 5141.56 | 5137.31 | 0.00083% |

由表中数据可得出，使用本文算法压缩获取的三角模型表面积与其他两种算法进行对比，本文算法表面积变化率有明显优势，表明通过本文的压缩方式得到的最终点云物体表面特征改变较小。

3.3.7重建水平误差评价

本文通过计算重建后Mesh模型提取的点云集与原始点云数据的均方误差值来判断压缩后点云与原始点云的误差。均方误差是一种判断被评价者与原始数据之间的差异程度的度量标准。具体采用式3.14的公式对压缩后点云点集与原始点云计算均方误差RMSE。其中*n*为点数目，*di*代表当前点到原始点云中最邻近点所在平面的距离。

由下表3.5数据可见，本文算法在最大最小误差距离上均小于比例压缩法和逐点前进算法，表明本文算法能够有效地保留原始特征点，避免相对于原始点云过于离散。同时RMSE数值均小于另外两种算法，可见本文算法在重建保留特征数据方面优势明显。

表 3.5 不同算法复原误差比较

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 压缩算法 | 提取点云数 | 最大误差距离 | 最小误差距离 | RMSE值 |
| 比例压缩 | 367771 | 0.747588 | 2.79397e-08 | 0.0375422 |
| 逐点前进 | 368321 | 0.760116 | 1.21072e-08 | 0.0360975 |
| 本文算法 | 368081 | 0.76834 | 1.86265e-09 | 0.0356602 |

综合以上各种对比实验，我们可以得出以下结论：本文算法在取得了较好的压缩率和压缩时间的同时最大限度地保留了特征点，使得重建后的工业点云误差更小，能够更加精确地还原原始点云数据。这也符合了我们设计一种适用于复杂表面工业品的点云压缩方法的初衷。

* 1. 本章小结

本章节针对复杂表面的工业产品点云缺少合适的压缩方式的问题，考虑到使用传统压缩方法无法特殊处理缺陷点，存在时间效率较低、压缩率与重建精确度较差的情况，设计了一套采用缺陷点、非缺陷点分离压缩最终组合重建的流程。首先将工业点云数据缺陷点部分与非缺陷点部位分割；随后通过设计曲率分级压缩方法实现缺陷点点云的压缩，尽量保留缺陷点削减平坦点；然后通过跨点前进法实现非缺陷点的压缩，目的在于压缩点云的同时更好地保留特征点；接着将压缩后的缺陷点点云与非缺陷点点云结合得到完整的压缩后点云；最后将点云进行泊松重建计算点的均方误差评价重建质量。本文方法与传统PCL库压缩算法、比例压缩算法与逐点前进法进行了压缩率、压缩时间、表面积变化与重建误差等比较，本文得到的结果相比这几种传统算法有明显提升，凸显了本文设计流程在应用于工业领域的优势。

另外，对于缺陷点的提取部分仅仅是依靠索引提取，并没有通过一种特殊的算法直接提取，效率较低局限性较大，后续会在该点上进行继续的研究。

# 基于优化残差块与逐通道卷积的点云压缩框架研究

* 1. 引言

近年来，点云在交通、医疗、工业等领域得到广泛应用，研究人员通过点云压缩的形式极大提高了数据的传输效率。早期的点云压缩方法研究通常采用传统算法进行点云特征的提取，从而进行点云编码压缩或数据压缩，但是这些传统方法往往难以高效地从庞大且复杂的点云数据中提取特征数据，因而会产生丢失大量关键数据以及压缩效率局限于目标点云压缩参数设置的问题。目前在工业领域内并未应用过一种能够有效提取点云特征的深度点云压缩方法。得益于深度学习神经网络框架的不断发展，大量研究者对点云的识别、补全、配准等任务纷纷开始采用深度学习的方式，最终他们提出的框架均取得了有效的成果。于是，借助深度学习方法设计一种点云压缩方法便有了大量可参考的经验。

Nguyen[59]等人使用基于稀疏张量的卷积神经网络有效地利用点云中的特征和点依赖关系，从而为算术编码器构建了一个准确的自回归上下文模型，实现了点云几何与属性的无损压缩。

Wang等人[60]提出了一种名为PCGC的端到端多尺度点云几何有损压缩模型，该模型使用了稀疏卷积多尺度特征提取的方式，通过对体素化的点云输入特征进行多次下采样，每次下采样后成为一个尺度并对其进行特征提取，接着进行下一尺度的下采样与特征提取，最终通过提取到的多尺度特征作为输出特征表示进行编码压缩。该模型在能够达到较好的特征提取的同时能够根据需求获取不同特征尺度的重建点云，在率失真指标方面能够在实验的ShapeNet数据集上明显优与MPEG工作组的GPCC模型与VPCC模型。

Liang等人[61]提出了一种基于Transformer的深度点云几何有损压缩框架，该模型与Wang的模型同样采用了多尺度下采样特征提取的方式，不同之处在于其在编解码器端使用了Transformer进行特征提取与特征重建。该模型相较于MPEG基准模型在Semantic KITTI实地场景点云数据集上得到了明显的峰值信噪比优化。总的来说，该模型能够达到更好的特征提取，但是由于多次使用Transformer模块不可避免的产生了模型复杂度高，计算资源消耗较大的问题。

为了实现更加准确的点云特征的提取，达到更高质量的点云压缩重建指标，同时实现点云的高效传输，本章节以PCGC模型为基准模型，通过深度学习的方法结合多种优秀模块设计与改进了一种点云压缩网络。本章节的贡献在于：

（1）设计并改进了基于优化的下采样模块、深度可分离卷积模块与残差特征提取块进行多尺度点云特征提取的深度点云压缩框架，在公开数据集ShapNet上达到了与点云压缩标准MPEG下的G-PCC模型与V-PCC模型对比的40%至80%的码率节省，相比与PCGC模型节省了约等于10%的码率，表明设计的深度压缩模型能够有效地实现点云压缩任务。

（2）收集工业点云网格模型构建了一套工业点云数据集，并使用该数据集在本章设计的网络模型中进行了实验，通过与传统算法的对比验证了本章节网络模型对于工业领域点云压缩的有效性。

* 1. 方案介绍
     1. 点云压缩框架

如图4.1为本章节提出的基于深度学习的点云几何压缩方法，该框架由编码器、熵编码模型与解码器构成。点云数据首先进行体素化占用的形式进行表示，即通过1表示体素被占用，0表示未占用，将点云数据表示为稀疏张量的形式，为点云几何坐标信息，为体素占用情况。在编码器端，通过对点云信息进行稀疏卷积不断下采样得到多尺度的点云、、，在此过程中，点云的局部特征信息，即体素的占用情况被表示为属性进行保存。接着将的几何信息通过八叉树无损编码，特征信息经过量化随后通过熵编码模型进行无损编码。相应地，在解码端将几何信息和特征信息进行解码，并且通过多尺度上采样进行重建。

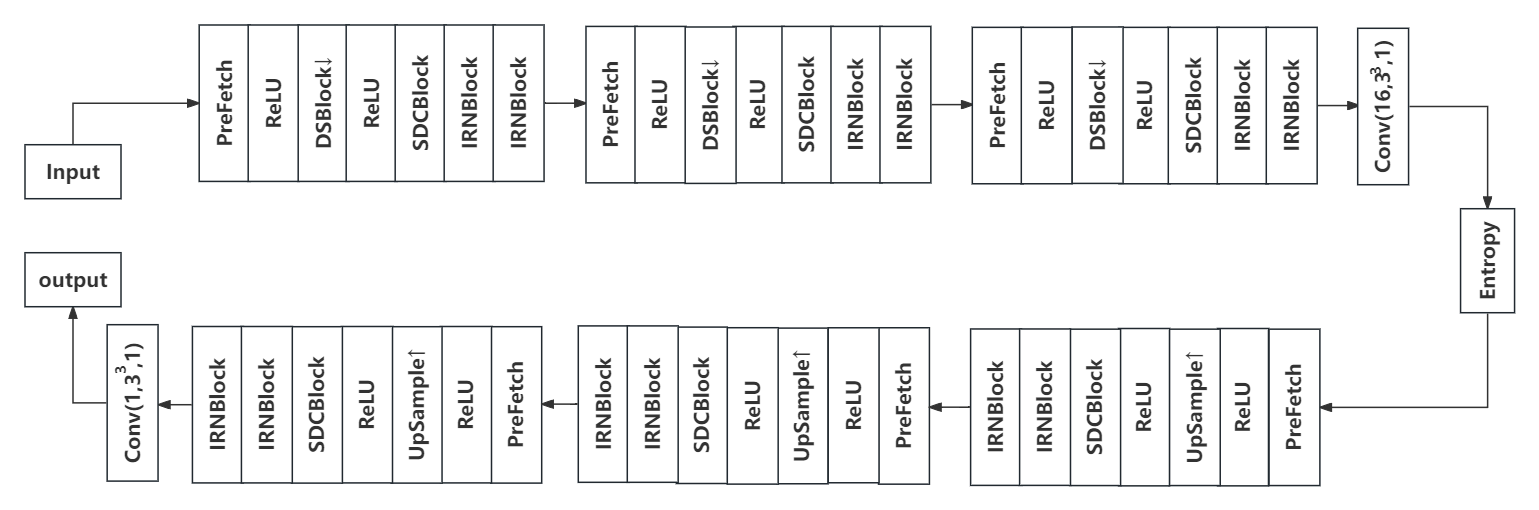


图4.1 模型整体框架

具体地，在编码端首先通过初步特征提取模块PreFetch模块，采用了简单的步长为1，卷积核大小为3\*3\*3的稀疏卷积对输入点云数据进行初步的特征信息提取，随后通过优化的下采样DSBlock模块进行第一次下采样，再通过SDCBlock逐通道残差稀疏卷积块与两个IRN残差特征提取块以相同的通道数输出加强的特征信息。同样地，再紧接着以相同的步长为2的两次DSBlock下采样特征信息增强，最终取得点云尺度。在解码端，采用步长为2，大小为3\*3\*3的卷积核稀疏卷积进行上采样同时通过SDCBlock逐通道残差稀疏卷积与两个IRN块进行特征重建获得点云几何与特征信息与。总的来说，整体模型的编码压缩与解码重建是对称的。

* + 1. 点云的稀疏卷积

点云具有无序性并且在三维空间的分布较为稀疏，基于体素的点云表示方法虽然在数据查找方面具有优势，但会产生内存大量占用的问题。为了克服点云数据的稀疏性、不规则性以及维度高的问题，可以利用稀疏张量与稀疏卷积来运行整个压缩流程[62]，一方面，稀疏张量允许我们仅使用其坐标和相关特征或属性存储已占用的3D空间，另一方面，稀疏卷积是专为稀疏张量而设计的，它与传统卷积非常相似，可以有效地节省内存和计算成本。在Wang与Nguyen的网络模型中都实践了稀疏卷积对于点云表示的优势，于是本章节所设计的网络模型也采用稀疏卷积来进行点云数据的高效表示。

NIVIDA开源的Minkowski Engine库能够对点云数据进行稀疏表示，该库是稀疏张量的自动微分库，作用在高维度空间稀疏数据的操作。该库能够应用于点云的分割、补全、重建、检测等多领域的任务。

通过Minkowski Engine库，点云数据可被表示为坐标矩阵和特征矩阵，具体如式4.1与式4.2。

其中，表示点云的坐标，表示属于batch中的哪一个点云，*N*表示1个batch中所有点的数量，表示第个点的特征，可以表示1维、2维、3维或者其他维度的数据。通过以上的稀疏表示，相比于三维卷积的表示，能够节省空间，并且根据前人实验适用于点云数据的表示。

常规的卷积操作包含卷积运算、归一化、池化等操作，Minkowski Engine库重写了这些常用的函数，使其适应了稀疏卷积的数据表示规则，调用规则与Pytorch一致。

* + 1. 下采样模块与特征提取模块

为了进行多个尺度的特征提取，我们逐步降低点集的基数，在每个初步特征提取块之后构建了一个下采样模块（DS-Block），该下采样模块通过优化的残差块模式进行工作，具体如下图4.2所示。我们使用GELU函数作为激活函数，使用该激活函数能够在梯度为0的时候比ReLU函数更为平滑，因而能在训练中避免ReLU函数产生的死亡神经元问题，能够有效提高模型的鲁棒性。另外，过多的激活层与归一化层会对模型训练产生不利因素，于是我们去除了第一个卷积后的归一化以及激活层以及最后一个卷积的激活层，通过两次大小为1\*1\*1的卷积与3\*3\*3卷积增强细节特征保留，同时采用残差形式对特征进行相加。具体地，首先对输入稀疏张量进行通道数的升高以降低下采样带来的特征丢失，随后通过步长为2的稀疏卷积进行下采样特征保留，接着进行步长为1的稀疏卷积进行特征稳定，最后将特征进行相加完成整个下采样数据精简。

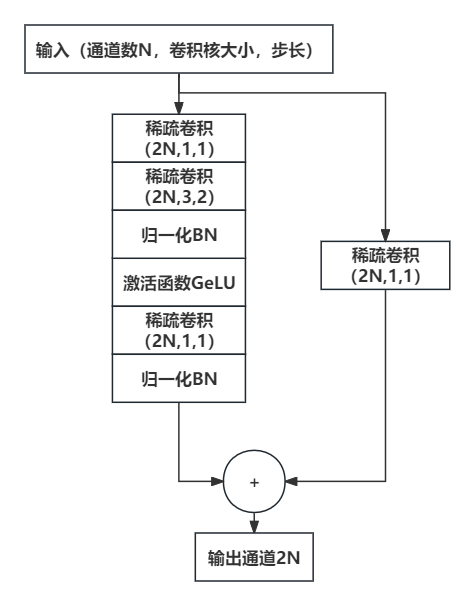


图4.2 DS-Block细节

接着，本章节提出的模型收到了深度可分离卷积能够减少计算量提高特征提取效率的启发，将其与稀疏卷积结合，构造了逐通道稀疏卷积模块（SDC-Block），旨在对特征进行增强，提高模型的计算效率，具体结构如图4.3所示。

另外，模型通过两个连续的Inception Residual模块将大卷积核分解为小卷积核，通过多个计算结果相结合的方式进行特征信息地充分提取，完成一轮次的特征提取。具体地，该模块将前置进行下采样的稀疏张量作为输入，首先将输入的N通道稀疏张量进行步长为1，卷积核大小为1\*1\*1的稀疏卷积进行初步特征提取，随后通过使用步长为1，大小为3\*3\*3的卷积核，group为通道数的逐通道稀疏卷积进行各个通道的稀疏卷积加强特征，最后通过残差形式，将初始输入与卷积后的输出进行相加，从而达到特征结合完成初步特征增强。接着进行第二部分的特征增强，使用两个连续的IRN（Inception Residual Network）模块，具体操作先通过1\*1\*1卷积学习通道相关性，随后将主路与支路通过不同输出通道的稀疏卷积进行合并，最后以残差网络的形式与前置输入相加获得最终的特征张量。完整的IRN特征提取模块如下图4.4所示。

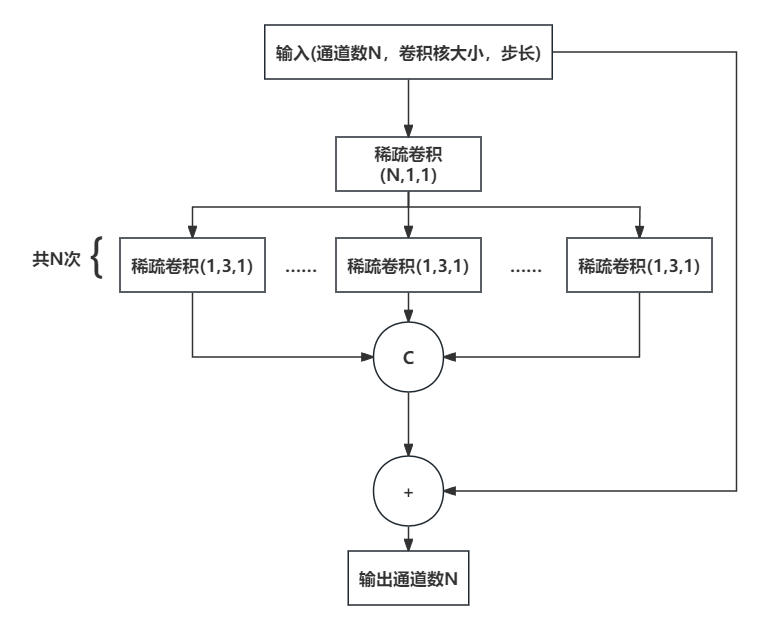


图 4.3 SDC-Block细节

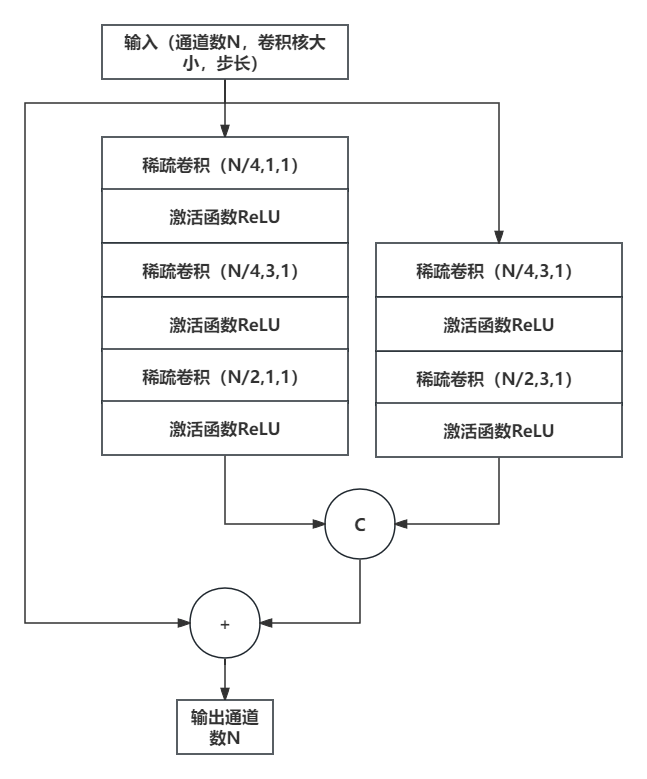


图 4.4 IRN-Block细节

* + 1. 量化与熵编码

具体的量化与熵编码结构如下图4.5所示。

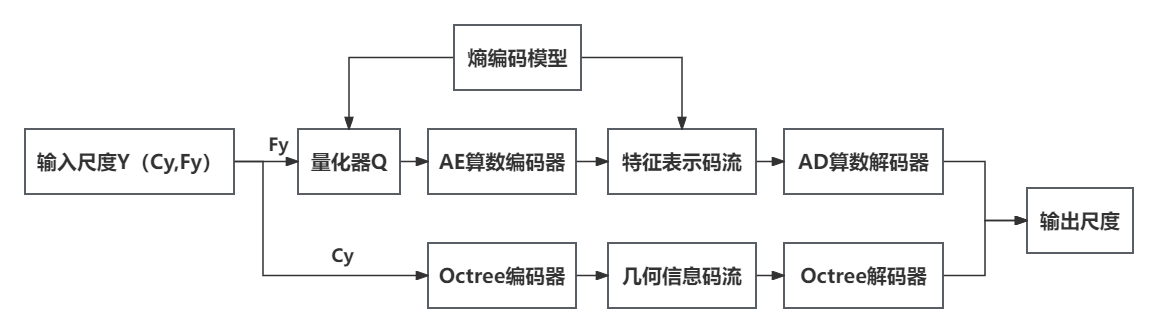


图4.5 量化与熵编码模型细节

对于经过多尺度特提取的点云，为经过三个尺度特征提取获得的特征表示，为几何坐标信息，为体素占用特征表示。前者由八叉树编码模型进行无损压缩获得几何信息码流，后者通过量化器*Q*量化后使用算数编码器模型进行无损压缩获得特征潜在表示码流。对于潜在特征表示量化的过程中，通过添加均匀噪声来模拟实际情况，设置均匀噪声在-0.5至0.5的范围内。除了加入均匀噪声外，还可以通过软量化[63]或跳过量化层来实现梯度反向传播，以保证模型的训练。

* + 1. 率失真损失函数

在点云有损压缩问题中，通常必须在离散化潜在特征的熵和压缩引起的误差之间进行权衡。因此，我们的目标是使占用体素的码率 *R* 和失真 *D* 加权和最小，使率失真性能达到最大化。本文模型损失函数如式4.3所示。

损失函数由两部分组成，其中R即为通过编码输出的特征表示信息码流大小，*D*为*LBCE*表示重建点云与原始点云的二元交叉熵，参数来控制码率的大小以得到不同码率的模型。

上式4.4为*LBCE*损失函数，式中*N*为体素数量，表示原始点云数据体素占用情况，使用1表示占用，0表示未占用，为体素被占用的概率，如果输出结果中的每个体素占用的概率越接近真实值则该损失值就越小，表示重建获得的点云数据更加接近真实值。

由于本章所设计的压缩模型通过多个尺度，每个尺度的损失值构成了失真率*D*，即。其中，*i*为尺度序号，*M*代表了经过特征提取的轮次。

* 1. 实验
     1. 模型训练

本章节设计的模型采用ShapeNet作为数据集，其中包含了约51300个CAD表面模型，训练时随机选取了20000个点云模型，从中抽取10%作为测试集，将其通过随机旋转最终量化坐标至7bit。

在测试时，我们从ShapeNet中选取了10个密集点云进行测试，这些点云数据被MPEG点云压缩标准设置为标准测试序列，更有助于评价模型压缩的效率，测试集具体包括分类下的8iVFB中4个点云（Longdress、Loot、Redandblack、Soldier）、MVUB中4个点云（Andrew、David、Phil、Sarah）、Owill中4个点云（Basketball player、Dancer、Exercise、Model）。通过在以上数据集的测试，我们实验了MPEG标准下的GPCC（Octree）模型、GPCC（Trisoup）模型、VPCC模型与Wang等人的PCGC模型。

所有操作的实验平台使用了AMD EPYC 7642 48-Core Processor的CPU，RTX3090的GPU，80GB的内存。实验环境为PyTorch1.8.1+ Python3.8+ Cuda 11.1。训练轮数设置为50，batchsize为8，初始学习率为0.0008，通过设置每一个epoch的衰减系数为0.5最低至0.00001，采用Adam优化器。另外通过控制为0.25,1,2,4,8获得了不同码率的模型。

* + 1. 对比实验

MPEG发布了官方的度量指标pc\_error，其中规定了多种点云评价指标，其中点到点均方误差D1（p2point）与点到面均方误差D2（p2plane）是较为常用的测量指标。

通过以上两种指标归一化方式来评估失真，将其误差转化为峰值信噪比PSNR进行度量，于是可得两种评价方式分别为D1-PSNR与D2-PSNR，转化公式如式4.5，其中B代表评价点云中的点，A代表原始点云中的点，表示D1或D2均方误差值，表示距离的峰值。

我们通过实验比较绘制了MPEG标准下tmc13模型下的GPCC（Octree）与GPCC（Trisoup）方法、tmc2模型下的VPCC方法与Wang的PCGC模型的峰值信噪比D1-PSNR与D2-PSNR曲线，根据以上两种指标来评价本章节模型的性能。绘制的曲线以码率作为横坐标，PSNR为纵坐标。在同一垂直线上的值表示相同PSNR条件下，需要的码率更少，性能则更高。由下表4.1与4.2数据所示，本章节方法在测试数据上相比GPCC(octree)达到了约90%的码率节省，相比于PCGC模型有约10%的码率节省，在单个点云文件上的表现也均优于各个传统方法。

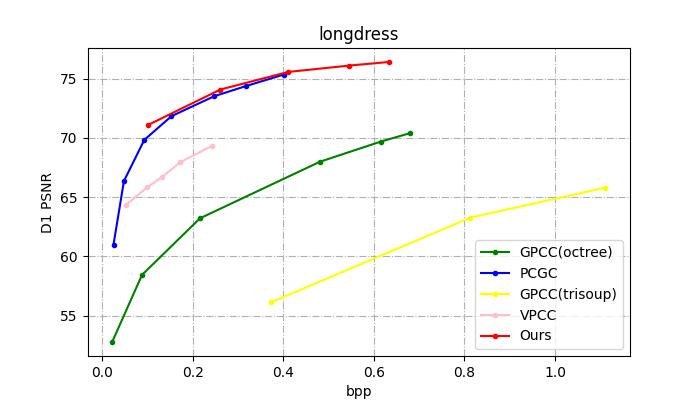
表 4.1 8iVFB测试数据码率优化，（-）代表曲线拟合度不足无法统计

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试点云 | GPCC(octree)  D1 D2 | | GPCC(trisoup)  D1 D2 | | VPCC  D1 D2 | | PCGC  D1 D2 | |
| Longdress | -88.59 | -84.64 | - | - | -41.23 | -38.98 | -9.77 | -7.6 |
| Loot | -88.90 | -83.92 | -79.34 | -88.97 | -39.45 | -38.33 | -8.57 | -5.36 |
| Redandblack | -88.84 | -84.32 | -79.14 | -78.21 | -40.33 | -39.27 | -9.98 | -9.33 |
| Soldier | -87.32 | -82.98 | -78.24 | -75.33 | -40.02 | -37.83 | -8.57 | -7.86 |

表 4.2 本文方法相较于其他方法的码率优化

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 压缩模型 | 8iVFB  D1 D2 | | MVUB  D1 D2 | | Owlii  D1 D2 | | 平均  D1 D2 | |
| GPCC(octree) | -88.41 | -84.97 | -91.41 | -87.33 | -86.24 | -79.84 | -88.67 | -84.05 |
| GPCC(trisoup) | -78.91 | -80.84 | -74.98 | -71.82 | -80.53 | -76.25 | -78.14 | -74.86 |
| VPCC | -40.26 | -38.60 | -51.24 | -45.22 | -41.07 | -37.97 | -44.19 | -40.60 |
| PCGC | -9.22 | -7.54 | -12.38 | -11.80 | -10.91 | -8.30 | -10.84 | -9.21 |

如图4.6与4.7展示了Longdress与Loot点云在不同压缩模型下的D1-PSNR与D2-PSNR曲线，可见我们设计的模型大幅超越了MPEG标准下的传统算法，相较于PCGC模型也得到了一定的提升，结果表明我们设计的模型具备更好的压缩效果。



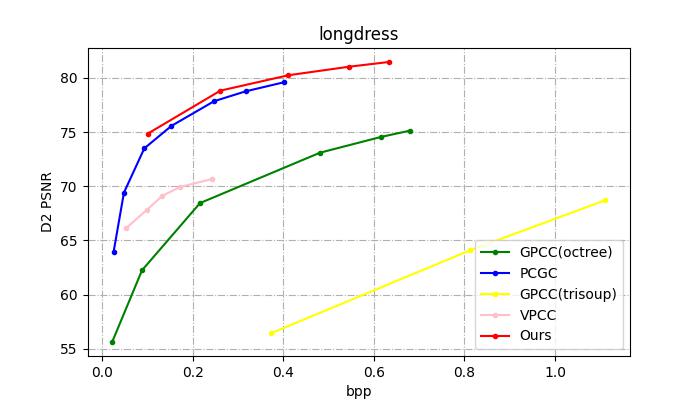
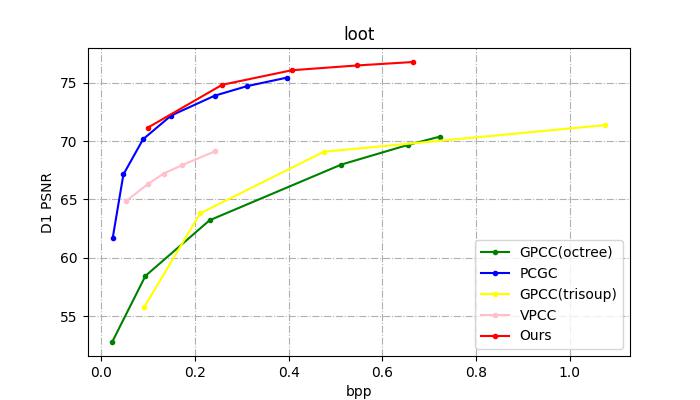


图 4.6 longdress点云在不同压缩模型下的D1-PSNR与D2-PSNR曲线



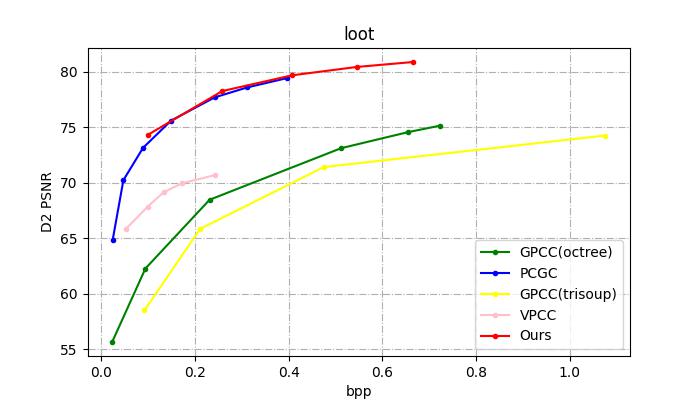


图 4.7 loot点云在不同压缩模型下的D1-PSNR与D2-PSNR曲线

除了对D1-PSNR与D2-PSNR曲线进行比较外，我们还实验了不同码率下编解码8iVFB数据集上Longdress点云的时间消耗，为了保证各个算法的复杂度对比公平，我们在相同的设备上对于高码率以及低码率的点云模型各进行了5次编码与解码实验，对结果取平均值以确定最终的耗时。由表4.3数据可知，MPEG标准下的GPCC与VPCC方法的编码耗时均较高，本章节设计模型的编解码耗时较低，并且略优于PCGC模型。

表 4.3 高码率与低码率的编解码耗时

|  |  |  |  |
| --- | --- | --- | --- |
| 压缩模型 | 码率 | 编码时间/s | 解码时间/s |
| GPCC(octree) | 低码率 | 3.11 | 0.15 |
| 高码率 | 33.45 | 0.87 |
| GPCC(trisoup) | 低码率 | 3.45 | 0.23 |
| 高码率 | 48.32 | 1.35 |
| VPCC | 低码率 | 48.10 | 0.45 |
| 高码率 | 79.13 | 1.34 |
| PCGC | 低码率 | 5.93 | 0.95 |
| 高码率 | 7.97 | 1.02 |
| Ours | 低码率 | 3.62 | 0.89 |
| 高码率 | 5.80 | 0.93 |

另外，我们对不同方法在近似码率的重建点云进行了可视化，通过图像4.8能够直观地看出相近码率下VPCC存在略微的空洞部分，GPCC(octree)与GPCC(trisoup)均有明显的重建空洞，PCGC与本章节模型均有较好地重建效果在视觉上差距较小，但本章节模型在略低于PCGC模型的码率上仍然能有相似的重建效果。由此可见本章节模型能够较好地保证压缩重建后的点云模型质量较高，失真较低。



图 4.8 不同方法的重建效果图

1. 原始 ② VPCC0.243bpp ③ GPCC(octree)0.216bpp ④ GPCC(trisoup)0.373bpp ⑤ PCGC0.287bpp ⑥ Ours0.261bpp
   1. 工业数据集的应用

在先前章节已经通过对比实验的形式验证了本章设计的基于深度学习的点云压缩框架在公开数据集上相比于其他模型的优势。为了验证本章节设计的模型能够有效地在工业领域进行点云压缩任务，同时能够展示基于深度学习的点云压缩框架的优势，我们设计了一个工业数据集，随后通过与传统PCL点云压缩算法对比凸显了该模型的优秀压缩重建效果。

* + 1. 工业数据集的构建

由于现今点云压缩领域缺少公开的工业部件数据集，想要验证本章模型的效果要设计适应工业领域的数据集进行测试。

进行工业数据集地制作可以通过激光雷达等仪器采集，但是会受到仪器与环境的较大限制，于是我们通过网络收集与分类工业部件的3D网格模型，使用网格采样法生成各种工业部件的完整点云，然后通过对各种工业部件点云进行X、Y、Z轴的旋转操作生成多个ASCII格式的ply点云文件，最后通过python的h5py与numpy库生成h5文件构建数据集。

网格采样能够确保生成点云的均匀性，同时最大限度地减少点重叠保证生成点云的基本质量。为了生成表面清晰无噪声的完整点云，我们采样了多种工业零件obj网格模型，包括各样式的螺栓、叶片、螺帽等工业部件网格模型，对收集到的工业部件网格模型采用泊松盘采样技术进行点云生成。生成的一些工业零件点云通过open3d可视化后如图4.9所示，其中某零件点云细节通过Cloud Compare可视化观察如图4.10所示。



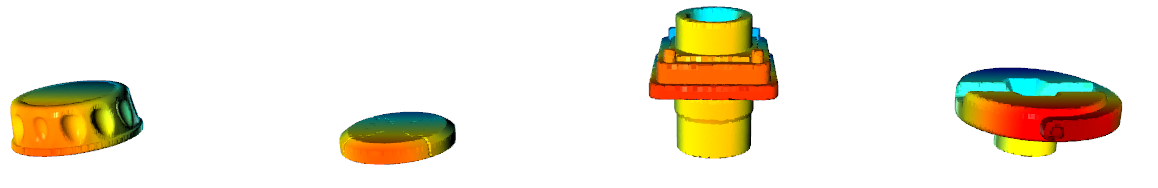


图 4.9 多种工业零件点云的可视化效果

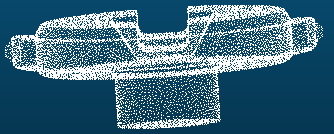


图 4.10 某工业零件点云细节

* + 1. 实验测试

为了将本章设计的压缩模型应用于实际点云压缩任务中，本节对数据集中一种圆顶零件点云进行了测试，该点云原始点个数为458781，原始ply文件大小为12.6MB，该点云网格模型可视化具体如下图4.11所示。体素下采样压缩设置体素中心保留1个点，八叉树分辨率为0.01，PCL无损压缩设置点坐标的精度为0.001，八叉树分辨率为0.01，本章模型采用训练的bpp约为1.3的模型。

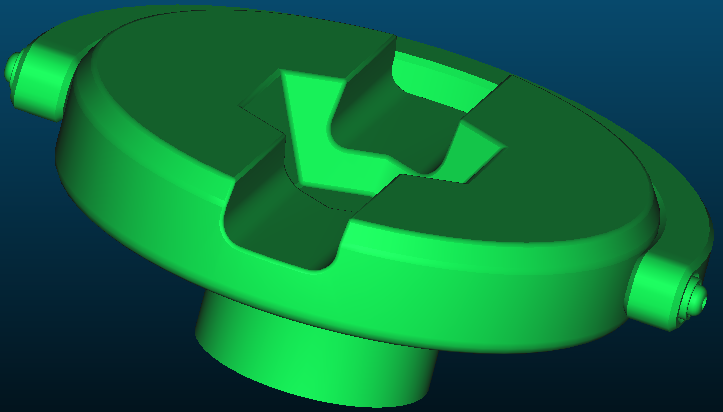


图 4.11 点云测试模型可视化效果

由表4.4中数据可以观察发现本章模型在压缩文件大小相比其他两种编码方式取得了巨大的提升，这得益于本章模型将点云几何坐标与体素占用二值信息分开编码的模式。在编码时间上本章模型相较于其他两种算法也有一定的优势。

表 4.4 不同方法压缩效果对比

|  |  |  |  |
| --- | --- | --- | --- |
| 压缩方法 | 解码后点数 | 文件大小/MB | 编解码时间/s |
| 体素下采样压缩 | 444032 | 0.49 | 1.599 |
| PCL无损压缩 | 458781 | 1.04 | 2.075 |
| 本文方法 | 442883 | 0.03 | 1.065 |

该工业零件的重建结果如下图所示。经过编解码后，对工业部件进行泊松三维重建并进行可视化操作如图4.12所示，在采用传统泊松重建的情况下，可以观察到重建后点云虽然存在略微的边缘褶皱，但是整体上仍然能够较高程度地接近原始模型。结合压缩体积、编解码耗时与重建效果这几项数据指标的实验可以证明该模型能够有效地运用于工业部件点云的压缩，成功实践了基于深度学习的点云压缩框架在工业零件压缩的可行性。

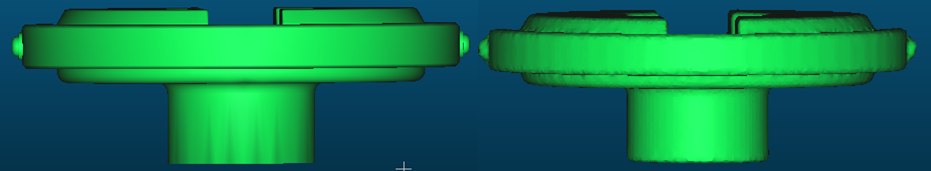


图 4.12 重建模型可视化效果（左侧为原始模型，右侧为实验模型）

* 1. 本章小结

本章提出了一种基于优化残差块与逐通道卷积的深度学习点云压缩框架，旨在使用深度学习的方式设计一种点云压缩方法并投入工业领域压缩任务中。首先，该方法通过稀疏卷积结合深度可分离卷积与残差网络的结构进行点云特征信息的多尺度编码，随后使用八叉树编码器和算数编码器分别编码点云位置信息与体素占用信息，接着通过多尺度的上采样逐步恢复点云特征，然后通过比较率失真曲线、编解码时间与重建视觉效果来验证本模型的效果。在公开数据集的实验对比结果表明，本章方法能够取得更好的码率节省、时间节省以及重建质量。最后，我们构建了工业数据集并应用于本章模型的训练中，通过与传统点云压缩算法的实验对比验证了本章设计的基于深度学习的点云压缩算法的有效性。

# 面向工业点云的压缩系统设计与实现

本章基于前文章节研究的内容，设计并实现了一套面向工业点云的压缩系统，该系统集成了点云格式转换、三维重建与可视化等多种功能，支持了工业点云的压缩解压流程。接下来，本章将对本章系统功能进行详细地介绍与分析。

* 1. 需求分析

该系统的主要功能是输入特定格式的点云文件获得相应的压缩文件与解压文件，由于工业点云压缩是本文研究的重点，所以本系统的压缩对象为工业点云。根据系统使用的步骤，用户首先通过文件选择的方式选取需要压缩的点云文件与文件保存路径，并可通过格式转换功能将目标点云的格式转换为符合系统运行的格式。在完成了对象点云的设置后，用户可以使用点云压缩功能对输入的点云文件进行压缩，随后能在页面显示压缩结果，并能在文件保存路径获得压缩后的文件。另外，用户也能对压缩后的文件进行解压获得解压点云文件。最后，用户可以对解压完成的点云进行三维重建获取Mesh模型，以上每一个环节获得的点云文件或Mesh模型均能通过文件选择的方式进行可视化显示。

基于以上分析，本系统可能会使用到以下功能：

（1）格式编辑功能，该功能包括文件编辑、格式转换与算法选择功能。此功能是系统的基础，为本系统的核心功能提供了正确格式的目标文件。在该功能中，系统为用户提供了格式转换的选择，包括pcd转换ply格式与obj转ply格两个选项。另外，系统提供了目标点云文件路径、保存路径、压缩算法的选择。

（2）压缩功能，该功能包括点云压缩、点云解压缩与点云重建功能。此功能是系统的核心。用户选择好压缩算法后，系统便能够使用配置好的算法对选取的目标点云文件进行压缩，压缩完成的提示信息与压缩后的文件属性会在窗体界面上显示。另外用户可以选择解压功能，将上一步压缩好的文件通过深度学习框架的解码器进行特征重建解压。最后，用户可以使用点云重建功能，将解压后的点云文件进行泊松重建获取Mesh模型。

（3）可视化功能，该功能包括文件预览与点云可视化功能。该功能为系统提供了一定的

辅助。用户能够在系统界面上直接可视化观察选择的文件，并且在压缩、解压、重建后能够通过文件选取的方式进行文件可视化显示。

* 1. 系统设计

为了满足需求分析中的功能，下一步则是进行具体的系统设计。结合前文对点云压缩方法的研究，本系统的使用方式如图5.1所示，具体有以下几个步骤：首先，用户能够进行文件选择，选取本系统适用的二进制ply文件格式的点云，当格式不满足要求时用户可以使用格式转换功能生成符合规范的点云文件，并可以在界面预览点云细节。接着，用户能够选择压缩文件的保存路径，避免文件丢失。最后，用户可以使用需求的算法进行点云压缩、解压与重建并获取各个阶段的文件进行可视化查看。

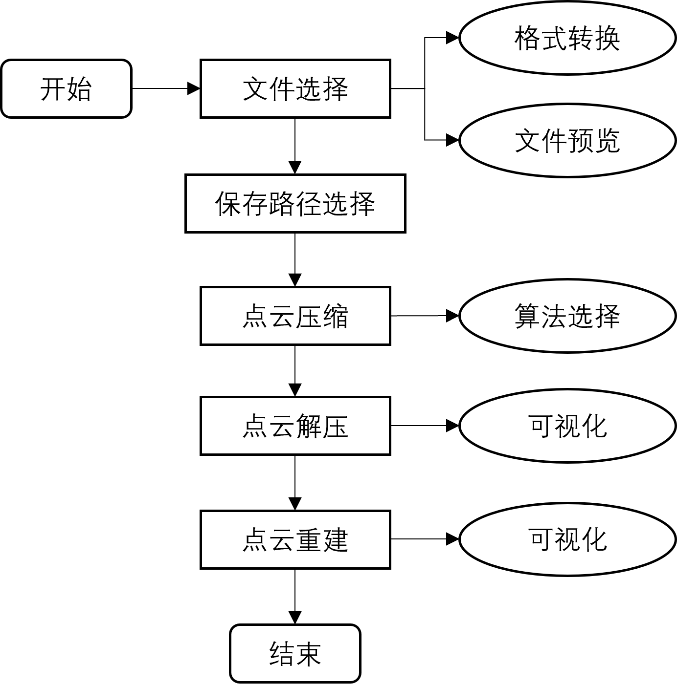


图 5.1 系统流程图

* + 1. 系统架构设计

系统架构是软件设计的核心，它定义了软件的整体构造、组成部分及其层次。考虑到系统的功能与使用范围，本文设计的面向工业点云的压缩系统将系统的架构设计为视图层与业务层相结合的形式，如图5.2所示。

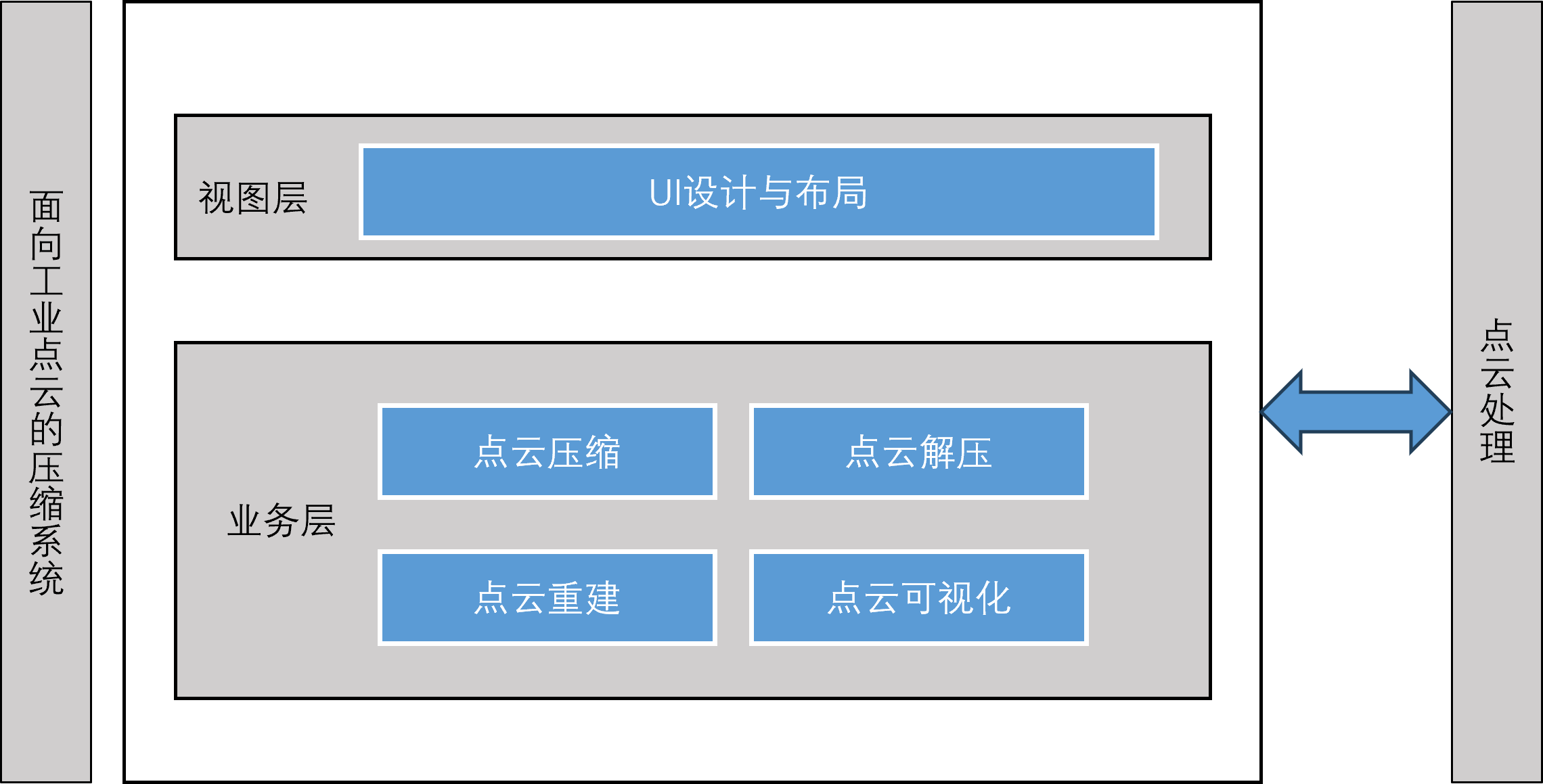


图5.2 系统架构设计图

视图层是用户与系统交互的门户，包含了所有的功能模块，良好的UI设计与布局能够清晰地将功能展示给用户，达到良好的使用体验。在该层，本系统使用了菜单栏与UI控件结合的方式将功能模块更加直观地展示给用户。业务层是系统的核心，为用户提供了解决问题的功能。在设计业务层时必须保证模块与模块间的独立以提高系统功能的可扩展性与可维护性，便于后期添加新的功能。本系统为用户提供了多种核心业务模块例如点云压缩、点云解压、点云重建、点云可视化。总体上，该系统采用了简单的视图业务架构，更便捷地将功能提供给用户。

* + 1. 系统详细设计

考虑到实验环境的限制，本系统在 Windows平台上，使用python语言进行开发，视图UI界面通过PyQt5与Qt Designer进行设计。为了在该系统下实现工业点云压缩的任务，该系统结合了上文设计的压缩算法与深度压缩框架，设计了格式编辑模块、点云压缩模块、可视化模块，如图5.3所示。

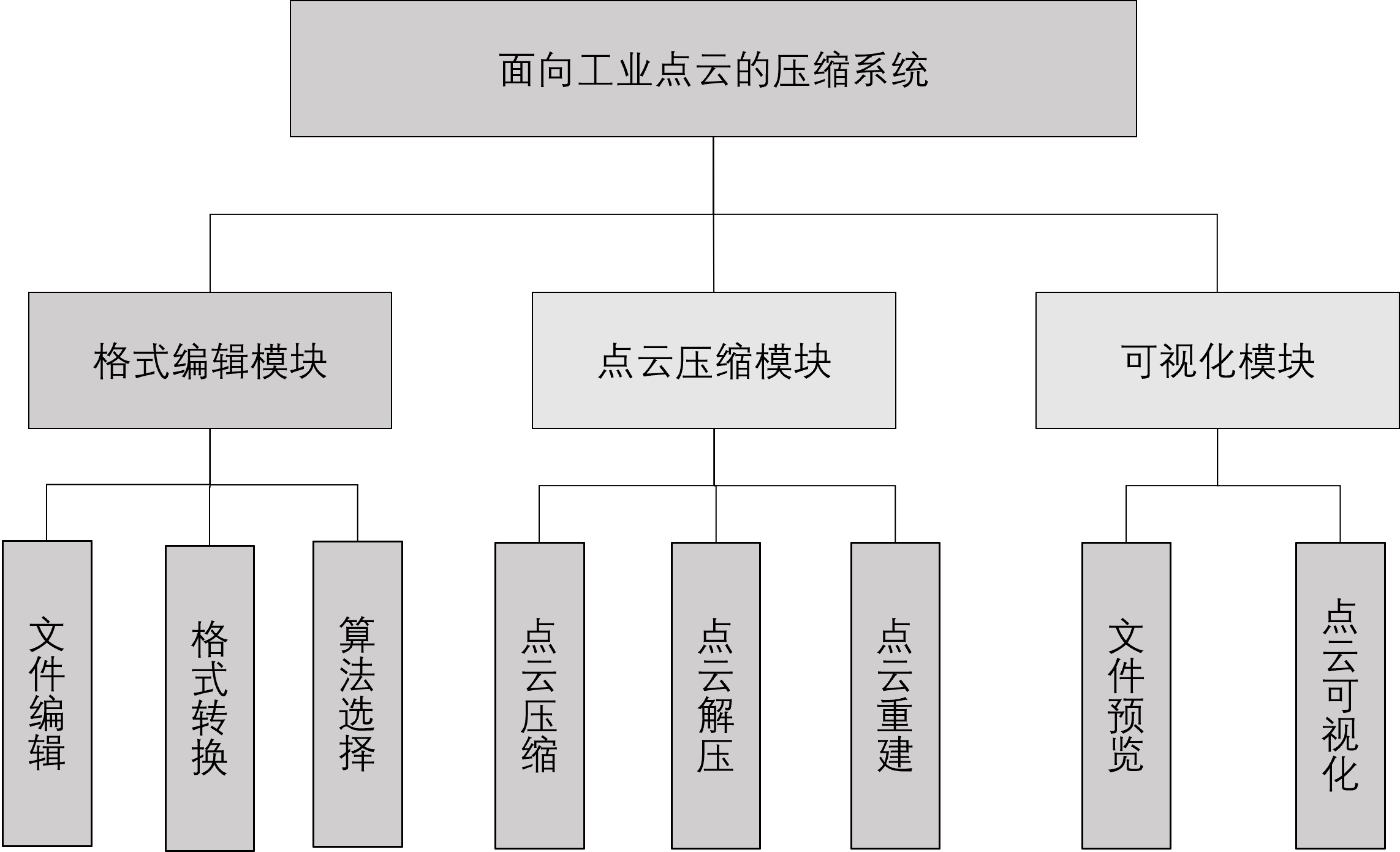


图5.3 系统详细模块示意图

格式编辑模块包含文件编辑、格式转换、算法选择功能。其中格式转换功能通过UI设计了下拉选择菜单栏，使用点击事件触发，提供了Pcd文件转Ply文件、Obj文件转Ply文件的功能，结果在保存目录下生成了对应的二进制Ply点云文件。算法选择功能同样采用了菜单栏的形式，通过点击选定算法，该功能提供了第三章算法与第四章深度学习模型两种压缩方式，旨在适应不同场景要求下的点云压缩。文件编辑功能的实现通过使用了LineEdit组件，设计了按钮点击触发文件目录选择，将当前选中的文件在页面上显示。

点云压缩模块包含点云压缩、点云解压、点云重建功能。其中点云压缩与点云解压功能会读取菜单栏中选中的算法做出相应的配置，对选择的点云文件使用相应的算法进行压缩或解压，具体压缩结果与解压结果通过TextEdit组件进行显示。点云重建功能使用了Open3D包下的泊松重建算法，对选定的重建点云模型进行重建，重建的点云Mesh文件会保存在提前设置好的保存路径中。

可视化模块包含文件预览与点云可视化功能。其中文件预览功能使用GroupBox作为背景，GLViewWidget作为可视化容器，通过 pyqtgraph库的opengl包与open3d库实现了当前选中文件的预览，在系统的UI界面上以小窗口的形式预览当前文件的可视化效果。点云可视化功能使用了GraphicsView组件，通过文件选取的形式实现了选中点云或模型的可视化。

* 1. 系统功能测试

本节通过模拟一次点云压缩的过程来对本章设计的系统进行功能模块的测试。在实验前，我们选取了一个工业零件的点云文件作为测试输入，在实验过程中按照章节5.2中的运行流程进行了一套完整的操作测试，具体的模块测试使用情况如下文所示。

* + 1. 格式编辑模块测试

格式编辑模块的使用如图5.4所示，本系统通过上方菜单栏的形式提供了算法选择与格式转换的功能，通过文件选择按钮点触的方式选择了目标点云文件。结果显示，当前文件与保存目录均能正确显示。



图 5.4 格式编辑模块测试

* + 1. 压缩模块测试

UI界面的左侧边栏中提供了点云压缩、点云解压、点云重建、清空页面几项功能，依次点击这几个按钮，窗体中心的信息框就会跳出相应的运行信息包括文件信息与保存路径等。图5.5展示了一次压缩解压重建过程的信息反馈，压缩文件选取了第四章构建的工业零件数据集中某零件点云通过旋转操作生成的文件。由图5.5信息框中文字可知，该文件通过深度压缩框架被成功压缩，其压缩文件也被存储于保存路径下的文件夹中。



图 5.5 压缩模块测试

* + 1. 可视化模块测试

界面右侧的预览按钮提供了当前文件选择路径下的点云文件的小窗预览，便于用户在选择文件的同时观察目标点云。如图5.6中右侧小窗所示，目标文件的预览功能可以正常运作，在该小窗中能够通过鼠标左键与滚轮进行点云的三维空间放大缩小旋转操作。界面右下方的按钮提供了文件可视化的功能，单击该按钮便可通过文件选择的方式查看目标文件的可视化效果，如图5.7所示。



图 5.6 文件预览测试



图 5.7 结果可视化测试

* 1. 本章小结

在本章中，我们主要介绍了一个面向工业点云的压缩系统，旨在对工业点云进行高水平的压缩。接着详细分析了构建系统的主要需求功能，将系统需求的功能细化为格式编辑、点云压缩与可视化三个模块，并对这几个模块进行了界面与触发事件的代码实现。为了测试该系统的功能，我们通过实验的方式使用了设计的功能，结果显示，这三个模块的功能均能正常运行，可以为工业点云零件压缩提供有效的解决方案。

# 总结与展望

* 1. 总结

本文首先针对传统算法领域点云压缩方法难以有效地保留表面复杂工业点云的缺陷点特征的问题，提出了一种缺陷点分离的、面向复杂表面工业产品的点云压缩方法。此外，为了拓展点云压缩在深度学习领域的研究并为工业点云提供一种深度压缩方式，我们提出了一种结合优化残差块与逐通道卷积的深度点云压缩框架。现对本文的总结如下：

（1）本文提出了一种基于缺陷点分离的工业产品的点云压缩方法，该方法首先对目标点云进行缺陷点提取分为缺陷点云与非缺陷点云两部分。然后，通过曲率分级法对缺陷点云进行压缩，使用改进的跨点前进法对非缺陷点进行压缩。接着，将两部分压缩后的点云组合形成完整的压缩后的点云。最后，对压缩后的点云使用泊松重建获取重建模型并进行压缩效果的评价。通过实验对比的结果可见，该方法相较于其他算法能够实现较好的压缩率与较短的压缩时间，更重要的是能够实现更好的重建精度与可视化质量，保证了缺陷点云特征的高效保留。

（2）本文提出了一种基于优化残差块与逐通道卷积的点云压缩框架。该方法首先使用了稀疏卷积作为初始特征输入，在编码器端的一个压缩尺度上使用了优化的下采样模块与基于深度可分离卷积的逐通道卷积特征提取模块。接着，在编码过程中进行连续三次的下采样与特征提取完成多尺度的特征保留。然后，通过算数编码器与八叉树编码器结合的形式对提取的特征进行熵编码。最后，在解码器端将编码的码流数据通过多尺度上采样特征复原的方式进行了点云的特征重建获取重建后的点云。该方法在公开数据集上与多种基准算法进行了实验比较，验证了本文设计的深度点云压缩框架在公开领域上是行之有效的。另外，本文构建了一个工业零件数据集，并在提出的深度压缩框架上进行了训练，测试结果表明，该框架在工业点云数据集上能够很好地完成点云压缩任务，得到高压缩率与高重建质量的点云文件。

（3）为了验证本文提出的基于缺陷点分离的工业产品点云压缩方法与基于优化残差块与逐通道卷积的点云压缩框架的应用价值，本文结合这两种算法框架设计了一个面向工业点云的压缩系统。该系统包括格式编辑、点云压缩与点云可视化模块。该系统允许用户使用本文研究的两种算法框架进行点云压缩任务，通过一键调用的形式对目标点云进行压缩、解压、重建与可视化作用。

* 1. 展望

通过实验结果初步证明本文设计的一种面向复杂表面工业产品的点云压缩方法与基于优化残差块与逐通道卷积的点云压缩框架均能够有效地进行点云压缩任务。然而，我们也在实验与应用中发现了一些不足之处，具体问题如下所示：

（1）在基于缺陷点分离的工业产品点云压缩方法的研究中，缺乏一种高效的点云缺陷提取方式。本文实验的目标点云均包含一处密集缺陷点，通过索引提取的方式能够有较高的效率。然而对于多处密集缺陷点的情况，该方法在提取缺陷点的效率便十分低下。因此，我们的下一步工作是研究并设计一种高效的工业点云缺陷点提取方式。

（2）在基于优化残差块与逐通道卷积的点云压缩框架的研究中，我们构建了一套工业零件数据集，实现了深度学习方式的点云压缩功能。然而，该数据集涵盖的点云类型与数量规模较小，且点云精度稍低。因此，我们的下一步计划是通过使用精密的激光雷达等仪器提取更多更高质量的工业点云以进一步扩充工业零件点云数据集。

# 参考文献

1. 艾达, 卢洪颖, 杨玉蓉, 等. 三维点云数据压缩技术研究综述[J]. 西安邮电大学学报, 2021, 26(1): 90-96.
2. Cao C, Preda M, Zaharia T. 3D point cloud compression: A survey[C]//Proceedings of the 24th International Conference on 3D Web Technology. 2019: 1-9.
3. Liu G P, Song Z H. Adaptive network bandwidth control method for H. 264 video stream transmission[J]. Journal of Zhejiang University. Engineering Science, 2012, 46(12): 2146-2154.
4. Yuan W, Khot T, Held D, et al. Pcn: Point completion network[C]//2018 international conference on 3D vision (3DV). IEEE, 2018: 728-737.
5. Pomerleau F, Colas F, Siegwart R. A review of point cloud registration algorithms for mobile robotics[J]. Foundations and Trends® in Robotics, 2015, 4(1): 1-104.
6. Jenke P, Wand M, Bokeloh M, et al. Bayesian point cloud reconstruction[C]//Computer graphics forum. Oxford, UK and Boston, USA: Blackwell Publishing, Inc, 2006, 25(3): 379-388.
7. Edl M, Mizerák M, Trojan J. 3D laser scanners: history and applications[J]. Acta Simulatio, 2018, 4(4): 1-5.
8. LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature, 2015, 521(7553): 436-444.
9. Xu Y, Zhang K, He L, et al. Introduction to point cloud compression[J]. ZTE Communications, 2018, 16(3): 8.
10. He Y, Ren X, Tang D, et al. Density-preserving deep point cloud compression[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 2333-2342.
11. Sun H, Kwok W, Zdepski J W. Architectures for MPEG compressed bitstream scaling[J]. IEEE transactions on Circuits and Systems for Video Technology, 1996, 6(2): 191-199.
12. Chua, Leon O.CNN: A Paradigm for Complexity[M]. WORLD SCIENTIFIC,1998.
13. Mao J, Wang X, Li H. Interpolated convolutional networks for 3d point cloud understanding[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 1578-1587.
14. 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.
15. Dai A. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017.DOI:10.1109/CVPR.2017.261.
16. Behley J, Garbade M, Milioto A, et al. Semantickitti: A dataset for semantic scene understanding of lidar sequences[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 9297-9307.
17. 孙吉贵, 刘杰, 赵连宇. 聚类算法研究[J].软件学报, 2008(1):14.DOI:10.3724/SP.J.1001.2008.00048.
18. Botsch M, Wiratanaya A, Kobbelt L. Efficient high quality rendering of point sampled geometry[C]// Eurographics. Eurographics Association, 2002. DOI: doi: http://dx.doi.org/.
19. Meagher D. Geometric modeling using octree encoding[J]. Computer graphics and image processing, 1982, 19(2): 129-147.
20. Chen Y J D. Lossless compression of point-based 3D models[J].Pacific Graphics, 2008.DOI:http://dx.doi.org/.
21. Moore A W. An introductory tutorial on kd-trees[J]. Computer Laboratory, University of Cambridge, Tech. Rep, 1991, 209.
22. 危育冰. 基于八叉树编码的散乱点云数据压缩[J].武汉大学学报:工学版, 2020, 53(8):6.DOI:10.14188/j.1671-8844.2020-08-012.
23. Pan Y. Dynamic Update of Sparse Voxel Octree Based on Morton Code[D]. Purdue University, 2021.
24. Han H , Han X , Sun F ,et al.Point cloud simplification with preserved edge based on normal vector[J].Optik: Zeitschrift fur Licht- und Elektronenoptik: = Journal for Light-and Electronoptic, 2015(126-19).
25. 王艺楠, 郝矿荣, 杨焕宇. 基于FPFH特征和模糊聚类的自适应点云压缩[J]. 电子科技, 2017, 030(011):73-77,80.
26. 张旭康, 牛保宁, 张锦文. 向量相似度可复原三维点云压缩算法[J]. 计算机科学与探索, 2020, 14(4):12.DOI:10.3778/j.issn.1673-9418.1903033.
27. Quach M, Valenzise G, Dufaux F. Improved Deep Point Cloud Geometry Compression[J]. 2020.DOI:10.1109/MMSP48831.2020.9287077.
28. Isik B, Chou P A, Hwang S J, et al. LVAC: Learned Volumetric Attribute Compression for Point Clouds using Coordinate Based Networks.[J]. 2021. DOI:10.48550/arXiv.2111.08988.
29. Schwarz S, Preda M, Baroncini V, et al. Emerging MPEG standards for point cloud compression[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2018, 9(1): 133-148.
30. Gao L, Fan T, Wan J, et al. Point Cloud Geometry Compression Via Neural Graph Sampling[C]//IEEE International Conference on Image Processing.IEEE, 2021.DOI:10.1109/ICIP42928.2021.9506631.
31. Nguyen D T, Quach M, Valenzise G, et al. Learning-based lossless compression of 3D point cloud geometry[J]. 2020.DOI:10.48550/arXiv.2011.14700.
32. Liang Y, Zhong R, Ren P. A research on organization and rapid allocation method for large-scale 3D points cloud data[C]//The 2nd International Conference on Information Science and Engineering. IEEE, 2010: 6410-6413.
33. You K, Gao P, Ma Z. Efficient and Generic Point Model for Lossless Point Cloud Attribute Compression[J]. arXiv preprint arXiv:2404.06936, 2024.
34. Huang T, Liu Y. 3d point cloud geometry compression on deep learning[C]//Proceedings of the 27th ACM international conference on multimedia. 2019: 890-898.
35. Zhang C, Florencio D, Loop C. Point cloud attribute compression with graph transform[C]//2014 IEEE International Conference on Image Processing (ICIP). IEEE, 2014: 2066-2070.
36. Javaheri A, Brites C, Pereira F, et al. Improving PSNR-based quality metrics performance for point cloud geometry[C]//2020 IEEE International Conference on Image Processing (ICIP). IEEE, 2020: 3438-3442.
37. Rusu R B, Cousins S. 3d is here: Point cloud library (pcl)[C]//2011 IEEE international conference on robotics and automation. IEEE, 2011: 1-4.
38. Graziosi D, Nakagami O, Kuma S, et al. An overview of ongoing point cloud compression standardization activities: Video-based (V-PCC) and geometry-based (G-PCC)[J]. APSIPA Transactions on Signal and Information Processing, 2020, 9: e13.
39. Jang E S, Preda M, Mammou K, et al. Video-based point-cloud-compression standard in MPEG: From evidence collection to committee draft [standards in a nutshell][J]. IEEE Signal Processing Magazine, 2019, 36(3): 118-123.
40. Schwarz S, Preda M, Baroncini V, et al. Emerging MPEG standards for point cloud compression[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2018, 9(1): 133-148.
41. Freitas D R, Peixoto E, de Queiroz R L, et al. Lossy point cloud geometry compression via dyadic decomposition[C]//2020 IEEE International Conference on Image Processing (ICIP). IEEE, 2020: 2731-2735.
42. Hu Z, Chen Z, Xu D, et al. Improving deep video compression by resolution-adaptive flow coding[C]//Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. Springer International Publishing, 2020: 193-209.
43. Schnabel R, Klein R. Octree-based Point-Cloud Compression[J]. PBG@ SIGGRAPH, 2006, 3: 111-121.
44. De Queiroz R L, Chou P A. Compression of 3D point clouds using a region-adaptive hierarchical transform[J]. IEEE Transactions on Image Processing, 2016, 25(8): 3947-3956.
45. Li L, Li Z, Zakharchenko V, et al. Advanced 3D motion prediction for video-based dynamic point cloud compression[J]. IEEE Transactions on Image Processing, 2019, 29: 289-302.
46. Ji S, Xu W, Yang M, et al. 3D convolutional neural networks for human action recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2012, 35(1): 221-231.
47. Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2818-2826.
48. Al-Qizwini M, Barjasteh I, Al-Qassab H, et al. Deep learning algorithm for autonomous driving using googlenet[C]//2017 IEEE intelligent vehicles symposium (IV). IEEE, 2017: 89-96.
49. Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, inception-resnet and the impact of residual connections on learning[C]//Proceedings of the AAAI conference on artificial intelligence. 2017, 31(1).
50. Lin J. Divergence measures based on the Shannon entropy[J]. IEEE Transactions on Information theory, 1991, 37(1): 145-151.
51. Moffat A. Huffman coding[J]. ACM Computing Surveys (CSUR), 2019, 52(4): 1-35.
52. Witten I H, Neal R M, Cleary J G. Arithmetic coding for data compression[J]. Communications of the ACM, 1987, 30(6): 520-540.
53. 何宽, 孙瑞, 官云兰, 等. 基于逐点前进法的点云数据精简[J]. 测绘通报, 2022 (9): 167.
54. Zhao L, Shi G. A trajectory clustering method based on Douglas-Peucker compression and density for marine traffic pattern recognition[J]. Ocean Engineering, 2019, 172: 456-467.
55. 裴书玉, 杜宁, 王莉, 等. 基于自适应分层的文物点云数据压缩算法[J]. Application Research of Computers/Jisuanji Yingyong Yanjiu, 2018, 35(11).
56. Zhao Y, Jie Z, Zhang Y, et al. Negative Gaussian curvature regulated pattern evolution on curved bilayer system[J]. International Journal of Mechanical Sciences, 2024, 267: 108969.
57. Dewez T J B, Girardeau-Montaut D, Allanic C, et al. Facets: A cloudcompare plugin to extract geological planes from unstructured 3d point clouds[J]. ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2016, 41: 799-804.
58. 李金涛, 程效军, 杨泽鑫, 等. 基于曲率分级的点云数据压缩方法[J]. Laser & Optoelectronics Progress, 2019, 56(14): 142801.
59. Nguyen D T, Kaup A. Lossless point cloud geometry and attribute compression using a learned conditional probability model[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2023.
60. Wang J, Ding D, Li Z, et al. Multiscale point cloud geometry compression[C]//2021 Data Compression Conference (DCC). IEEE, 2021: 73-82.
61. Liang Z, Liang F. TransPCC: towards deep point cloud compression via transformers[C]//Proceedings of the 2022 International Conference on Multimedia Retrieval. 2022: 1-5.
62. Choy C, Gwak J Y, Savarese S. 4d spatio-temporal convnets: Minkowski convolutional neural networks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 3075-3084.
63. Luo G, He B, Xiong Y, et al. An optimized convolutional neural network for the 3d point-cloud compression[J]. Sensors, 2023, 23(4): 2250.

# 附录1 攻读硕士学位期间撰写的论文

[1] 导师第一作者,本人第二作者, International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). IEEE, 2023年9月已录用.

# 附录2 攻读硕士学位期间申请的专利

[1] 发明专利, 导师第一发明人, 本人第二发明人, 2024年1月公开

# 附录3 攻读硕士学位期间参加的科研项目

(1) 参与了一项工业软件重点研发计划子课题.