

班级 1901015

学号 19010100277

本科毕业设计（论文）

外文资料翻译

毕业设计题目	G-PCC Trisoup 点云几何信息 编码优化
--------	------------------------------

外文资料题目	A Survey on 3D Point Cloud Compression Using Machine Learning Approaches
--------	--

学 院	通信工程学院
-----	--------

专 业	通信工程
-----	------

学 生 姓 名	姚凯
---------	----

指导教师姓名	张伟
--------	----

摘要

机器学习已被广泛应用于解决各种数据处理任务，最近在数据压缩领域也开始有了应用，特别是在点云（PC）领域。基于深度学习（DL）方法的压缩技术，如卷积神经网络（CNN），已经能够利用高位相关性来提高性能。对于点云压缩（PCC），最常见的基于 DL 的选择是自动编码器，而使用循环神经网络（RNN）和全连接神经网络的实现很少。本文综述了目前使用 ML 方法对 PCC 进行的研究。还包括具有性能度量的基准数据集。本次研究表明，基于机器学习的编码方法的性能与传统编码方法相当，同时指出了未来有希望改进的方向。

关键词：点云压缩，深度学习，自适应编码，卷积神经网络

目 录

摘 要.....	ii
目 录.....	i
第一章 引言.....	1
第二章 自适应编码.....	3
2.1 点云几何自适应编码器 (PCG-AE)	3
2.2 隐式和显式量化的 RD 控制	4
第三章 自适应分层区域变换.....	9
第四章 系数编码.....	13
4.1 拉普拉斯分布式系数的算术编码.....	错误!未定义书签。
4.2 将子带参数送入解码器.....	错误!未定义书签。
4.3 系数编码.....	错误!未定义书签。
第五章 测试结果.....	15
第六章 总结与展望.....	17
致谢.....	错误!未定义书签。
参考文献.....	19

第一章 引言

随着虚拟现实、增强现实（VR、AR）在自动驾驶^[1]、游戏行业^[2]等现代化应用中的使用越来越多，该领域对高效传输 3D 数据的需求上升。点云（PC）是 3D 数据的一种体积表现格式，由 3D 空间中的一组无序点组成。每个点由空间坐标（X,Y,Z）和属性信息组成，例如颜色值（R,G,B）又或者是描述视觉外观^[3]的法向量（ n_x, n_y, n_z ）。PC 提供了对像的高度还原表示，这有助于为用户创建交互式、沉浸式视觉体验。

拥有多个视点的优势在实际应用中很有帮助，可以依靠场景或对象的实时多维视图执行复杂的决策任务^[5]。例如，机器人应用^[6]、文化遗产或者历史建筑保护、5G 系统^[2]的视频流等应用都可以从点云数据表示中受益。

随着 3D 点云采集技术的进步，现在可以更加方便高效的生成高精度的 PC，从而捕捉到更加精细的细节。然而，在通过通信信道^[7]进行 PC 数据传输时，存在带宽增加以及信息存储和处理方面的开销。PC 这种体积数据的传输在无线传输^[8,9]领域更具挑战。

正如文献^[8]所述，由于缺乏有效的编码、自适应的传输方法、有效的质量评估标准和准确的视角预测，使用无线信道传输 PC 数据更具有挑战性。此外，传输 PC 数据需要的资源通常是有限的，因此，有效的点云压缩（PCC）已经成为 PC 实际应用中日益重要的问题^[10]。

在过去的十年里，机器学习（ML）已经成为了一个新兴领域，为图像和视频相关问题提供解决方案。这些利用机器学习的解决方案还包括数据压缩技术，其性能结果传统方法^[12,13]相当。因此，ML 技术被视为点云压缩的潜在解决方案。

此外，人工智能（AI）和 ML 的进步为在实时 PC 视频流中使用 AI、ML 提供了机会^[14]。使用 AI 技术传输 PC 数据在相同网络环境下提供了 30 倍以上的压缩比。

尽管 ML 可以灵活的利用数据中的 3D 相关性，但是由于点云数据的非均匀性和非结构化性质，ML 技术在 PCC 领域的应用并不简单。因此，大多数与 ML 相关的方法都是先进行预处理，将 3D 点云数据转换为相应的 2D 格式数据，然后再利用基于 ML 的压缩算法^[15]。有一系列研究调查提供了关于最近 PCC 技术的概述，

但是据我所知，本文是近年来第一次尝试对基于 ML 的 PCC 解决方案进行结构化回顾。本文包括基准数据集，其性能指标用于评估以前的贡献^[16]。表一总结了指标比较。

表一 基于机器学习的 PCC 方法比较

Paper	Feature	Technique	Dataset	Benchmark	Metrics	Loss Function	Optimizer
[18]	Geometry	CNN-based AE	MPEG	PCL	RD curve, D1	SGD (BCE)	Adam
[19]	Geometry	CNN-based AE	MPEG	MPEG GPCC	RD curve, D1	RD Loss	-
[20]	Geometry	3D CNN AE	Train: ModelNet40 Test: MVUB	MPEG GPCC	D1, D2	RD Loss	Adam
[21]	Geometry	3D Stacked CNN	Train: ShapeNet Test :MPEG, JPEG	MPEG GPCC, PCL MPEG VPCC	D1, D2	RD Loss WBCE	-
[22]	Geometry	FCNN, MLPs	ShapeNet	MPEG GPCC	D1	RD Loss Chamfer Dist	Adam
[23]	Geometry	Folding-based NN	ShapeNet ModelNet10	Fully connected decoder	-	Chamfer Dist	Adam
[24]	Geometry	RNN	TierIV	Octree, JPEG	SNNRMSE	-	-

依据 PC 数据的基本表示形式，与点云压缩相关的研究问题可分为两类：

- 1) 几何压缩：主要对点的位置信息进行压缩
- 2) 属性压缩：目的在于利用给定位置的点的属性值的冗余信息

由于利用 ML 对点云数据进行处理的方法，尤其是与几何压缩^[17]相关的，相对处于发展的初始阶段，因此该领域的一些近期的贡献成为了本文的调查重点。本文涉及的方案分类如图 1 所示。

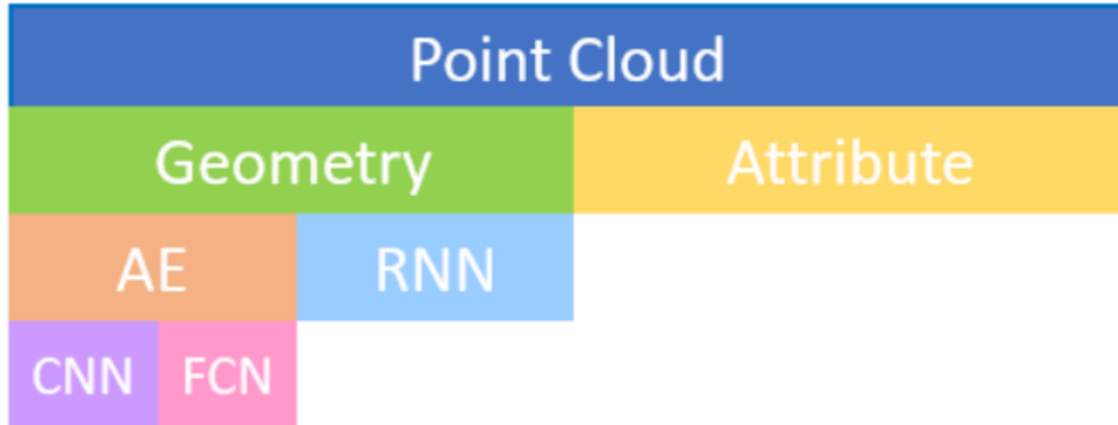


图 1 方案分类

本文的其余部分安排如下：第二节讨论了各种基于 CNN 的 PCC 自动编码实现，并评估了他们的优缺点；第三节介绍了全连接卷积神经网络（FCNN）在 PCC 中的使用；第四节介绍了基于循环神经网络（RNN）的 PCC 方法；第五节讨论了 AI 在传输压缩 PC 数据中的使用以及使用 AI 的潜在优势；第六节总结归纳了本文。

第二章 自适应编码

CNN 被认为是在共享模式、相关性的统一结构中提取有用特征的最有效的神经网络之一。这些特征大多出现在图像和视频中，因此 CNN 被广泛使用。因此，最近提出的基于深度学习的 PC 几何压缩技术采用了涉及卷积层的自编码器(AE)神经网络设计。本节描述了一些最近提出的使用基于 CNN 的自编码器的 PC 几何编码方案。

2.1 点云几何自适应编码器 (PCG-AE)

PCG-AE 为 PCC 提出了 DL 方法的基本实现，以突出 AE PC 编码模式的潜在改进。PC 的一般 3D 坐标表示被更改为一组二进制 3D 块格式，其中体素用“1”和“0”分别表示占用和未被占用。由于 PC 中的点通常以无序的方式排列，使用神经网络对其进行编码是非常具有挑战性的。因此，这种类型的格式引入了一些类似于图像和视频的相关性，以供基于 CNN 的 AE 在 3D 域中使用。其基本思想就是使用自适应的正向和反向变换，而不是使用固定预测变换，如 DCT^[18]。

自适应变换的结构如图 2 所示。首先，将输入的原始点云几何图形划分为大小为 $32 \times 32 \times 32$ 的三维二进制体素块，并将原始点云中的 3D 子块的位置作为最终比特流的索引发送。然后使用 AE 将 PC 几何映射到特征空间中，并将其作为隐数据，如图 2 所示。因此，原始的三维几何现在由编码过程结束时较低的维数表示。然后使用一定的阈值(例如 0.5)对这些隐数据进行二值化以生成比特流。

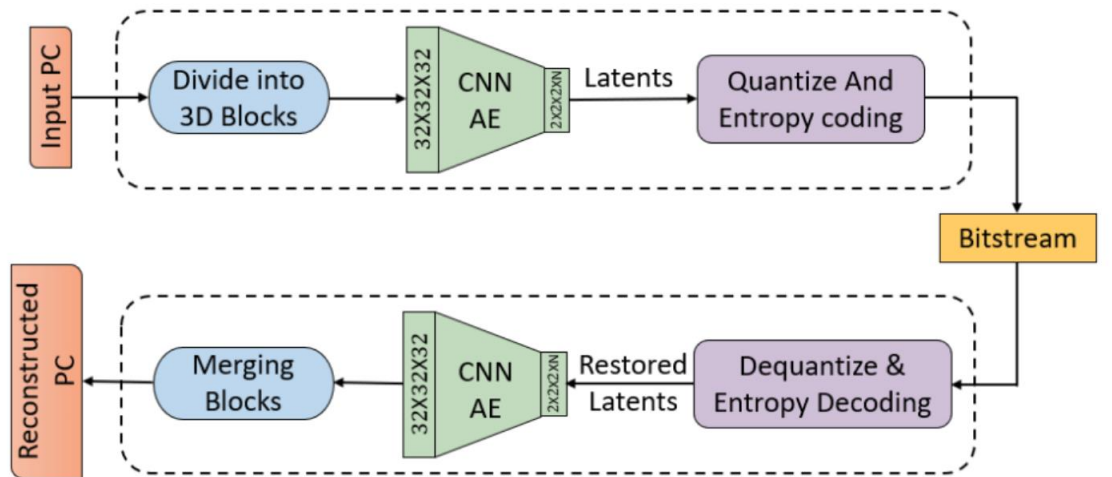


图 2 PCG-AE 编码方案流程图

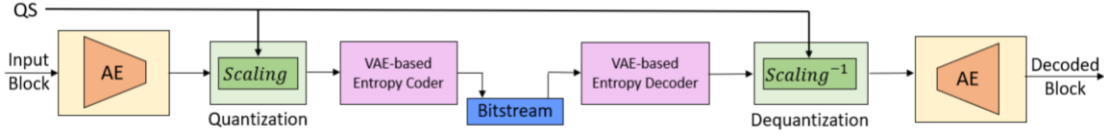


图3 采用隐式-显式量化的基于 DL 的 PC 编码架构

在解码端，通过解码传入的比特流来恢复隐数据，然后将其馈送到 CNN AE，目的是重建尽可能接近原始几何结构的几何结构。然后利用索引作为辅助信息，将恢复后的隐数据的 3D 块放置在各自的位置，最终重建整个 PC。

自适应变换：这个实现使用了一个非常基本的 3D CNN 设计，它是通常用于 2D CNN 图像处理的一个扩展。它在编码器上有 4 个卷积层，在解码器上有 4 个卷积层，每个层的宽度、深度和高度为 3，步幅为 2，以降低隐数据的维度。N 代表过滤器的数量，这是一个超参数，四个模型为这 N 值训练:32,64,96 和 128。N 决定了与比特率相关的重建质量，从而影响了 RD 性能。N 经过实验调整。所有 CNN 层使用 sigmoid 作为激活函数，随机梯度下降(SGD)作为收敛函数，二元交叉熵(BCE)作为损失函数，尽可能准确地重建体素的可用性。该解码器具有对称结构，下采样被上采样替换为 2 倍。

编码器和解码器的训练是同时进行的。使用来自 MPEG 数据集的点云进行训练和测试。以点云库(PCL)编解码器为基准，评估该方案的性能。以重构点与原始点的比值作为性能指标。结论是，使用更多的过滤器(N)可以达到预期的更高质量。虽然在低比特率情况下，PCL 的性能优于 PCG-AE，但在中高比特率情况下，PCG-AE 的性能明显优于 PCL。

2.2 隐式和显式量化的 RD 控制

在传统的基于变换的编码方法中，用(1)中定义的 RD 损失函数(更具体地说是拉格朗日乘子 λ)来控制基于 DL 编码方案中的 RD 权衡。

$$Loss = dist + \lambda \times rate \quad (1)$$

其中， $dist$ 是原始 PC 与重构 PC 之间的失真， $rate$ 是量化隐数据的熵。

PCG-AE 的扩展最近被提出来提供灵活的研发控制功能^[19]。典型的 RD 控制可以使用以下任何一种实现：

- 1) 隐式量化，采用图2所示隐熵和重构误差之间的平衡（由 λ 控制）。

2) 显式量化采用参数量化步骤(QS)来获得不同的 RD 点。

尽管隐式量化方法需要为每个所需的 RD 点训练多个 DL 模型，但它的性能明显优于显式量化。文献^[2]中的方案建议使用隐式和显式量化的组合，与纯隐式量化编码相比，这降低了达到多个 RD 点的训练复杂性和内存需求因此，RD 权衡是使用少量训练好的 DL 模型来实现的。

该方案的概述如图 3 所示，其中它使用公式 1 中定义的损失函数进行训练。针对不同 λ 值训练的模型允许我们在不使用显式量化的情况下实现不同的 RD 点。新的 PC 将使用这些训练过的模型进行压缩。

与前一节提到的方法类似，隐变量在量化后进行熵编码。但为了更有效地进行熵编码，使用变分自编码器(VAE)使熵编码模型更加适应当前隐数据。

当图 3 中所示的模型被训练为 $QS = 1$ 时，即使我们显式地指定了所需的量化量，它也对应于隐式量化。采用较高的 QS 值($QS > 1$ 即显式量化)可以实现 RD 控制，使用相同的 DL 模型生成多个 RD 点。标度块通过 QS 对隐数据进行降维，并对其进行进一步的熵编码。在解码器中，*Scaling - 1*块通过相同的 QS 放大隐值。熵模型适应于某一特定 QS 值。

现在，为了得到多个 RD 点，单个训练模型可用于：

- 1) $QS=1$ 隐式量化
- 2) $QS > 1$ 显示量化

另外，由于模型最初训练为 $QS = 1$ ，采用更高的 QS 可能会有一些失真，因此需要进行隐式和显式的平衡。这种情况的产生是因为使用单个训练过的 DL 模型对生成的 RD 点数量有一定的限制。

质量评估：在这种方法中，与 PCG-AE 不同，PC 被分成大小为 $64 \times 64 \times 64$ 的块。 QS 取值范围为 1 ~ 20。对于纯隐式方案， λ 从 50 变化到 20000。可以观察到，单个模型在高比特率训练时无法达到较低的比特率(QS 增加)，在低比特率训练时无法达到较高的比特率(QS 减少)。因此，隐式-显式量化方案不能像多个经训练的 DL 模型那样适用于大范围的质量和比特率。因此，为了解决这个问题，使用了三个模型而不是一个模型来覆盖广泛的比特率。使用相当少的模型($\lambda = 100, 1000, 10000$)进行隐式 - 显式量化可以达到与使用 10 个训练模型 ($\lambda = 100, 250, 500, 750, 1000, 2000, 3000, 10000, 20000$)进行隐式量化相似的性能，在某些

情况下甚至略好。

2.3 深度学习 (LCT)

LCT 方法采用三维卷积自适应编码器, 直接对体素进行运算。解码可以理解为一个分类问题, 通过将几何信息定义为跨越体素网格的二进制信号来预测、推断体素的占用情况。该方法的体系结构如图 4 所示。

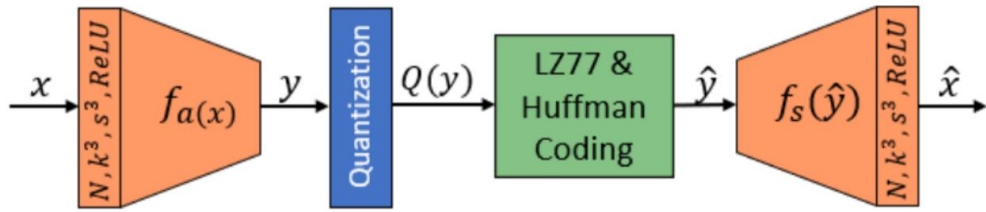


图 4 LCT 架构

其中, N 表示滤波器个数, k 表示滤波器大小, s 表示步幅, $f(x)$ 、 $f(y)$ 表示函数。

隐值采用 $y = f_a(x)$ 表示, 再用均匀量化进一步量化得到 $\hat{y} = Q(y)$, 然后解压缩得到 $\hat{x} = f_s(y)$ 。 f_a 和 f_s 是在训练过程学习的。编码器和解码器分别使用卷积和转置卷积, 两者内容一致, 步长为 2^3 。第一层 k 值为 9, 没有函数的第二层和最后一层 k 值为 5。

该方案没有使用任何熵编码器, 而是使用 LZ77 和 Huffman 编码的组合来压缩 $Q(y)$ 。虽然解码被认为是一个分类问题, 但由于点云的稀疏性, 在被占据体素和空体素之间产生了不平衡。这个问题采用 α 平衡焦损失解决。但最终损失与公式(1)中定义的相同。

结果表明, 尽管使用 ModelNet40 网格数据集(覆盖广泛的种类和数量)进行训练, 并且在完全不同的数据集(MVUB-Microsoft Voxelized Upper Bodies 数据集^[4])上进行测试, 该模型也能够很好地泛化。所提出的方案在所有序列的所有比特率下都优于 MPEG 的 anchor。在比特率相似的情况下, 与 MPEG 的 anchor 相比, 它实现了更低的失真和更多的重构点, 从而获得了更好的质量。不同的 λ 值(10^{-4} , 5×10^{-5} , 10^{-5} , 5×10^{-6} , 10^{-6})用于生成 MPEG 的 anchor 的 RD 点和不同的八叉树深度。测试数据的比特率(BDBR)平均提高了 51.5%。

2. 基于深度学习的点云几何压缩

在 LCT 方法之后,文献中出现了其他新的方法。其中 Learned PCGC 是一种基于 DNN 的变分自适应编码器。方案概述如图 5 所示。

预处理: 该模块包括三个步骤:体素化、缩放和分区。体素化是可选步骤,如果 PC 已经是由 3D 体积表示,则跳过此步骤。使用笛卡尔坐标系进行体素化,如果体素被占用,则设置为 1,否则设置为 0。体素间相关性可以通过 3D 卷积来生成非常紧凑的 3D 块^[21]的隐式表达式。

与图像和视频压缩中使用的图像下采样类似, PCC 中也使用了下采样,以便在较低的比特率下保持质量。进行缩放以减少稀疏性,从而增加 PC 的密度。研究发现,这样做大大提高了效率,特别是对于稀疏 PC,如动态获取的 MPEG 数据集点云(类别 3)。具体操作如下:

$$\hat{X}_n = \text{round}(X_n \times s) \quad (2)$$

其中, $(X_n \times s) = (i_n \times s, j_n \times s, k_n \times s)$, X_n 表示原始点云几何坐标, $n = 1, 2, \dots, N$ 。 X_n 按 s 缩放, $s < 1$ 。然后将 PC 划分为大小为 $W \times W \times W$ 的 3D 不重叠立方体,被占据的立方体的各自位置与每个立方体中被占据的体素被标识,从而增加了非常少的元数据量。

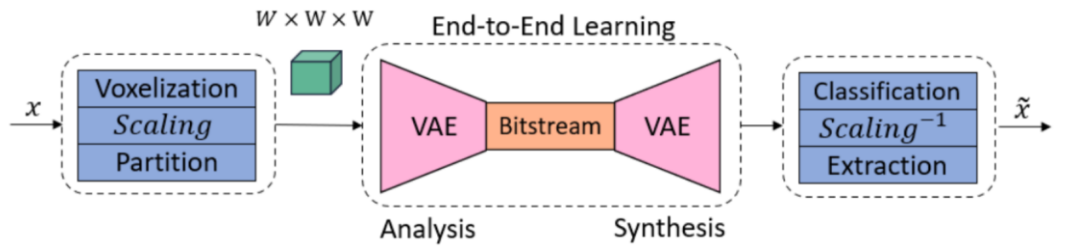


图 5 PCGC 的深度学习概述

3D 立方体被一个接一个地馈送到由 3D 块堆叠 CNN 组成的分析变换中,以生成尺寸(通道,长度,宽度,高度)紧凑的隐式表达,定义为 $y = f_e(x; \theta_e)$, 其中 θ_e 是卷积权值的集合。对于解码端,综合变换是指将量化隐式值 \hat{y} 用于重建立方体,公式为 $\tilde{x} = f_d(\hat{y}; \phi_d)$, 其中 ϕ_d 为参数集。

在变换中使用的基本三维卷积单元是 Voxelnet - Resnet (VRN), 因为它的效率是由剩余和初始网络^[25]获得的。

该方案使用相当小的内核($3 \times 3 \times 3$), 有着较低的复杂性。类似于 LCT 方法,

超先验概率用于熵建模。

后处理：它包括分类、逆缩放和提取。尽管阈值的中值为 0.5，但它被证明不是最优的，因此，分类步骤使用自适应阈值来推断重建立方体 \tilde{x} 中的体素是否存在。立方体中体素的分类还受到原始立方体中已占用点的数量的影响。该信息在编码期间作为元数据发送。逆缩放在下一步实现，它将重构几何图形以 $\frac{1}{s}$ 的速度进行渲染和显示。提取是一个可选步骤，它将体积表示更改为原始文件格式，如多边形文件格式(.ply)或 ASCII。

为了进行模型泛化，该方案使用 ShapeNet 作为训练数据集，用 MPEG 和 JPEG 的 pleno 标准化数据集进行测试。通过改变 λ (拉格朗日)和缩放因子 s 对每个比特率进行端到端训练。训练时使用加权二元交叉熵损失(WBCE)，而推理时使用自适应阈值。仿真结果表明，基于深度学习的 PCGC 比 GPCC(八叉树)的性能分别高出 77% 和 69%，比 GPCC(Trisoup)的性能分别高出 67%和 62%，在 D1(点对点)和 D2(点对面)上分别比 PCL 的性能提高 88%和 82%。

第三章 全连接神经网络 (FCNN)

CNN 不仅保留了图像和视频中的空间关系，而且具有权值共享的优势，大大降低了 ML 解决方案的复杂性。因此，CNN 在实际应用中已经成为解决基于视觉的任务的最常用选择，特别是在 PC 和网格^[4]等高级可视化数据表示中。

然而，很少有方法直接工作在 PC 的几何坐标上实现压缩。本章将讨论其中两种方案，并与之前总结的基于 3D CNN 的 AE 解决方案进行比较。

3.1 基于深度 AE 的 PCGC

本方案的重点是设计一种适合 PC 机特点的编解码器，提高压缩效率。所提出的网络结构如图 6 所示。它由四个主要模块组成:作为分析变换^[26]的基于点网的编码器，统一量化器，中间的熵编码块，以及作为非线性综合变换的解码器。PC 的几何图形表示为由 n 个点组成的 3D 点。首先对这 n 个点进行下采样，以降低原始 PC 的稀疏性。采样层将产生重复的点，这些点合并在一起，得到更少的点(m)，这些点是原始采样 n 个点^[22]的子集。

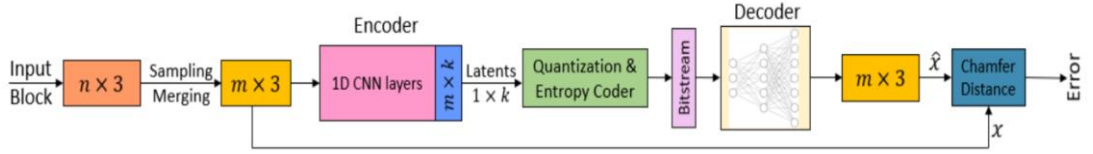


图 6 深度 AE 架构

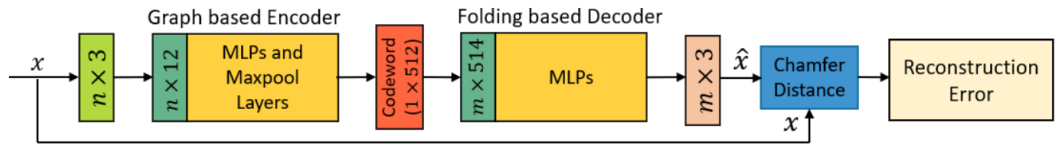


图 7 FoldingNet 体系结构

接下来，将 m 个点送入编码器，编码器由 5 个一维卷积层组成，核大小为 1，然后是一个 ReLU 和批量归一化层。每一层的过滤器分别是 64、128、128、256 和 k 。这里 k 是由输入点的数量决定的。然后利用最大池化层生成 k 维潜伏码 z ，利用熵编码器对潜伏码进行量化编码。

译码器采用 FCN，利用 3 个全连接层对隐码 z 进行解码，隐层为 256、256 和 $n \times 3$ 个神经元，生成 $n \times 3$ 个重构 PC。整个过程可以用下式概况：

$$z = D \left(Q \left(E(S(x)) \right) \right) \quad (3)$$

其中 S, E, Q 和 D 分别为采样层、编码器、量化和解码器。分析和综合变换公式为

$z = f_e(x; \theta_e)$ 和 $\hat{x} = f_d(y; \phi_d)$, 其中 x 为原始 PC, z 为压缩函数, \hat{x} 为重构 PC。 θ_e 和 ϕ_d 为可训练参数。

量化为达到较小的损失用到了四舍五入的操作函数, 使得求导后数据为零或者未定义。因此, 所提出的方案用加性均匀噪声代替量化。

$$[f(x)] \approx f(x) + \mu \quad (4)$$

其中 μ 是随机噪声。畸变计算使用倒角距离定义如下

$$d_{CH}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2 \quad (5)$$

其中 S_1 是 n 个点的原始 PC, S_2 是 m 个点的重构 PC。所提出的方案使用 ShapeNet 作为单个类的训练集和测试集, 分成 90% 和 10% 两部分。该项测试结果与 MPEG 的 anchor 作对比。该模型使用了四个类别的数据: 椅子(火车:6101, 测试:678)、飞机(火车:3640, 测试:405)、桌子(火车:7658, 测试:851)和汽车(火车:6747, 测试:750)。基于深度 AE 的 PCGC 比 MPEG GPCC 在 BD 率上平均获得 73.15% 的增益。

3.2 FoldingNet: PC 与 AE 之间的深度网络变换

一个 CNN 需要它的邻居在一定距离范围内, 以保证邻居共享固定的空间关系。由于 PC 在原始格式中没有这样的特征, 因此对 PC 进行体素化, 使 PC 上的卷积运算更有意义。但是体素化牺牲了 PC 原始表示的细节。在没有体素化的情况下, 很少有方法可以在原始 PC 上工作。其中之一就是最近提出的 FoldingNet。它是一个基于 MLP 的 AE, 利用二维网格结构通过折叠操作^[23]构造 PC。

图 7 显示了 FoldingNet 体系结构。编码器由 MLP 层和基于图的最大池化层组成。PC 的几何形状 $n \times 3$ (其中 n 是点的数量) 是编码器的输入。然后使用坐标计算每个点 v 大小为 3×3 的局部协方差矩阵, 该坐标是 v 的一跳邻居, 并向量化到 1×9 。协方差 ($n \times 9$) 和几何 ($n \times 3$) 连接到大小为 $n \times 12$ 的矩阵, 并馈送到 3 层感知器。感知器的输出被馈送到两个连续的图层, 最终生成一个码字 (1×512), 它映射输入 PC 的高维嵌入。将码字复制 m 次, 并与大小为 $m \times 2$ 的二维固定网格点连接, 生成 $m \times 514$ 的矩阵。该矩阵是 3 层感知器的输入, 生成第一次折叠操作 ($m \times 3$) 的输出, 然后再与复制码字连接, 成为第二次折叠操作的输入, 从而得到重构 PC。两次折叠操作可以被理解为 2D 到 3D 的转换/映射, 从而产生精细的表面。第一个折叠 2D 网格到 3D 空间, 第二个折叠在 3D 空间内部。输出点的数量 m 不一定与 n 相同。输入和重构 PC 之间的失真误差是使用上一节定义的倒角距离

来计算的

在训练过程中，模型从随机折叠开始，最终变成一个有意义的 PC。利用 ShapeNet 数据集的 16 个类别对模型进行训练。如图 7 所示，基于折叠的解码器的一个主要优点是它只使用了解码器 7% 的参数，与 FCNN 解码器相比，它更专注于解决分类问题，而不是压缩 PC。然而，这一想法可以适用于 PCC，以验证与国际标准化相比的性能。在考虑分类问题时，使用编码器末端生成的码字进行 SVM 分类。

将基于折叠的解码器与 FC 解码器进行了比较。在训练过程中，重构损失(倒角距离)不断减小，表明重构后的 PC 越来越接近原始 PC。对码字进行训练后的线性 SVM 分类准确率有所提高，这意味着码字在训练过程中变得更加线性可分。为了验证，使用 ShapeNet 训练模型进行迁移学习，并在 ModelNet 数据集上进行测试，实现了 88.4% 的分类准确率。

第四章 循环神经网络

我们使用统一的标量量化器对变换系数进行量化，然后用一个算术编码器对每个 AC 量化系数进行熵编码[21]。由拉普拉斯分布驱动，其参数对每个子波段都是唯一的，AC 系数被编码并从编码器传输到解码器。

使用 RNN 进行点云压缩的 ML 方法非常有限。本节将讨论其中一种，重点是来自 3D 激光雷达传感器的数据。作为标准化的一部分，传统的方法在将点云减小到非常小的体素方面并不是很成功，特别是 3D 激光雷达序列。因此，一些方法对来自 3D 激光雷达的原始数据包数据进行操作的方法被证明是非常经济有效的压缩方法。这种方法虽然可以无损地将原始数据包数据转换为 2D 矩阵，但如果不经校准，它们在本质上非常不规则，并且不共享任何空间相关性。然而，这样做的好处是减少了 PC 的稀疏性，有助于进一步处理^[24]。

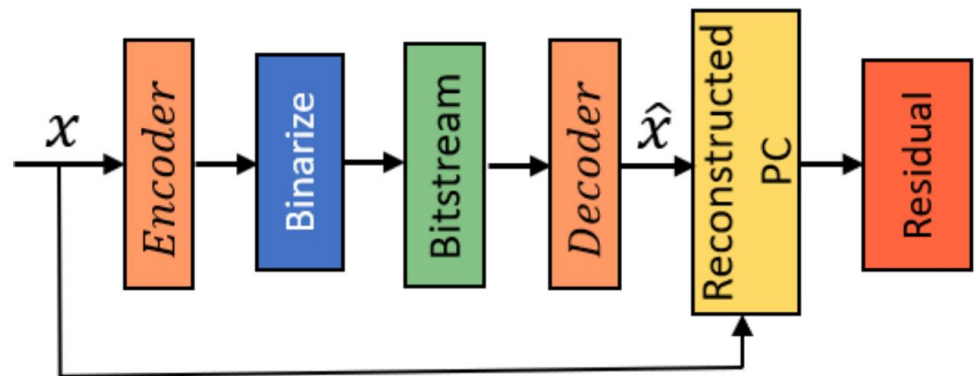


图 8 残差计算

该方案针对静态 PC，引入了一种基于 RNN 的残差块压缩方法。首先，通过转换原始数据包数据的帧来构造二维矩阵。然后，在数据被馈送到由编码器、二进制器和解码器组成的网络之前，执行二维矩阵归一化作为预处理步骤。由于激光雷达检测扫描仪有更高精度和更远量程，2D 格式可以有更大的可能值范围，该方案使用几个步骤进行归一化：数据包是从 100 帧数据中随机选择的，用此来计算平均值 μ 并构建距离值的直方图。然后对数据矩阵 R 归一化如下： $(R - \mu)/\theta$ ，其中阈值 θ 设置为使用 95% 的距离低于此值。

图 8 中的编码器、二进制器和解码器分别由卷积层和卷积 LSTM 层的组合组成。解码器也会使用到剩余块，因为在使用这样的网络时，获得准确的解压缩可能

具有挑战性。具体来说，解码器迭代地使用传入的二进制文件重构原始输入，并将重构数据与原始 PC 机进行比较来计算残差。这个残差成为下一次迭代的输入。重复这个过程，直到解压缩的数据相当准确为止。产生这种残留物的过程也如图 8 所示。解压损失计算使用 SNNRMSE(对称邻居均方根误差)^[24]。该方案在日本 11 个地区的驾驶数据上进行了测试，结果表明，该方案优于八叉树压缩和 JPEG 压缩标准的方法。

第五章 基于 AI 的 PC 传输

在资源受限的信道上传输 PC 视频数据是一个挑战。此外，利用现有网络的带宽实时传输 PC 视频数据也很困难。因此，PC 数据压缩方案的可用性对于实现这样的 PC 视频流应用是非常有用的。

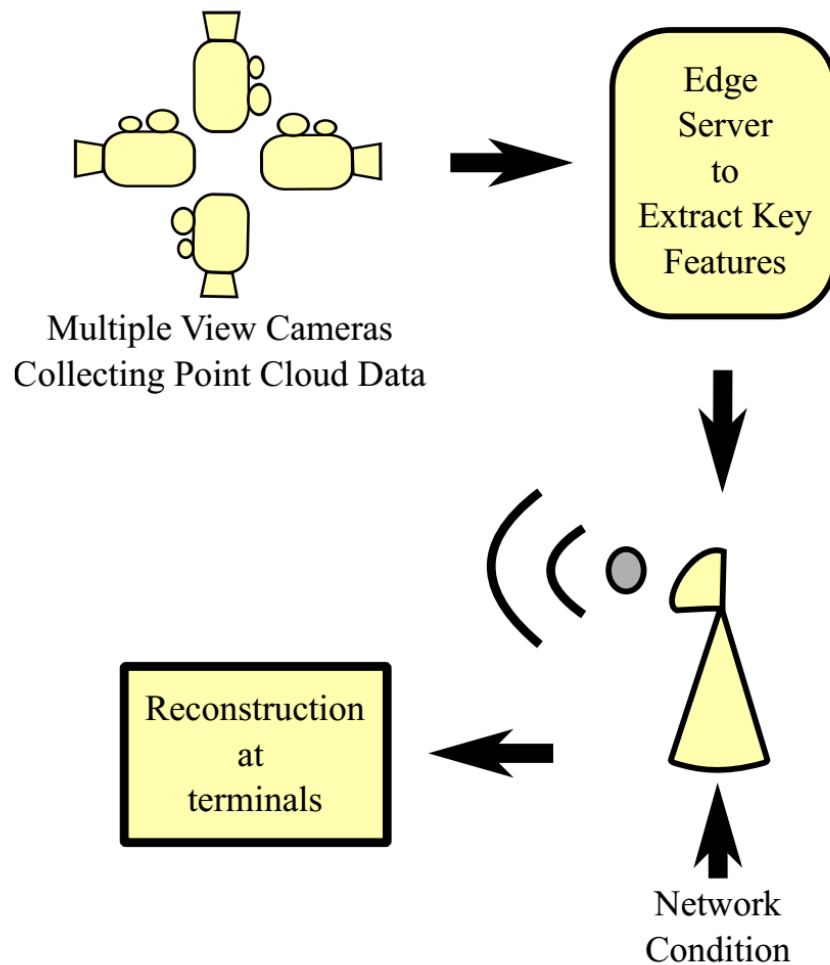


图 9 AItransfer 系统的组成部分

如前所述，使用 AI/ML 来协助压缩 PC 数据产生了有希望的结果。同样，使用 AI/ML 辅助视频点云数据的传输将在本节讨论。资料显示^[14]，一种名为 AItransfer 的 AI 启用技术已被提出。该方法使用人工智能驱动的自适应传输技术，该技术具有带宽感知能力。通过从 PC 机上提取和传输关键特征，可以减少带宽消耗和减轻计算量。因此，使用 AItransfer 技术的好处之一是在端到端架构设计中使用特征提取和重建等基本概念，将动态网络带宽纳入其中。AItransfer 的另一个贡献是使用在线适配器来感知网络带宽，以匹配最优推理模型。

图 9 显示了 Altransfer 的体系结构，突出显示了它的组成部分和系统流程。如图 9 所示，该技术利用基于人工智能的神经网络，从多视角摄像机采集的 PC 数据中提取关键特征。提取这些特征可以压缩 PC 数据，使其与网络带宽处理能力更加兼容。然后利用现有网络传输特征提取的数据，并在预期的终端上进行重构。特征提取步骤如图 10 所示。

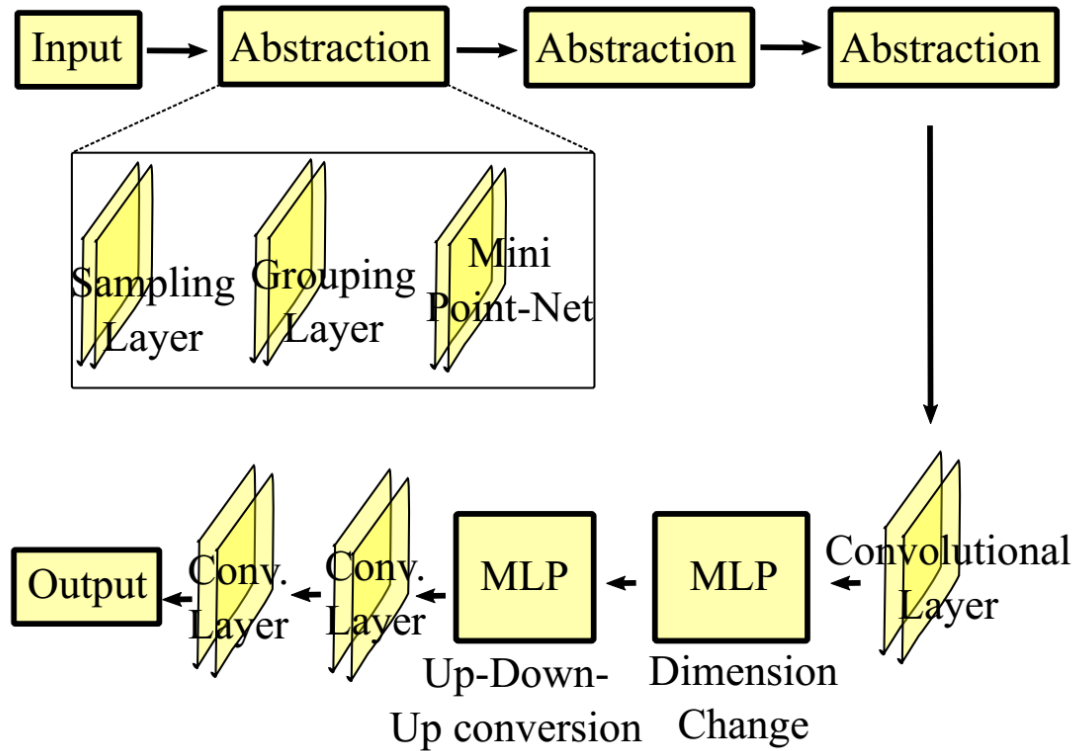


图 10 网络架构的设计

从图 10 可以看出，基于人工智能的网络中用于提取关键特征的各个层都有几个抽象步骤，然后是卷积层、维数变化和更多的卷积层，从而产生压缩的 PC 输出。这些数据再根据网络条件进行传输，然后在手机、VR 头显等终端上进行重构。文献[14]通过一个案例来评估 Altransfer 的性能，结果表明 Altransfer 的压缩比几乎可以达到 30 比 1 以上。

第六章 总结与展望

本文对使用深度学习、机器学习技术的点云压缩(PCC)这一新兴研究领域进行了综述。由于这方面的发展处于早期阶段,本篇综述仅总结了 8 篇研究文章。除了对这一领域的最新发展进行总结外,我们还对所调查的技术进行了比较性能研究。

文献中提出的使用 DL/ML 的 PCC 技术大多与几何压缩有关,很少有技术关注 PC 和视频压缩的属性压缩。我们还看到,大多数方法使用基于卷积的自动编码器、多层感知器和完全连接的神经网络解码器来实现 PCC。最近,使用基于 RNN 的技术来执行 PC 压缩也被研究,这种方法也包括在本次调查中。

PC 数据的传输和压缩 PC 都具有挑战性,这使其成为了一个新兴的领域。本文还简要讨论了一种最新的基于 AI 的压缩和传输技术,即 AItransfer。

参考文献

- [1] C. Cao, M. Preda, and T. Zaharia, “3d point cloud compression: A survey,” 07 2019, pp. 1–9.
- [2] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, “An overview of ongoing point cloud compression standardization activities: video-based (v-pcc) and geometry-based (g-pcc),” *APSIPA Transactions on Signal and Information Processing*, vol. 9, p. e13, 2020.
- [3] D. T. Nguyen, M. Quach, G. Valenzise, and P. Duhamel, “Learningbased lossless compression of 3d point cloud geometry,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toronto, Canada, Jun. 2021. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-03141577>
- [4] C. Loop, Q. Cai, S. O. Escolano, and P. A. Chou, “Microsoft Voxelized Upper Bodies – A Voxelized Point Cloud Dataset,” *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document m38673/M72012*, May 2016. [Online]. Available: <http://plenodb.jpeg.org/pc/microsoft/>
- [5] R. Hooda and W. D. Pan, “Early termination of dyadic region-adaptive hierarchical transform for efficient attribute compression of 3d point clouds.” *IEEE Signal Processing Letters*, 2021.
- [6] L. Wiesmann, A. Milioto, X. Chen, C. Stachniss, and J. Behley, “Deep compression for dense point cloud maps,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2060–2067, 2021.
- [7] R. Hooda and W. D. Pan, “Tree based search algorithm for binary image compression,” in *2019 SoutheastCon*, 2019, pp. 1–6.
- [8] Z. Liu, Q. Li, X. Chen, C. Wu, S. Ishihara, J. Li, and Y. Ji, “Point cloud video streaming: Challenges and solutions,” *IEEE Network*, vol. 35, no. 5, pp. 202–209, 2021.
- [9] R. Hooda and W. D. Pan, “Lossless compression of bilevel roi maps of hyperspectral images using optimization algorithms,” *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.
- [10] M. Quach, G. Valenzise, and F. Dufaux, “Improved deep point cloud geometry compression,”

2020.

- [11] X. Wen, X. Wang, J. Hou, L. Ma, Y. Zhou, and J. Jiang, “Lossy geometry compression of 3d point cloud data via an adaptive octreeguided network,” in 2020 IEEE International Conference on Multimedia and Expo (ICME), 2020, pp. 1–6.
- [12] P. A. Chou, M. Koroteev, and M. Krivokuca, “A volumetric approach to ‘ point cloud compression—part i: Attribute compression,” IEEE Transactions on Image Processing, vol. 29, pp. 2203–2216, 2020.
- [13] T. Chen, H. Liu, Q. Shen, T. Yue, X. Cao, and Z. Ma, “Deepcoder: A deep neural network based video compression,” in 2017 IEEE Visual Communications and Image Processing (VCIP), 2017, pp. 1–4.
- [14] Y. Huang, Y. Zhu, X. Qiao, Z. Tan, and B. Bai, “Aitransfer: Progressive ai-powered transmission for real-time point cloud video streaming,” in Proc. of the 29th ACM Int. Conf. on Multimedia. New York, NY, USA: Association for Computing Machinery, 2021, p. 3989–3997. [Online]. Available: <https://doi.org/10.1145/3474085.3475624>
- [15] W. Jia, L. Li, A. Akhtar, Z. Li, and S. Liu, “Convolutional neural network-based occupancy map accuracy improvement for video-based point cloud compression,” IEEE Transactions on Multimedia, pp. 1–1, 2021.
- [16] E. Alexiou, K. Tung, and T. Ebrahimi, “Towards neural network approaches for point cloud compression,” in Applications of Digital Image Processing XLIII, A. G. Tescher and T. Ebrahimi, Eds., vol. 11510, International Society for Optics and Photonics. SPIE, 2020, pp. 18 – 37. [Online]. Available: <https://doi.org/10.1117/12.2569115>
- [17] A. F. R. Guarda, N. M. M. Rodrigues, and F. Pereira, “Adaptive deep learning-based point cloud geometry coding,” IEEE Journal of Selected Topics in Signal Processing, vol. 15, no. 2, pp. 415–430, 2021.
- [18] A. F. R. Guarda, N. M. M. Rodrigues, and F. Pereira, “Point cloud coding: Adopting a deep learning-based approach,” in 2019 Picture Coding Symposium (PCS), 2019, pp. 1–5.
- [19] A. F. R. Guarda, N. M. M. Rodrigues, and F. Pereira, “Deep learningbased point cloud geometry coding: Rd control through implicit and explicit quantization,” in 2020 IEEE International Conference on Multimedia Expo Workshops (ICMEW), 2020, pp. 1–6.
- [20] M. Quach, G. Valenzise, and F. Dufaux, “Learning convolutional transforms for lossy point

- cloud geometry compression,” CoRR, vol. abs/1903.08548, 2019. [Online]. Available: <http://arxiv.org/abs/1903.08548>
- [21] J. Wang, H. Zhu, Z. Ma, T. Chen, H. Liu, and Q. Shen, “Learned point cloud geometry compression,” 2019. [Online]. Available: <https://arxiv.org/abs/1909.12037>
- [22] W. Yan, Y. Shao, S. Liu, T. H. Li, Z. Li, and G. Li, “Deep autoencoder-based lossy geometry compression for point clouds,” CoRR, vol. abs/1905.03691, 2019. [Online]. Available: <http://arxiv.org/abs/1905.03691>
- [23] Y. Yang, C. Feng, Y. Shen, and D. Tian, “Foldingnet: Interpretable unsupervised learning on 3d point clouds,” CoRR, vol. abs/1712.07262, 2017. [Online]. Available: <http://arxiv.org/abs/1712.07262>
- [24] C. Tu, E. Takeuchi, A. Carballo, and K. Takeda, “Point cloud compression for 3d lidar sensor using recurrent neural network with residual blocks,” in 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 3274–3280.
- [25] Y. Zhou and O. Tuzel, “Voxelnet: End-to-end learning for point cloud based 3d object detection,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4490–4499, 2018.
- [26] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 77–85.

