南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

计算机科学与工程系
Department of Computer Science and Engineering

计算机科学与工程系
CS326-创新实践 II

基于八叉树的 3D 点云压缩实现

作者: 叶成伟 12010524 王宇航 12012208 陈张杰 12012524
指导教师: 于仕琪
时间:2023/5/2

I. 项目背景

在人类历史上的长久岁月中，我们一直生活在一个三维世界中。但由于技术的限制，人们只能在二维平面上绘制现实世界的投影，从岩壁、帆布、纸张、到胶片。这种方式只能呈现物体的表面信息，而缺乏对其立体形态和空间结构的精准表达。然而，随着数字化时代的到来和 3D 技术的迅猛发展，人们终于能够在真正的三维语言体系下记录、处理甚至呈现 3D 物体。这一技术突破使得我们能够更加准确地模拟和理解真实世界中的物体，更加便捷地进行工程设计、医学手术规划等活动，也使得机器人和自动驾驶系统更加智能和自适应。

目前，点云是一种流行的描述 3D 数据的结构。然而，点云往往承载着巨量几何信息与特征，想让一个动态点云达到视觉上平滑流畅的清晰度和帧率，需要每帧约 100 万个点，帧率达到 30fps，占据约 500Mbps 到 1Gbps 带宽 [1] [2]。可以预见，如果不对 3D 点云进行体积上的压缩，数据传输将承受巨大的带宽开销。这其中一个典型的例子就是自动驾驶。自动驾驶领域中广泛使用 3D 点云来刻画车辆周围环境，但未经压缩的原始点云数据存储和传输面临着极大的困难。如果不对原始点云数据进行压缩，点云处理相关算法难以真正落地。因此,3D 点云的压缩成为了一个新兴的研究方向。2020 年,Motion Picture Experts Group(MPEG) 推出了两个 3D 点云压缩标准，其中 Video-based Point Cloud Compression(V-PCC) 主要基于已成熟的图像与视频压缩技术,Geometry based Point Cloud Compression (G-PCC) 则是直接根据其三维几何信息进行压缩。本项目拟采用 G-PCC 标准，进行点云的压缩与解压缩。

	queen	longdress	loot	redandblack	soldier
Average number of points (in 300 frames)	1,005,000	834,000	794,000	727,000	1,076,000
Bitrates for transmitting uncompressed video (Mbytes/s)	514.47	542.22	490.61	448.21	681.96

图 1: 带宽估计

From Cao C, Preda M, Zaharia T. (2019). 3D Point Cloud Compression: A Survey. 1-9. 10.1145/3329714.3338130.

II. 研究情况

本项目的研究内容为: 在 OpenCV 5.x 分支中, 基于八叉树实现可选分辨率的点云数据压缩算法。本项目程序接收 3D 点云数据作为输入, 构建八叉树用于承载点云数据。接着，采用多种不同的方式，如直接编码，预测编码，熵编码对八叉树进行编码，最终输出一串比特流作为压缩结果。与此同时，该算法支持点云的颜色压缩。针对上述的压缩方案，提供对应的解压缩方案。

本项目延续创新实践 I 的成果，针对创新实践 I 已有的编码方式熵编码进行改进，提供更加灵活的编码方法，以应对不同编码场合，通过多种编码方法的结合，达到较为理想的压缩率。项目团队已确定改进方向主要包括直接编码，预测编码，颜色编码，帧间编码。目前，本项目的重点将放在直接编码与预测编码，颜色编码这两个模块的实现上。下面将分别介绍。

A. 直接编码与预测编码

在 G-PCC 中，直接编码模式（DCM）用于压缩八叉树深度较深时的节点数据，其中对于一些孤立的点，直接编码可以提高压缩效率。而预测编码，则是利用了点云邻域信息，发掘点云之间的空间相关性，压缩冗余的空间信息。

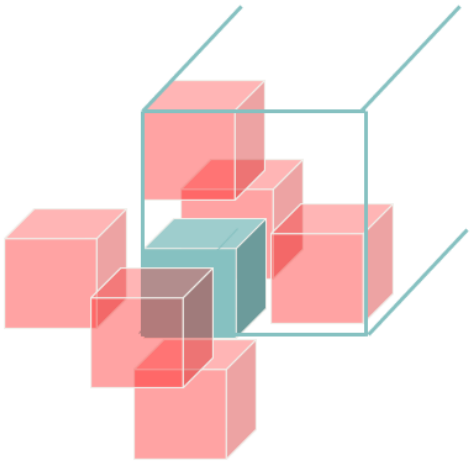

图 2: 孤立点判断准则

受卷积和 MPEG 论文的启发，我们认为：三维点云类比二维图像，均具有空间局部冗余。因此，如图 2 所示，我们选用 6N 邻域（六个相邻方块空间）作为预测的立足点和信息来源。与此同时，6N 判断同样适用于直接编码，通过 6N 判断查看当前节点物理空间上相邻的 6 个节点是否有被占用，若均未被占用，且该节点包含的点云数据量为一，则该点云为孤立点，进入直接编码模式。

实现 6N 判断是上述两种压缩方法的关键环节，针对 6N 判断，本项目定义了一种映射关系，记录了每个节点相邻的 6 个同级节点，具体实现方法如图 3 所示：针对每个节点，本项目维护了它的父亲节点与所有孩子节点的指针，通过这些指针，在构建八叉树时，通过父亲节点的 6N 信息可以推算出本节点的 6N 信息。

针对预测编码，本项目维护了一张加和表与权重表，具体如图 4 所示，以此完成预测编码的主要部分

B. 颜色编码

在图像压缩领域中，Haar 小波变换是一种常用的数学技术，它通过将图像分解成一系列小波系数来捕捉图像的高频成分，可以有效减小数字图像的大小，同时保留数字图像的基本特征。

RAHT(Region-Adaptive Hierarchical Transform, 区域自适应层次变换) 是一种层次子带变换，也是 Haar 小波变换在 3D 点云上的实现。它的思路是根据八叉树中低层数节点相关的颜色来预测下一层节点的颜色。RAHT

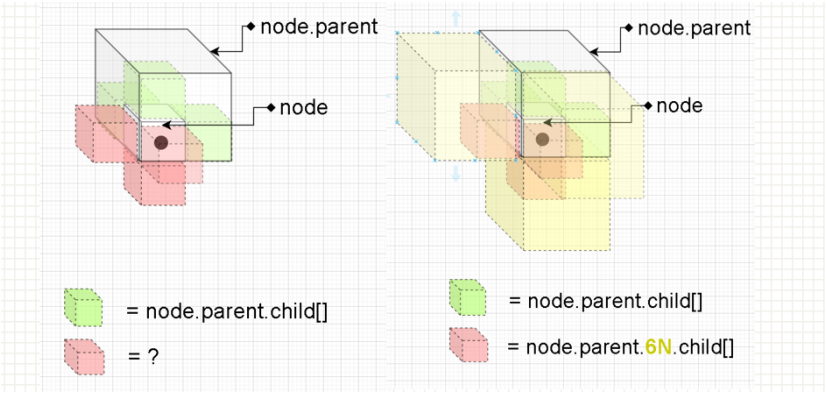

图 3: 6N 方法实现

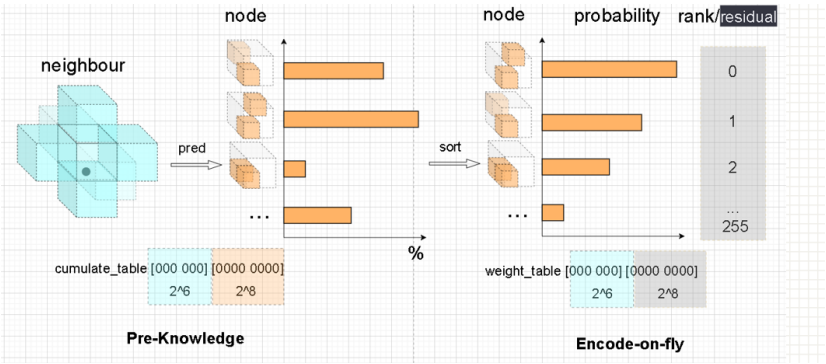

图 4: 预测编码实现

算法对八叉树进行反向构建，每一步都将同一级别的两个体素组成为更大的体素，直到到达根节点，每一次都会产生 RAHT 系数。之后，对 RAHT 变换系数进行量化，然后使用算术编码器 (AC) 对系数进行熵编码达到压缩效果。

颜色编码的实现分为两个阶段，第一个阶段：针对八叉树结构，从叶子节点层开始，逐层合并节点的颜色信息，并记录相应的系数，最终得到合并后的根节点和系数数组。这一组数据描述了整棵八叉树的颜色信息，并挖掘了其隐藏的层次关系和位置信息，针对这组数据进行熵编码，可获得较好的效果。第二阶段，实现对系数数组的编码，将其转化为字节流交由熵编码模块。

针对系数数组的构建，本项目采用如下方法：本项目采用八叉树作为点云数据的承载结构，因此，采用回溯的方式能够十分完美的解决从叶子节点逐层合并这个需求。我们采用了类似于二叉树后序遍历的方式，访问父亲节点前，需要访问全部的孩子节点，这样，所有孩子节点合并所得的信息恰好代表了父亲节点的信息，并继续用于合并。同时，维护一个系数数组记录所得系数。

针对系数数组的编码，本项目拟采用如下方法：

由于得到的系数数组元素为整数，而熵编码接收的输入是一串字节数组，故需要将整数用字节表示出来。值得注意的是，需要对系数为负数的元素做一个转化。因为对于整数存储，计算机用最高位表示符号位，绝对值较小的负数会有大量的前导 1，如果直接存储，则将导致字节数组存在大量的八位比特位均为 1 的字节。与此同时，较小的正数会导致字节数组存在大量的八位比特全 0 的字节。在颜色编码阶段，我们发现，若不对负数进行转化，则字节数组中将存在大量且比例几乎持平的全 0 字节和全 1 字节，对熵编码效果产生巨大影响，故需要将负数映射为正数。

具体方法为：对于数组中的正数，全部采取左移一位的操作，保证其二进制形式末尾为 0；对于数组中的负

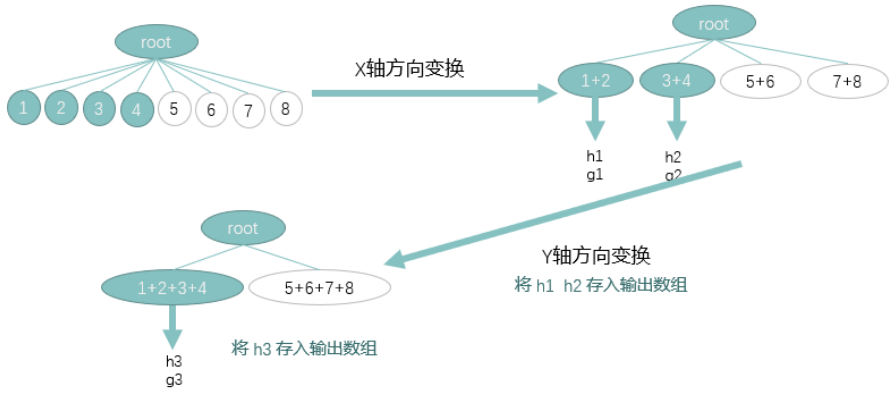

图 5: 颜色编码系数数组构建

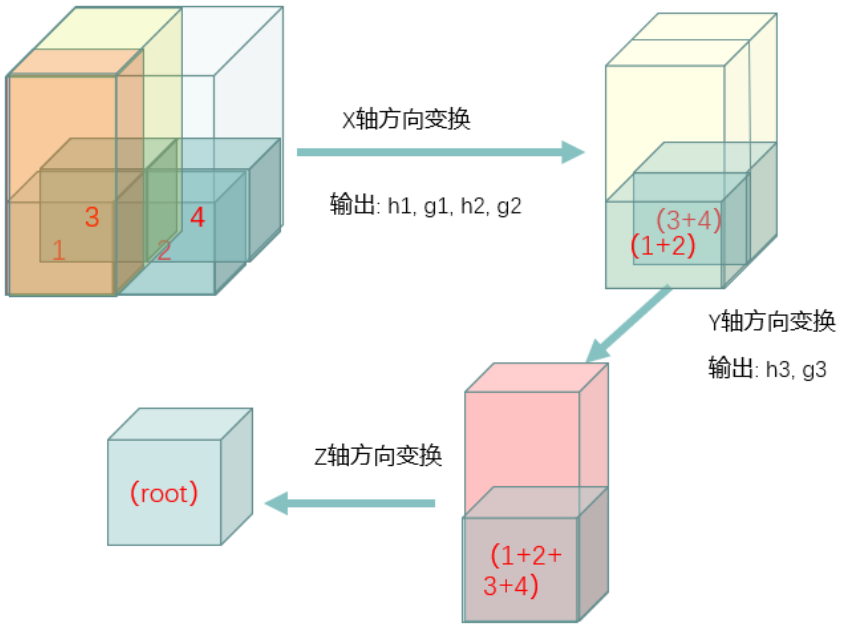

图 6: 颜色编码系数数组构建

数，先采取补码转换为正数，再采取左移一位的操作，对齐减一保证其二进制形式末尾为 1。

III. 阶段成果

目前本项目已完成的任务有：针对颜色压缩的可行性进行分析，完成颜色压缩系数数组的计算以及编码。完成直接编码，预测编码两种编码方式。针对以上所完成的工作，进行以下总结：

1) 颜色编码效果： 不难发现，颜色编解码的流程是完善的，还原出的点云颜色宏观上与原点云几乎一致。颜色编码的正确性得到了保障。颜色编码中对于产生的 RAHT 系数，采用不同的量化步长对系数进行量化处理，编码效果也有所区别。图 7 是原点云与量化步长为 10 和 50 的对比。当量化步长较大时，由于相近颜色的差值较小，在量化一步中被忽略，因而还原出的点云中，相近点的颜色会趋于相同，产生了图中色块的结果。当量化步

图 7: 不同 stepsize 下颜色编码效果

长较小时，相近颜色的差值也会被保留，因此还原出的点云颜色有更好的细节，和原文件差距较小。然而，当量化步长较大时，点云颜色数据在量化后更有规律性，因此压缩后占用空间更少。

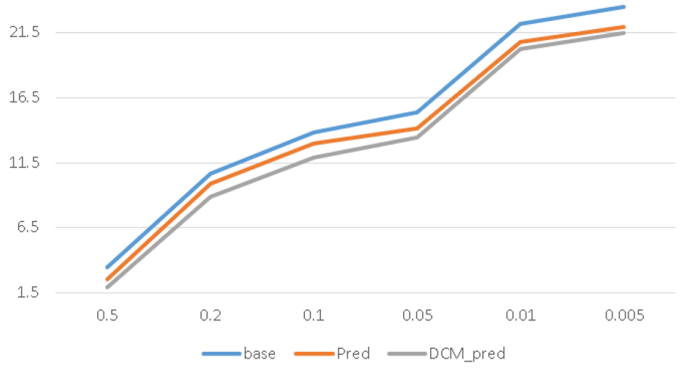

图 8: 不同 stepsize 下颜色编码效果

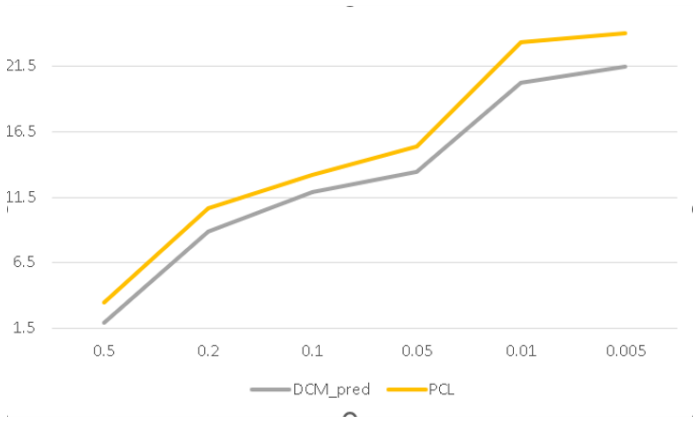

图 9: 不同 stepsize 下颜色编码效果

2) 直接编码与预测编码效果: 如图 8，图 9 所示，我们在不同精度下，分别测试了直接编码与预测编码后的压缩率，并与知名的点云处理库 PCL 进行横向对比。本项目采用 bpp(bit per point) 作为压缩率的评价指标，可以看出，直接编码与预测编码均对压缩率有着明显提高，结合两种编码方式后，在相同精度下的压缩率优于 PCL。

IV. 后续计划

1) 逐步提交已有代码，以便迅速反馈: 本项目成功申请到了 OpenCV 社区的 GsOC2023 点云压缩项目。针对该项目，我们拟定分期交付成果，以便迅速得到社区的反馈，同时方便审查。

2) 完成帧间编码: 目前工作集中于单帧静态点云，后续引入适用于动态点云的压缩的帧间编码，拟采用双缓冲八叉树结构实现。

3) 优化运行时间: 目前，本项目关注的重点始终在于压缩率的优化，而忽略了运行时间。现阶段，本项目的压缩率已经可以达到令人满意的程度。在完成剩余的编码方法后，我们将重点关注运行时间的优化。经过讨论，拟采用 SIMD 和多线程为主要手段，进行速度的优化。

参考文献

[1] Cao C, Preda M, Zaharia T. (2019). 3D Point Cloud Compression: A Survey. 1-9. 10.1145/3329714.3338130.

[2] Graziosi D, Nakagami O, Kuma S, Zaghetto A, Suzuki T, Tabatabai A. (2020). 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. 9. 10.1017/ATSIP.2020.12.

[3] Inc. 8i Labs. (2017). 8i Voxelized Full Bodies, version 2 –A Voxelized Point Cloud Dataset. ISO/IEC JTC1/SC29/WG11 m40059 ISO/IEC JTC1/SC29/WG1 M74006, Geneva, CH

[4] MPEG 3DG, Call for proposals for point cloud compression v2, ISO/IEC JTC 1/SC 29/WG 11 N16763, 2017.

[5] MPEG 3DG, G-PCC codec description v12, ISO/IEC JTC 1/SC 29/WG 7 N0151, 2021.

[6] 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, 2018:1-1.