



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: 2023/6/11

本项目的研究内容为，在 OpenCV5.x 分支中，基于八叉树实现可选分辨率的点云数据压缩及解压缩算法。同时，添加对属性编码（如颜色）的支持，并通过进一步改善编码技术，如直接编码，预测编码，帧间编码等方法，以充分利用点云数据的空间冗余和时间冗余，通过多种编码方法的结合，达到较为理想的压缩率。同时，需针对算法的效率做出优化，使其能够满足实际场景中的效率需求。

I. 项目意义

三维视觉是计算机视觉的下一阶段。在数字化时代和 3D 技术的快速发展下，三维数据的应用范围日益广泛。其中，三维点云作为三维数据的重要组成部分，具有巨大的发展潜力。

目前，自动驾驶领域是最具潜力的应用领域之一。自动驾驶最核心的技术，在于设计一个系统，使得车辆能够探测，感知周围环境的变化，并做出应答。在探测方面，车载激光雷达 (LiDAR) 因其具有可准确获取目标的三维信息、分辨率高、抗干扰能力强、探测范围广、近全天候工作等优点，在智能驾驶环境感知系统中占据了重要地位。而探测的结果和感知决策算法所处理的数据，往往以点云占主导。

点云数据由空间中的 xyz 坐标构成，可以准确地表示 3D 物体的位置和形状。传感器能够发送相应的信号并接收返回的信号，通过测量信号的时间差或光线的反射等原理，得到物体表面各个点的距离信息，故点云数据易于探测，且可以大范围覆盖。同时，激光雷达能够以较高的精度获取物体表面各个点的位置信息，且不依赖物体表面的纹理信息，不受光照和复杂几何结构的影响。

通过分析点云数据，自动驾驶系统能够精确感知周围环境中的障碍物、道路标记、行人和其他车辆的位置和运动状态、能够精确测量目标的位置、形状及状态，达到探测、识别、跟踪目标的目的，从而实现对周围环境变化的应答功能。现阶段，自动驾驶点云处理算法主要有物体检测，SLAM(simultaneous localization and mapping)，多传感器标定等。这些算法都使用点云数据进行环境信息感知和处理。

然而，3D 点云虽然保留了环境的重要细节，但实时处理是却是另一个更大的挑战。三维点云往往伴随着巨大的数据量，未经压缩的原始点云数据存储和传输面临困难，阻碍了点云相关算法在实际应用中的发展。

例如，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。自动驾驶的激光雷达获取的点云同样具有相似大小，而传统的车载网络带宽为 1Mb 到 100Mb，无力承载自动驾驶任务。此外，点云的巨大数据量也在挑战自动驾驶的算法处理速度，同时作为行车记录仪的一部分，点云必须要经过压缩储存，否则会极大提升储存空间的成本。

因此，三维点云的压缩有着较强的现实意义。

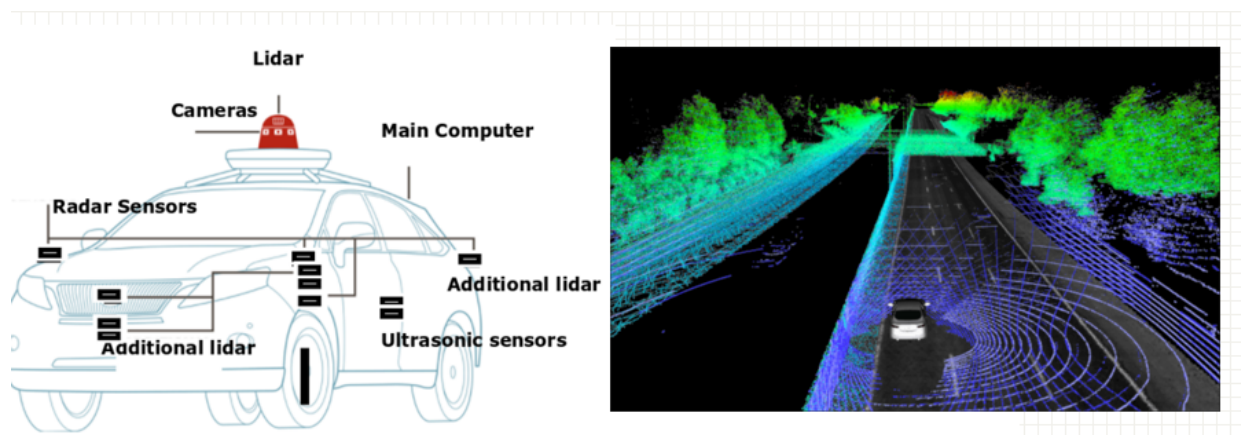


图 1. 自动驾驶

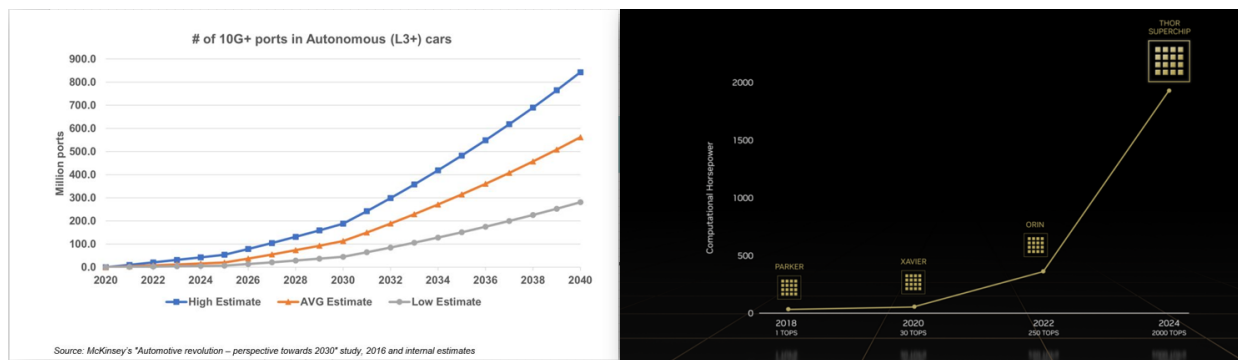


图 2. 自动驾驶的带宽需求和算力需求

与此同时，在不同的应用场景下处理点云数据时，人们对于数据的精度也有着不同的要求，在点云的不同应用场景下，浮点数所提供的 xyz 精度往往是冗余的，可以忽略最小容忍精度（肉眼可辨或 LiDAR 误差）之外的细节信息，而更多地关注精度之内的信息。通过这种方法，不仅能够加速点云数据的处理，也能节省点云数据的存储与传输代价。值得一提的是，知名计算机视觉开源库 OpenCV 也对点云压缩算法提出了需求，OpenCV5.x 版本将会是一次重大更新，与以往版本主要解决二维视觉问题不同，该版本更多的关注三维视觉的解决方案，而点云压缩，是其中亟待实现的基础点云处理算法。在 OpenCV Google Summer of Code 2023 中，社区提出了在 5.x 分支，实现点云压缩算法的需求，旨在丰富 Opencv5.x 分支对于点云处理的方法。综上所述，实现一种可选分辨率的点云压缩算法，在压缩率与还原质量上取得合适的平衡，给予上述问题一个成熟的解决方案，是十分必要的。目前，本项目已成功入选今年 GSoC 社区，希望本项目最终能贡献至 OpenCV，让项目成果服务于社会，实现其现实意义。

点云压缩领域主要分为基于传统方法的点云压缩和基于深度学习的点云压缩。而在传统方法领域，负责开发和制定数字多媒体编码和压缩标准国际标准化组织 Motion Picture Experts Group(MPEG) 于 2020 年推出了两个 3D 点云压缩标准 V-PCC 和 G-PCC。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。

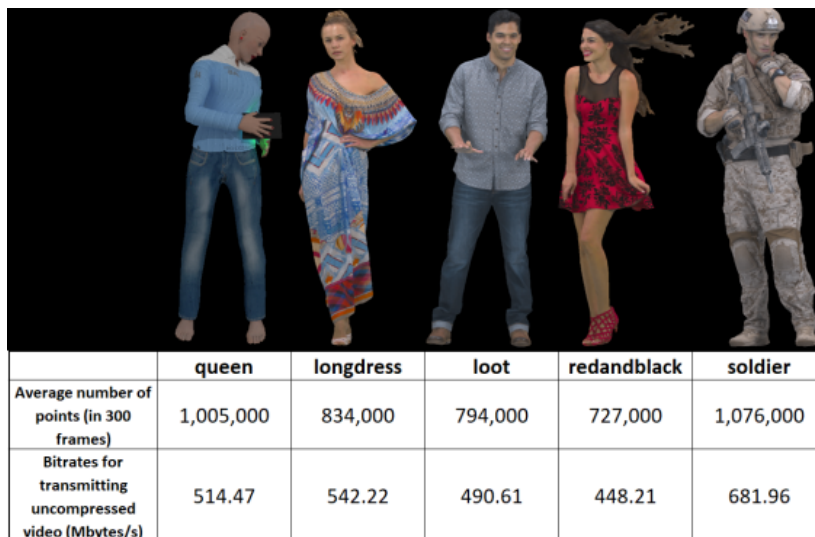


图 3. 带宽估计From Cao C, Preda M, Zaharia T. (2019). 3D Point Cloud Compression: A Survey. 1-9. 10.1145/3329714.3338130.

II. 研究情况

本项目研究历程主要分为两个阶段，分别为多种编码方法的实现与测试分析。

本项目目延续创新实践 I 的成果，在创新实践 I 中，项目成员实现了较为基础的压缩方案，因此，本学期研究的重点方向为压缩方案的改进。这主要体现在实现多种编码方式，以此提高点云的压缩率方面。针对静态点云，本项目采用了直接编码和预测编码相结合的方式，进一步挖掘点云的空间位置信息。针对动态点云，本项目拟采用帧间编码的方式，利用帧间的相关性，提升压缩率。同时，针对点云的颜色这一重要属性，采用 RAHT 算法，实现含颜色点云的编解码，支持颜色点云的压缩与还原。在测试分析阶段，采用 bpp，PSNR 这两项指标作为评估依据。其中，bpp (bit per point) 对应了压缩后点云数据中，平均每个点所占用的比特位数，这直接反映了算法的压缩率。而 PSNR (Peak Signal-to-Noise Ratio) 是衡量图像或视频质量的指标之一，这反映了本算法还原出来点云的质量。

1) 直接编码与预测编码：直接编码是指将八叉树末端稀疏分支直接以 XYZ 坐标的方式进行最简化编码。由于点云的空间相关性集中于八叉树的前几层，包含了点云的大致几何特征。而后几层为每个点的精细坐标划分，与空间中的其他点处于孤立位置。此时使用坐标编码理论上为最小编码方式。同时将细节编码从八叉树中剔除出去，也防止了对八叉树预测编码的污染。

预测编码是指在压缩过程中引入了预测步骤。预测编码利用点云数据中的空间相关性，通过八叉树邻居节点的信息对当前点的位置进行预测，编码残差以减少编码所需的比特数。预测编码可以在一定程度上提高压缩率。

2) 颜色编码：颜色编码主要采用 RAHT 算法实现。RAHT (区域自适应分层变换) 是一种在 3D 点云中类似于自适应 Haar 小波的分层子带变换。它的灵感来自于使用与较低层的八叉树节点关联的颜色来预测下一级别节点的颜色。它通过八叉树扫描的反向路径实现，从单个体素到整个空间，在每个步骤中将相同级别的体素按照每个方向进行聚合，直到达到根节点。之后使用量化器对变换系数进行量化，然后通过算术编码器对每个量化系数进行熵编码。以完全相反的方式可以实现颜色解码。

3) 测试分析：本项目使用八叉树作为承载点云的数据结构，而开源库 PCL(Point cloud library) 作为知名的点云处理工具，实现了诸多的点云处理算法，这其中就包括了点云压缩。由于 PCL 同样采用八叉树作为承载点云的数据结构，因此，本项目在测试分析阶段，主要与 PCL 进行横向对比，指标包括压缩率（通过 bpp 反映），重建质量（通过 PSNR 反映）。

III. 成果总览

在测试分析的过程中，项目成员做了大量的对比实验，实验所得数据表明，本项目实现的点云压缩算法，在压缩率上优于 PCL 库所实现的点云压缩算法。在相同精度设置下的重建质量方面，几乎与 PCL 持平。

PCL (Point Cloud Library) 是一个较为流行的开源点云处理库，提供了丰富的算法和工具，目前仍广泛应用于三维点云数据的处理和分析。

本项目的点云压缩算法在与 PCL 的对比实验中展现出了更好的压缩性能，并且在相同精度设置下的重建质量与 PCL 基本持平。这意味着我们的算法能够在减小点云数据存储空间的同时，保持较高的数据重建质量，为点云数据处理和传输提供了更高效的解决方案。

1) 实验数据集：本项目测试时采用的点云数据 (Andrew、Longdress、Soldier) 来自公开的点云数据集。这些数据集被广泛应用于点云处理和点云压缩的研究领域，在点云压缩领域的论文中常被用作性能测试。

Andrew 数据集来自微软 (Microsoft)，它包含了人物上半身的稠密肖像点云。

Longdress 和 Soldier 数据集来自 8i 公司，它们包含了人物全身的稠密肖像动态点云。

这些数据集在点云压缩领域的论文中常被引用，具有一定的特征和复杂性，评估所提出的点云压缩算法的性能和效果。通过使用这些公开的数据集，本项目的实验结果具有可重复性和可比较性，并能够与其他相关研究进行对比和验证。

这些数据集的单帧点云数量分别在十万和百万级别，测试用点云为连续点云帧中抽取的静态点云。下图展示了用于分析测试的三个点云数据的可视化结果：



图 4. 测试点云（从左至右分别为 andrew,longdress,soldier）

2) 不同压缩精度下的 *bpp*: 由于本项目与 PCL 一样，支持可选精度的压缩。测试集点云均进行过体素化，原始精度为 1，因此对比实验设计如下：分别在五种指定精度下（1, 0.5,0.25, 0.1, 0.05），对三组不同的点云数据进行压缩。并将压缩后文件大小与经 PCL 压缩所得文件大小进行对比。下表展示了三组点云数据压缩后的大小。

精度	andrew	andrew(pcl)	longdress	longdress(pcl)	soldier	soldier(pcl)
1	157	660	473	1948	401	2286
0.5	164	695	497	2065	434	2435
0.25	165	771	501	2305	441	2741
0.1	474	1026	1508	3124	1719	3761
0.05	562	1112	1758	3371	1944	4101

表 I
不同压缩精度下压缩文件的大小（KB）

从上表可以得出，指定的压缩精度（最小体素单位长度）越小，则保留原始点云的几何细节信息就越多，相应的压缩效果也会有所降低。然而实际点云应用中（自动驾驶，点云渲染等）往往只需要点云具有肉眼可辨的细

节程度即可。可选精度的压缩，可以保留用户所关注的关键信息，而忽略不必要的细节坐标信息，起到了降采样的作用，提高压缩率。

值得注意的是，在选定的三组点云数据的测试中，本算法产生的压缩文件大小均小于 PCL。(PCL 的八叉树精度和细节精度均与实验参数设置相同，其余设置依照 PCL 默认参数)

以上为点云压缩文件大小的直观展示。点云压缩中的更为规范的压缩率指标是 bpp (bit per point)，它表示每个点云数据点所占用的平均比特数。该指标用于衡量点云压缩算法在压缩过程中对数据的压缩效率，并且与点云文件的编码格式以及点云本身的点数无关。较低的 bpp 值表示更高的压缩效率，因为它意味着在保持一定的数据质量的情况下，每个数据点所需的存储空间更少。

在点云数据处理和存储中，bpp 压缩率通常是一项重要的指标，因为它直接影响存储需求和数据传输速度。

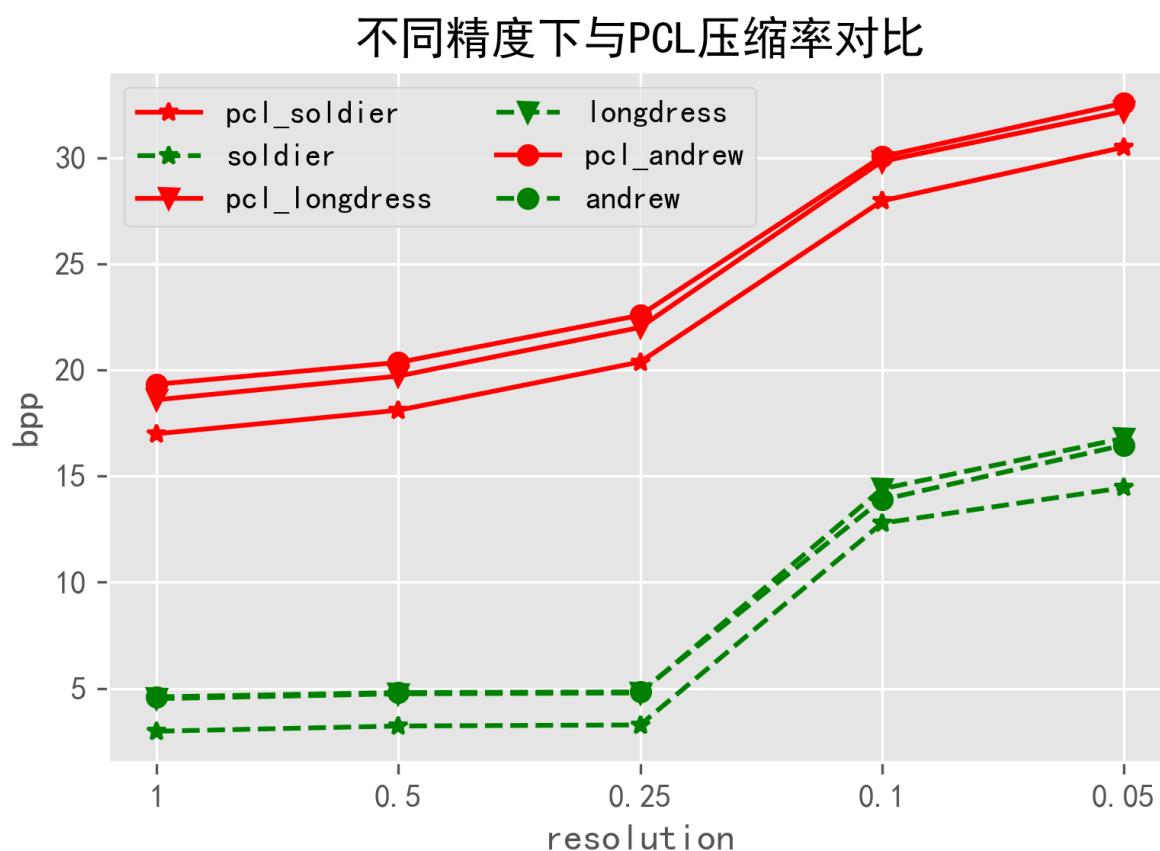


图 5. 不同压缩精度下的 bpp

经过实验数据的计算，两种方法下各压缩文件的 bpp 如上图所示。对于含颜色属性的点云而言，每个原始数据点需要使用三个浮点数表示几何信息 (x,y,z)，三个字节表示颜色信息 (RGB)，总共占用 15byte/point。然而，通过本算法充分挖掘点云数据之间的空间信息，并采用高效的熵编码技术，我们取得了出色的压缩效果。

具体而言，在精度设置为 1 的情况下，我们的压缩算法大幅降低了传输每个点所需的比特量，同时在相同的精度要求下，压缩后的 bpp 值小于使用 PCL 库进行压缩的结果。

这一结果说明我们的算法成功地利用了点云数据的空间相关性，并通过高效的编码方法实现了更好的压缩效果。相较于未经压缩的原始数据，我们能够以更少的存储空间来表示相同数量的数据点，从而在点云数据处理和存储方面实现了显著的空间节省。

3) 不同压缩精度下的峰值信噪比：上一节已经明确展示了本项目在压缩率方面优于 PCL 的结果。然而，除了压缩率外，还原质量同样是点云压缩中的一个关键指标。如果压缩后的点云数据无法提供满意的还原质量，具体表现为过度的几何信息降采样和颜色失真，那么压缩同样无法达到实用性的需求。因此，本节将同时比较不同压缩精度下的峰值信噪比（Peak Signal-to-Noise Ratio, PSNR），以评估本项目压缩还原后的点云质量是否符合要求。

PSNR 是一种常用的质量评估指标，它衡量了原始信号与压缩后信号之间的失真程度。PSNR 的计算方式如下：

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX^2}{MSE} \right)$$

其中，MAX 表示信号的最大可能取值，依据 MPEG 提供的标准，该值设定为恰好包围点云的包围盒对角线大小。MSE 表示均方误差（Mean Squared Error），用于衡量压缩后点云数据与原始数据之间的差异。MSE 的计算方式如下：

$$MSE = \frac{1}{N} \sum_{i=1}^N (d_i - r_i)^2$$

该值为压缩重建点云与原始点云每个点之间的偏移量，即欧式距离。

在相同的点云包围盒大小（峰值信号）下，每个点之间的偏移量（噪声）越小，峰值信噪比越大。故较高的 PSNR 值表示更好的还原质量。

表 II, 表 III, 表 IV 列出了三组点云数据在不同压缩精度下的峰值信噪比与 bpp 的关系。

精度	mseF	mseF(pcl)	colorr	color(pcl)	colorg	colorg(pcl)	colorb	colorb(pcl)
1(andrew)	23.3445375	23.3445375	27.3888597	28.5016731	39.3251784	40.9493512	39.025571	40.5516517
0.5(andrew)	29.3651374	29.3651374	32.9422533	43.5007657	39.4318651	50.9516353	39.2067016	50.5603898
0.25(andrew)	35.3857373	35.3857373	32.9422533	43.5007657	39.4318651	50.9516353	39.2067016	50.5603898
0.1(andrew)	43.3450616	43.3451898	32.9460122	43.5007657	39.169495	50.9516353	38.9098342	50.5603898
0.05(andrew)	49.3666861	49.3667414	32.9460122	43.5007657	39.169495	50.9516353	38.9098342	50.5603898
1(longdress)	13.0103	13.0103	30.4182763	30.6527108	39.1336922	40.943591	37.6510516	38.8653872
0.5(longdress)	19.0308999	19.0308999	39.6621261	43.4339236	42.1592881	51.0134545	42.1488242	50.5370141
0.25(longdress)	25.0514998	25.0514998	33.3003299	43.4339236	37.6179191	51.0134545	37.2905057	50.5370141
0.1(longdress)	33.0113147	33.0113274	33.2465978	43.4339236	37.5350615	51.0134545	37.2139472	50.5370141
0.05(longdress)	39.0311051	39.0340672	33.2465978	43.4339236	37.5350615	51.0134545	37.2139472	50.5370141
1(soldier)	13.0103	13.0103	34.8365562	35.2267779	50.1426207	49.9958509	50.5943164	49.6323744
0.5(soldier)	19.0308999	19.0308999	40.4334088	43.4233222	50.0013039	51.1129872	50.567077	50.5556745
0.25(soldier)	25.0514998	25.0514998	34.8981682	43.4233222	47.4808596	51.1129872	47.9784677	50.5556745
0.1(soldier)	33.0111148	33.0111729	34.8032716	43.4233222	47.4389696	51.1129872	47.9785884	50.5556745
0.05(soldier)	39.0305816	39.0346872	34.8032716	43.4233222	47.4389696	51.1129872	47.9785884	50.5556745

表 II

三组点云数据不同压缩精度峰值信噪比

根据实验过程中我们发现，由于均使用八叉树作为点云压缩算法的框架，本项目在点云的几何信息还原质量方面与 PCL 基本一致，而在颜色信息方面略有差异。从实验结果来看，本项目在相同的点云还原质量的情况下，压缩率优于 PCL。

图 5、图 6 展示了三组点云数据在不同 bpp 下的 PSNR 比较结果。左半部分为几何信息的 PSNR-bpp 曲线，而右半部分则为颜色属性（三个颜色通道）的 PSNR-bpp 曲线。

以 andrew 点云为例（图 4），根据图像的趋势可以得出以下结论：在相同的 bpp 下，本项目压缩方法的几何信息重建质量优于 PCL；而颜色属性的重建质量低于 PCL，但具有更低的 bpp。



resolution=2

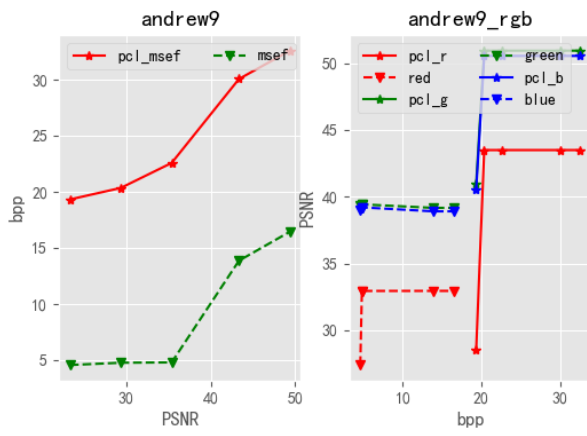


resolution=1

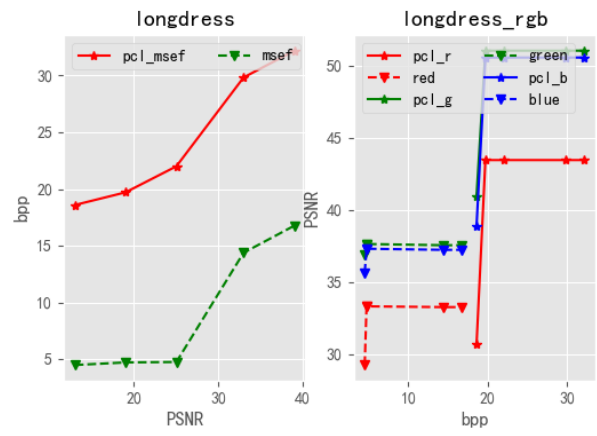


resolution=0.5

图 6. 本项目压缩算法在 andrew 点云上的压缩重建结果



点云 andrew 峰值信噪比



点云 longdress 峰值信噪比

图 7.

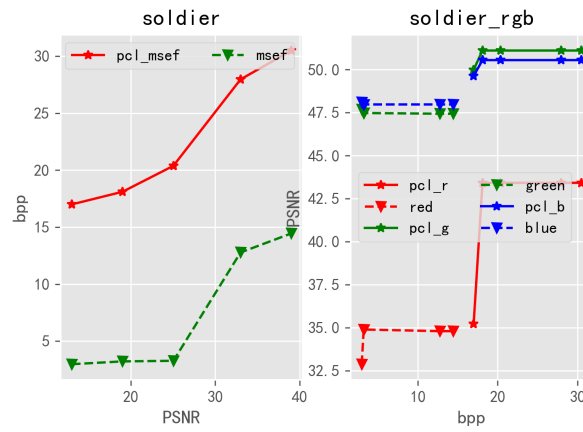


图 8. 点云 soldier 峰值信噪比

通过这些实验结果，我们可以得出结论：本项目在点云压缩算法中实现了与 PCL 相近的还原质量。同时，相同重建质量下，本项目所需的压缩空间更小，即具有更高的压缩率。

IV. 未来改进方向

1) 算法效率的优化：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。后续将重点关注运行时间的优化。本项目拟采用 SIMD 和聚类为主要手段，进行速度的优化。关于聚类算法，此处需要特别说明。由于八叉树是一种将无序的点云数据建模成层次结构分明数据的树结构，在此基础上，通过多种编码方式，进一步挖掘其中的位置关系，使得熵编码能够进一步逼近其理论的压缩极限。因此，我们可以采用聚类的方法，将点云数据划分成位置上相关的不同簇，针对不同的簇，采用并行的方式，进行数据的压缩。由于每个簇内的点云，位置信息更为相近，针对簇内点云使用熵编码的效果必然比直接对原始数据使用熵编码的效果更好。按照这种方法，在提升效率的同时，也提升了压缩率。

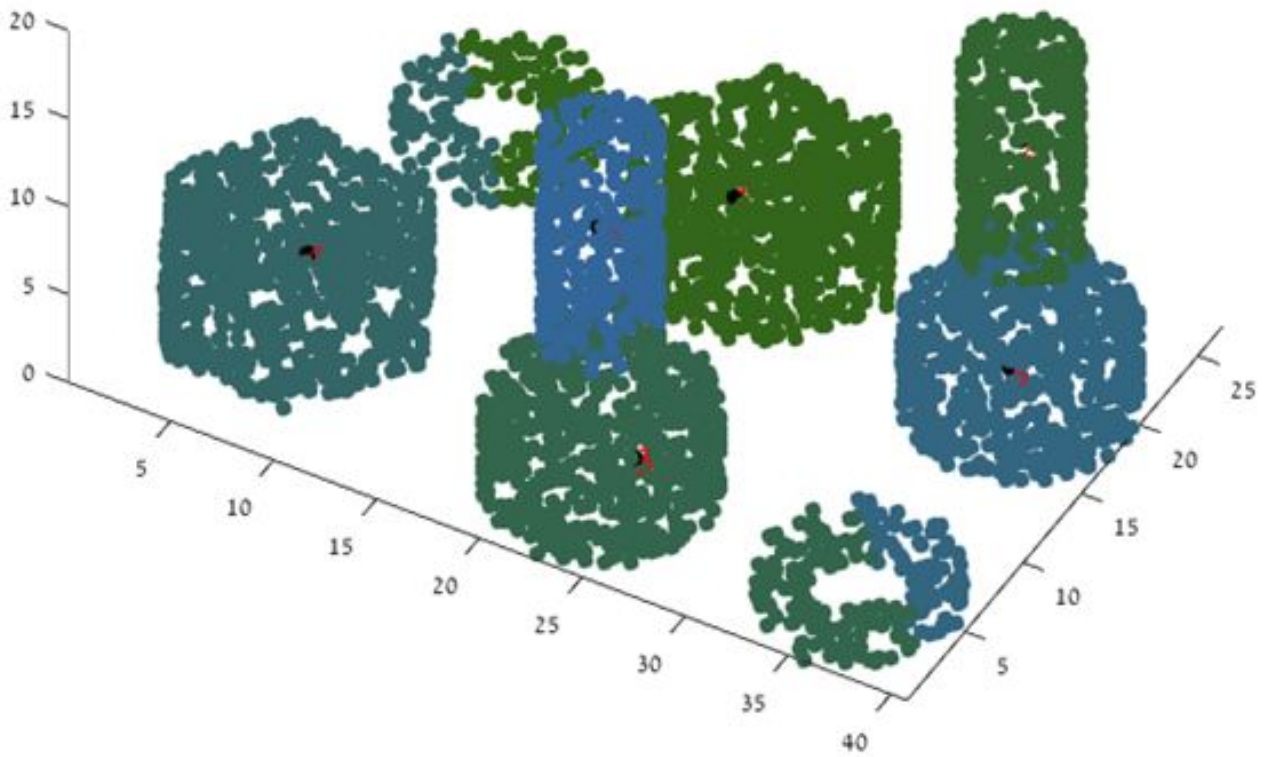


图 9. 三维点云聚类效果

2) 动态点云压缩：目前项目针对的是单帧点云的压缩。现实生活中的应用，如自动驾驶，实时点云传输等，都需要处理动态点云。动态点云即多个点云帧组成的“视频”，现实是动态点云中大部分静态物体（如墙面）的点集处在固定的面上，然而点云本身只是大量点数据的集合，是无序的，没办法体现相邻帧的关联，也不能保证组成一个面的点集在下一帧同样的面上处于同样坐标。八叉树则能很好的在保留目标的三维结构，使点云数据具有了空间相关性。通过不同帧间的八叉树，可以清楚地反映帧间的差异。因此，可以选定一个基本帧，后续特定时间段的帧，只编码帧间差异。这将要用到的数据结构为双缓冲八叉树。

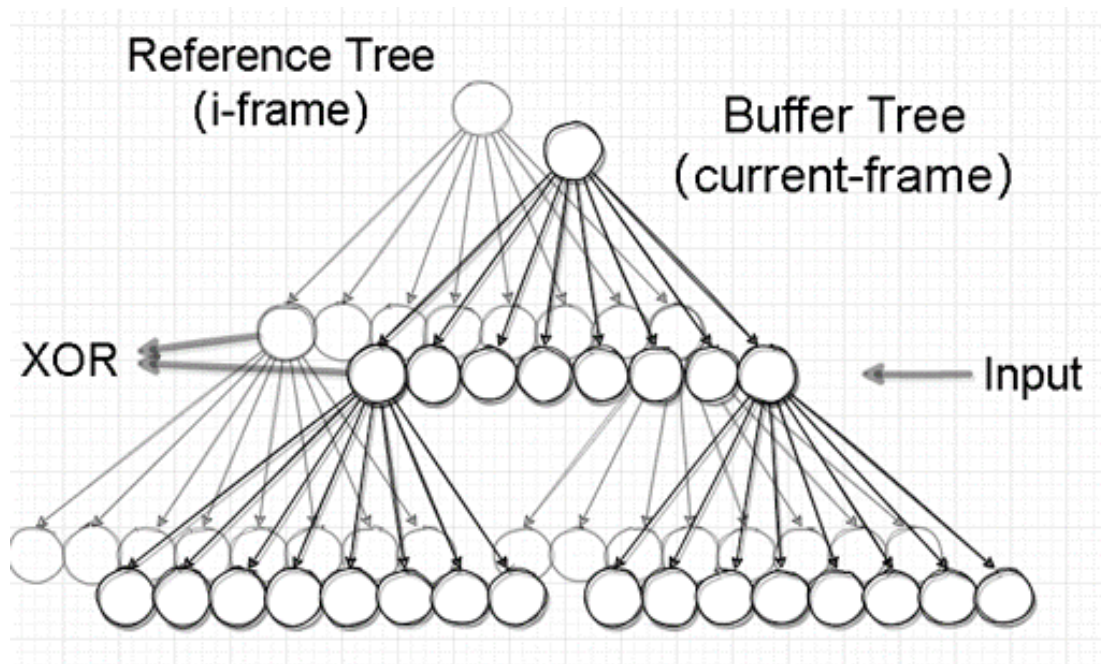


图 10. 双缓冲八叉树结构

未来项目将尝试实现双缓冲机制，识别关键帧用正常方法压缩，而非关键帧编码前后差值。使算法更好地支持动态点云的压缩。

3) 变码率传输：由于压缩流是以层序遍历的顺序输出，而八叉树的每一层在物理意义上可以视作不同清晰度下的 3D 结构。这意味着接收方可以在获取到目标清晰度后随时停止接收，发送方也只需要在本地存放一个高清晰度的八叉树压缩文件，即可满足不同清晰度的发送。这对于减轻网络拥塞具有较大意义。

4) *opencv* 库兼容：由于本项目的代码部分在 *opencv5.x* 下进行开发，需要解决与其他已有的八叉树功能兼容问题，同时，需要兼顾 OpenCV 库的开发风格。由于本项目入选了 2023GSoC，因此，项目成员在社区上提出了一个 issue，tag 为 GSoC。该 issue 为本项目的时间计划表，它将本项目分成了不同的分期交付成果，指导本项目的开发。

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] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro, "Evaluation metrics for point cloud compression," ISO/IEC JTC1/SC29/WG11 MPEG, m39966, 2017.
- [3] 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/AT-SIP.2020.12.
- [4] 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
- [5] MPEG 3DG, Call for proposals for point cloud compression v2, ISO/IEC JTC 1/SC 29/WG 11 N16763, 2017.
- [6] MPEG 3DG, G-PCC codec description v12, ISO/IEC JTC 1/SC 29/WG 7 N0151, 2021.
- [7] 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.
- [8] Grande, V. P., Schaub, M. T. (2023). Topological Point Cloud Clustering. arXiv preprint arXiv:2303.16716.
- [9] De Queiroz, R. L., Chou, P. A. (2016). Compression of 3D point clouds using a region-adaptive hierarchical transform. IEEE Transactions on Image Processing, 25(8), 3947-3956.

- [10] Kammerl, J., Blodow, N., Rusu, R. B., Gedikli, S., Beetz, M., Steinbach, E. (2012, May). Real-time compression of point cloud streams. In 2012 IEEE international conference on robotics and automation (pp. 778-785). IEEE.
- [11] Fan, Y., Huang, Y., Peng, J. (2013, October). Point cloud compression based on hierarchical point clustering. In 2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (pp. 1-7). IEEE.
- [12] Huang, Y., Peng, J., Kuo, C. C. J., Gopi, M. (2008). A generic scheme for progressive point cloud coding. IEEE Transactions on Visualization and Computer Graphics, 14(2), 440-453.
- [13] Liu, H., Yuan, H., Liu, Q., Hou, J., Liu, J. (2019). A comprehensive study and comparison of core technologies for MPEG 3-D point cloud compression. IEEE Transactions on Broadcasting, 66(3), 701-717.