

八叉树结构下动态点云的无损几何编码方法

王哲诚¹, 万帅^{1,2}, 魏磊¹, 杨付正³

(1. 西北工业大学电子信息学院, 710129, 西安; 2. 皇家墨尔本理工大学工程学院, VIC3001, 澳大利亚墨尔本; 3. 西安电子科技大学通信工程学院, 710071, 西安)

摘要: 为进一步提高动态点云无损压缩的性能,提出了一种八叉树结构下的几何信息熵编码方法。针对帧内空间相关性,利用当前八叉树节点的已编解码邻域信息,建立帧内邻居节点上下文和帧内邻居父节点上下文。针对帧间时间相关性,将已编解码的上一帧点云作为参考帧,并将参考帧中与当前八叉树节点同位置的八叉树节点作为参考节点。使用参考节点及其父节点进行帧间上下文建模。为充分利用已建模的上下文,并准确地估计不同上下文下当前节点为非空的条件概率,提出了一种基于指数移动平均的二级概率估计方法:分别在帧内邻居节点上下文和帧内邻居父节点上下文下进行概率估计;利用概率估计结果和帧间上下文建模新的二级上下文,并在二级上下文下再次进行概率估计;采用二进制算术编码器实现无损压缩。选取常用的微软体素化人物上半身和 8i 体素化人物全身数据集进行性能测试,实验结果表明:与近年的方法相比,所提方法的无损压缩性能更高,平均编码增益达到 2.2%~28.7%。

关键词: 点云压缩;八叉树;熵编码;算术编码;概率估计

中图分类号: TP391.4 **文献标志码:** A

DOI: 10.7652/xjtuxb202312002 **文章编号:** 0253-987X(2023)12-0011-09

Lossless Geometry Coding Method for Dynamic Point Clouds under Octree Structure

WANG Zhecheng¹, WAN Shuai^{1,2}, WEI Lei¹, YANG Fuzheng³

(1. School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710129, China;

2. School of Engineering, Royal Melbourne Institute of Technology, Melbourne VIC3001, Australia;

3. School of Telecommunication Engineering, Xidian University, Xi'an 710071, China)

Abstract: To further improve the performance of lossless compression of dynamic point clouds, a method of geometry information entropy coding under an octree structure was proposed. By this method, for intra-frame spatial correlation, the contexts of intra-frame neighbor nodes and the contexts of intra-frame neighbor parent nodes are established based on the coded neighborhood information of the current octree node; for inter-frame temporal correlation, the previously coded point cloud frame is employed as the reference frame, and the octree node inside the reference frame at the same position as the current octree node is used as the reference node. The inter-frame context is modeled using the reference node and its parent node. To fully utilize the modeled contexts and accurately estimate the conditional probability that the current node is non-empty un-

收稿日期: 2023-02-14。 作者简介: 王哲诚(1992—),男,博士生;万帅(通信作者),女,教授,博士生导师。 基金项目: 国家自然科学基金资助项目(62101409,62171353)。

网络出版时间: 2023-07-24

网络出版地址: <https://kns.cnki.net/kcms2/detail/61.1069.T.20230724.1136.002.html>

der different contexts, a second-level probability estimation method based on the exponential moving average was proposed. By this method, the probability is first estimated under the context of intra-frame neighbor nodes and the context of intra-frame neighbor parent nodes, respectively. Then, a set of second-level contexts are modeled using the inter-frame contexts and the results of the probability estimation, and the probability is estimated again under the second-level contexts. In the end, the lossless compression is realized by the binary arithmetic encoder. To evaluate the compression performance of the proposed method, the commonly used Microsoft voxelized upper bodies (MVUB) and 8i voxelized full bodies (8iVFB) datasets were selected for performance testing. The experimental results show that the proposed method has a higher lossless compression performance than the methods developed recently, with average coding gains of 2.2% to 28.7%.

Keywords: point cloud compression; octree; entropy coding; arithmetic coding; probability estimation

由于点云在增强现实和虚拟现实的应用迅速发展,人们对压缩三维点云数据的研究兴趣日益强烈^[1]。三维点云是一种由数十万甚至更多不规则的点组成的数据格式。除了点的几何位置信息外,它还包括一些附加属性信息来表示三维物体,比如颜色和反射率^[2]。与由规则的二维网格中像素表示的图像和视频相比,由三维空间中不规则点组成的点云更加难以被压缩。作为属性信息压缩的前提,几何信息的压缩也更具挑战性。此外,很多应用更是要求对动态点云的几何信息进行压缩。面对这一挑战,人们提出了很多动态点云的几何信息压缩方法^[3-4]。

目前,点云的几何信息压缩通常是基于八叉树结构进行的^[5]。八叉树结构能够比较容易地表示三维空间中点的几何位置,而且便于探索点云内部的空间相关性。早期,人们采用相邻帧的八叉树节点信息进行帧间编码,比如基于相邻帧对当前八叉树节点信息进行重排序来构建有效的上下文^[6],或者基于当前八叉树节点与相邻帧内八叉树节点之间的距离构建上下文^[7]。在八叉树结构下,当前八叉树节点的父节点信息也可以用作上下文。最后,再通过基于上下文的算数编码或者 LZW 编码实现点云的无损压缩^[8]。

2020 年,文献 [9]提出了一种基于八叉树结构的动态点云无损几何编码方法 P(Full)。它采用当前帧中当前八叉树节点的父节点信息作为帧内参考节点,或者采用上一帧中的节点信息作为帧间参考节点。该方法计算当前节点与各参考节点的最小汉明距离,选择汉明距离最小的参考节点作为上下文,通过基于上下文的算数编码进行无损压缩。相较于点云在八叉树结构下且不进行任何压缩的数据大小,该

方法带来了约 55% 的无损几何编码增益。然而,该方法并没有同时使用帧内和帧间的几何信息作为上下文,而是用基于最小汉明距离的方法选择其一,未能综合利用空间和时间相关性。此外,国际标准化组织运动图像专家组 (Moving Picture Expert Group, MPEG) 开发了基于几何的点云压缩 (geometry-based point cloud compression, G-PCC)^[10]。G-PCC 以通用点云编解码器为目标,力求涵盖静态和动态物体场景下的稀疏和密集点云的压缩。G-PCC 标准于 2023 年 3 月正式发布^[11],现阶段只涉及帧内编解码。G-PCC 主要采用八叉树结构进行几何编码,应用当前八叉树节点的父节点、相邻节点和相邻子节点的几何信息进行上下文建模,进而采用基于上下文自适应的二进制算术编码 (context-based adaptive binary arithmetic coding, CABAC)^[12-17]实现点云无损几何编码。因为缺少对帧间时间相关性的利用,G-PCC 在编码动态点云时压缩性能并不突出。

与上述文献在八叉树结构下实现动态点云无损几何编码不同,文献 [18-20]首先利用二元分解将点云沿 x 、 y 、 z 中的某方向平等划分为两部分,再根据轮廓投影将每一部分点云转换为二值轮廓图像,提出了基于轮廓图像中当前像素邻居像素的帧内上下文^[18]、基于参考轮廓图像中同位置像素及其邻居像素的帧内上下文^[18]以及基于上一帧点云同位置轮廓图像中同位置像素及其邻居像素的帧间上下文^[19]。为提高不同上下文条件下概率估计的精度,文献 [20]根据条件熵大小优先挑选高效率的上下文进行 CABAC。随着点云二元分解的进行,轮廓图像被逐一编码。实验结果表明,文献 [20]的

S4D-CSPRune 方法比文献 [9] 的 P(Full) 方法提高了大概 27% 的无损几何编码增益。然而,该方法仅使用了同层级像素,并没有使用已编解码的父层级像素构建上下文,未能有效利用点云的空间相关性。此外,还有大量的研究首先将三维动态点云投影到二维平面上,生成一个二维视频。再利用现有的视频编解码器进行压缩。该类方法掩盖了部分点云真实的三维分布特性,对于动态点云的无损压缩性能并不突出。此类方法通常用于有损压缩^[21-23]。

综合而言,上述方法都没有充分地利用点云帧内的空间相关性和帧间的时间相关性。本文针对动态点云几何信息的无损熵编码,旨在构建更高效的上下文模型,同时改进不同上下文条件下的概率估计,从而提高 CABAC 的编码效率。

1 八叉树结构下的无损几何编码方法

1.1 编码器设计

八叉树结构被广泛应用于表达各种类型的三维点云数据。一个点云被格式化为 $2^D \times 2^D \times 2^D$ 的立方体包围盒,定义为一个 D 位深度的点云。图 1 是一个八叉树分解示意。在八叉树的第一层,包围盒被划分为 8 个尺寸为 $2^{D-1} \times 2^{D-1} \times 2^{D-1}$ 的小立方体。立方体也被称为八叉树节点。当点占据一个八叉树节点时,这个节点再被细分为 8 个子节点。对于空的八叉树节点,则不再进行划分。通过递归地分割八叉树节点,就可以生成一个八叉树结构。

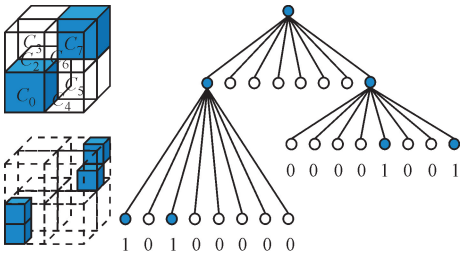


图 1 八叉树分解示意图

Fig. 1 Octree decomposition

为了表示每一个八叉树节点的占用状态,每次划分产生的节点按照莫顿扫描顺序从第 0 个到第 7 个进行索引。占据比特表示一个八叉树节点的占用状态,其定义为

$$c_i = \begin{cases} 1, & \text{如果第 } i \text{ 个节点被占据} \\ 0, & \text{否则} \end{cases} \quad (1)$$

基于八叉树结构的点云压缩即对所有八叉树节点的占据比特进行编码。由于占据比特是二值化符

号,所以非常适合使用二进制算术编码器进行压缩编码。

本文提出了一种八叉树结构下动态点云的无损几何编码方法 (octree structured dynamic point cloud lossless geometry coding method, octree-DPCLGC),编码器如图 2 所示。为提高编码效率,本文首先采用当前八叉树节点在帧内的邻居节点和父节点进行上下文建模。同时,将已编解码的上一帧点云作为参考帧,并将参考帧中与当前八叉树节点同位置的节点作为参考节点。然后,用参考节点及其父节点构建上下文。本文提出了一种不同上下文条件下的二级概率估计方法。最后,采用二进制算术编码 (binary arithmetic coding, BAC) 进行熵编码。

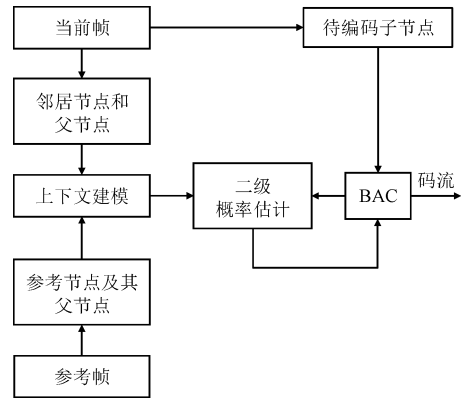


图 2 octree-DPCLGC 方法的编码器框图

Fig. 2 The encoder diagram of the octree-DPCLGC

1.2 点云帧内上下文建模

为充分利用点云帧内的空间相关性,本文分别根据当前八叉树节点的邻居节点和邻居父节点进行上下文建模。

与八叉树节点分解出子节点的莫顿扫描顺序一致,本文所提的编解码方法也采用莫顿顺序。因此,编解码当前八叉树节点时,在当前八叉树层级中只有相对于该节点坐标轴负方向分布的节点是绝对编解码过的。当然,采用全部的已编解码节点构建上下文对复杂度来讲是不现实的。为了确保选取的节点与当前节点有较强的空间相关性,本文在每个分布方向上仅选取 2 个最近的节点参与上下文建模。图 3 展示了当前节点 c_0 选取的 14 个邻居节点,根据占据比特可以构建出 $2^{14} = 16384$ 种上下文。上下文索引 R 可以表示为

$$R = \sum_{j=0}^{13} r_j \times 2^j \quad (2)$$

式中: r_j 代表第 j 个邻居节点的占据比特。

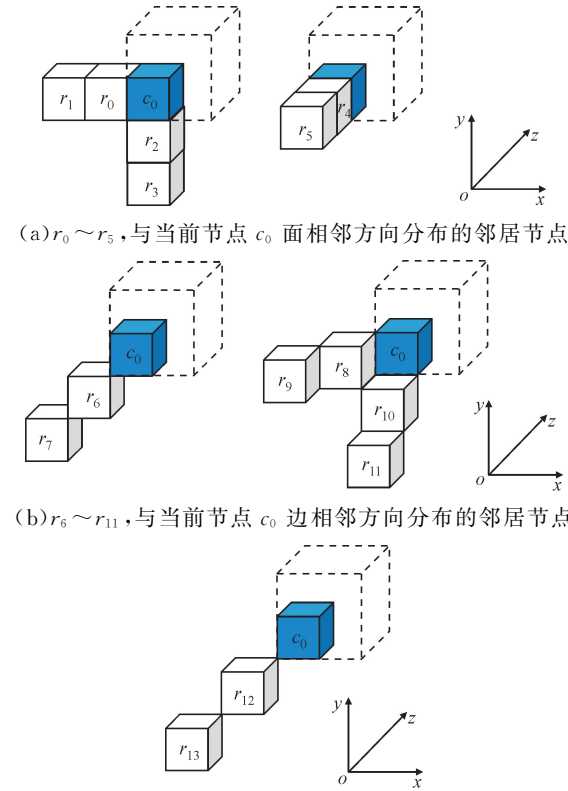
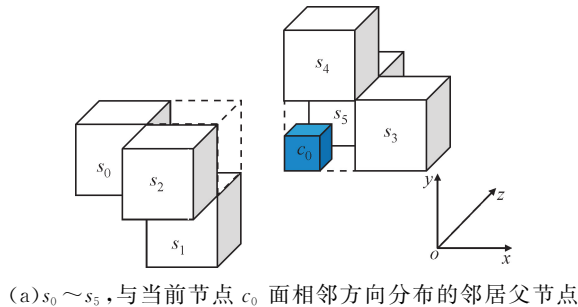


图 3 为当前节点 c_0 选取的邻居节点
Fig. 3 Selected neighboring nodes for the current node c_0

此外,本文也利用当前八叉树节点的邻居父节点进一步构建上下文。不同于当前八叉树层级,在父节点层级中所有父节点都是已编解码过的。因此,为上下文建模选取父节点不再仅限于相对于当前节点坐标轴负方向分布的父节点。不过同样考虑到复杂度的问题,不能采用所有父节点构建上下文。为了确保选取的父节点与当前节点有较强的空间相关性,本文采用与选取邻居节点同样的规则,即每个分布方向上仅选取 2 个最近的父节点参与上下文建模。图 4 展示了当前节点 c_0 选取的 14 个邻居父节点,可以构建 $2^{14}=16\,384$ 种上下文,其索引 S 为

$$S = \sum_{k=0}^{13} s_k \times 2^k \quad (3)$$

式中: s_k 代表第 k 个邻居父节点的占据比特。



(a) $s_0 \sim s_5$,与当前节点 c_0 面相邻方向分布的邻居父节点

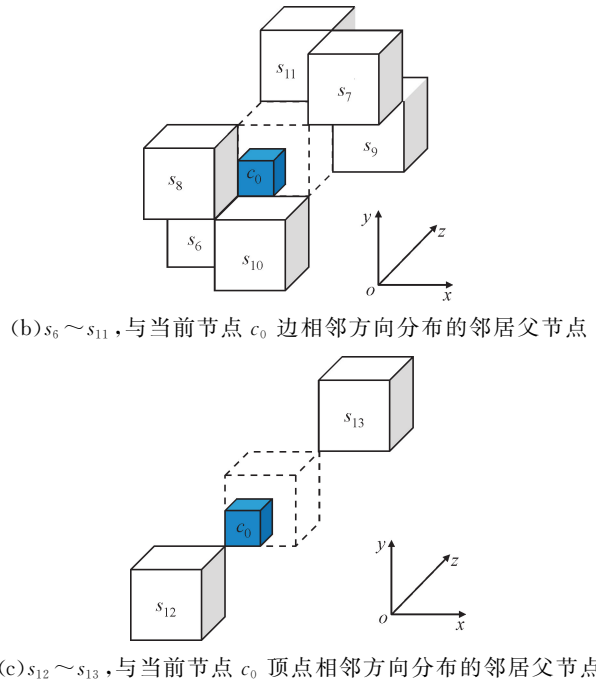


图 4 为当前节点 c_0 选取的邻居父节点
Fig. 4 Selected neighboring parent nodes for the current node c_0

1.3 点云帧间上下文建模

与二维视频相似,三维动态点云的相邻帧之间存在时间相关性。不同的是,这种相关性不仅表现在点的颜色(点云属性信息)上,也表现在点的位置(点云几何信息)上。针对动态点云的几何信息,图 5 展示了动态点云 Redandblack 第 1 450 序号帧

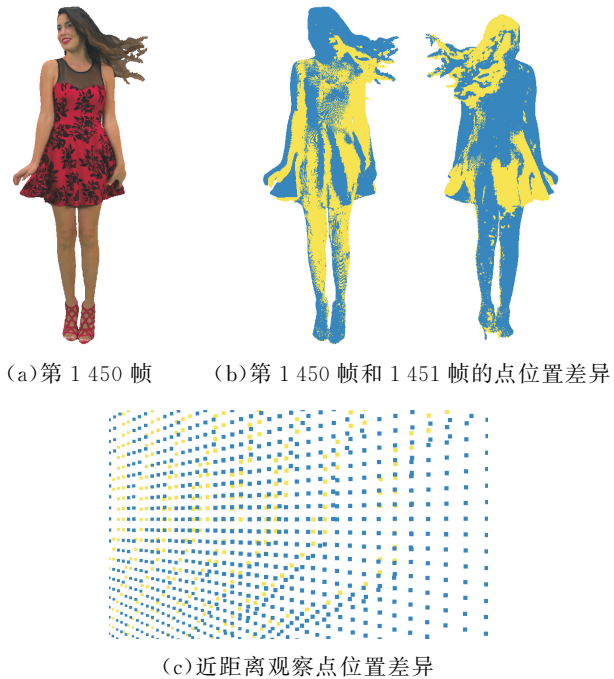


图 5 动态点云 Redandblack 第 1 450 帧和 1 451 帧的差异
Fig. 5 Differences between 1 450th and 1 451st frames of the dynamic point cloud Redandblack

与第 1 451 序号帧之间的点位置差异部分。图中,第 1 450 帧点位置差异用蓝色点表示,第 1 451 帧点位置差异用黄色点表示。

在八叉树结构下,这些点位置的差异势必导致当前帧和参考帧生成不同的八叉树。并且,随着八叉树层级的增加,两个八叉树的差异将会扩大。因此,参考帧的八叉树结构不再按照占据节点继续分解、空节点停止分解的规则来生成,而是在当前帧进行八叉树分解的同时,将参考帧按照同样的树结构进行分解,从而准确地找到参考帧中与当前节点同位置的节点,并作为参考节点。图 6 展示了基于当前帧的参考帧八叉树分解。然而,上述方法会导致参考帧的八叉树结构中存在空的八叉树节点继续分解的情况,如图 6(b)所示。

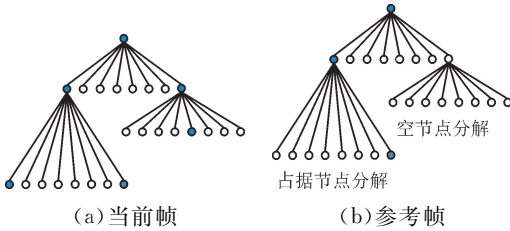


图 6 基于当前帧的参考帧八叉树分解

Fig. 6 Octree decomposition of the reference frame based on the current frame

为确保准确地利用点云帧间的时间相关性,空节点分解出的空子节点与占据节点分解生成的空子节点需要被区分出来。因此,参考节点的占据比特 n 及其父节点的占据比特 n_{parent} 都需要参与上下文建模。值得注意的是,当 $n_{\text{parent}} = 0$ 时, n 也一定为 0。因此,参考节点及其父节点仅构建 3 种上下文,其索引可以表示为

$$U = n + n_{\text{parent}} \quad (4)$$

1.4 二级概率估计

在基于上下文的熵编码中,上下文信息越多,熵就越小或者保持不变。前提条件为:必须在编解码开始前得到其符号概率。然而,当前被应用于点云压缩的 CABAC 并不满足这一条件。为避免预先计算概率带来的复杂度和延迟,CABAC 是在编解码过程中实时估计并更新概率的。

与传统二维视频编解码方法中 CABAC 所采用的概率估计类似^[12-16],本文也基于指数移动平均^[24]进行概率估计。给定一个占据比特序列 b_t ,其中 $t \in \{1, \dots, N\}$ 。占据比特 $b_{t+1} = 1$ 的概率 p_{t+1} 为

$$p_{t+1} = \alpha \sum_{i=0}^{t-1} (1-\alpha)^i b_{t-i} \quad (5)$$

式中: $\alpha \in [0, 1]$ 是一个决定概率更新速度的参数。式(5)还可以通过迭代的形式描述

$$p_{t+1} = p_t(1-\alpha) + b_t\alpha \quad (6)$$

式中: p_t 是上一时刻的估计概率。为了避免使用小数,简化概率估计的实现复杂度,参数 α 的取值被限定为 $\alpha = 2^{-\beta}$,且 $\beta \in \mathbf{N}^+$ 。因此,式(5)可改写为

$$q_{t+1} = q_t - (q_t >> \beta) + b_t((2^w - 1) >> \beta) \quad (7)$$

式中: q_t 是 p_t 的 w 比特整数表现形式; $>>$ 是右移运算符。本文采用 16 比特整数来表示概率,即 $w = 16$ 。这比二维视频编码标准高效率视频编码(high efficiency video coding, HEVC)的 7 比特多,但比通用视频编码(versatile video coding, VVC)中双概率估计总共采用的 24 比特少^[16]。此外,本文根据实验经验设定 $\beta=6$ 。

然而,由本文提出的帧内上下文 R 、 S 和帧间上下文 U 组成的上下文数量 ($16\ 384 \times 16\ 384 \times 3 = 805\ 306\ 368$) 非常庞大。这就意味着,超过 8 亿个概率 q_{t+1} 需要被维护和更新,共占用 1.5 GB 的内存。这空前提升了硬件实现的复杂度。此外,这些上下文中必然包含了大量的罕见上下文。因为缺少输入比特 b_t ,这些罕见上下文相应的概率就不会被很好地估计和更新。使用不准确的概率将会导致编解码器的性能欠佳。综合而言,直接基于 8 亿多种上下文进行概率估计的方法不可行。

本文提出了一种二级概率估计的方法,如图 7 所示。帧内上下文 R 、 S 的数量是造成上一段所述问题的根本原因。因此,首先分别基于 R 、 S 进行概率估计,得出 $q_{t+1}(R)$ 和 $q_{t+1}(S)$ 。将概率划分为 2^θ 个大小相等的概率区间,作为优化后的 2^θ 种上下文。上下文索引可以通过右移运算得出

$$R' = q_{t+1}(R) >> (16 - \theta) \quad (8)$$

$$S' = q_{t+1}(S) >> (16 - \theta) \quad (9)$$

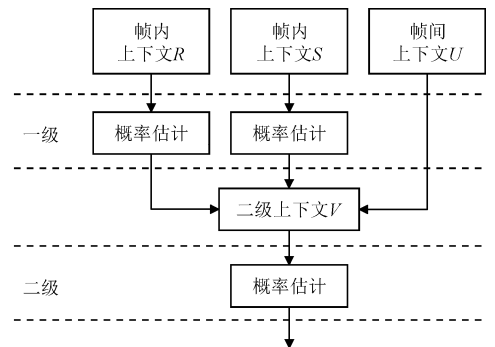


图 7 本文提出的二级概率估计方法

Fig. 7 The proposed second-level probability estimation

然后,基于优化后的帧内上下文 R' 、 S' 和帧间上下文 U 构建新的二级上下文 V 。通过调整 θ ,可以将二级上下文 V 的数量控制在合理范围内。根据实验经验,本文设定 $\theta = 5$ 。因此,二级上下文 V 的数量为 $2^5 \times 2^5 \times 3 = 3\,072$ 。最后,基于二级上下文 V 进行概率估计。

2 实验与分析

2.1 实验方法

为评估本文方法 octree-DPCLGC 的编码性能,使用 C++ 在个人计算机上完成了软件实现,并与下列点云无损几何编码方法进行了比较:基于二叉树结构的方法 P(Full)^[9]、国际标准 G-PCC^[11]、基于二值图像的方法 S4D-CSPRune^[20]。为直接与 P(Full)和 S4D-CSPRune 在各自文献中的实验结果对比,分别选取了微软体素化人物上半身(Microsoft voxelized upper bodies, MVUB)^[25]和 8i 体素化人物全身(8i voxelized full bodies, 8iVFB)^[26]两类点云数据集的第 2 帧到第 101 帧作为测试集。这两类点云数据集被广泛应用于评估动态点云压缩方法的性能^[6-9, 18-20]。表 1 给出了两类点云数据集中每个点云序列的名称、每占据体素比特数(I_{bpov})。图 8 和图 9 展示了每个点云序列的单帧内容。

表 1 测试集详细信息

Table 1 Detailed information of the test datasets

数据集	点云序列	每占据体素比特数
MVUB	Andrew	27
	David	
	Phil	
	Ricardo	
	Sarah	
8iVFB	LongDress	30
	Loot	
	RedAndBlack	
	Soldier	



(a) Andrew



(b) David



(c) Phil



(d) Ricardo



(e) Sarah

图 8 微软体素化人物上半身数据集

Fig. 8 Microsoft voxelized upper bodies datasets



(a) LongDress



(b) Loot



(c) RedAndBlack



(d) Soldier

图 9 8i 体素化人物全身数据集

Fig. 9 8i voxelized full bodies datasets

2.2 编码性能分析

点云无损几何编码的性能由压缩率 ρ 进行评估

$$\rho = \frac{I_{\text{bpov, compressed}}}{I_{\text{bpov, original}}} \times 100\% \tag{10}$$

式中: $I_{\text{bpov, compressed}}$ 表示点云被压缩后的每占据体素比特数; $I_{\text{bpov, original}}$ 表示原始点云的每占据体素比特数。从定义可见,压缩率越小说明压缩方法性能越好。

本文方法 octree-DPCLGC 相较于 P(Full)、G-PCC 和 S4D-CSPRune 方法的编码增益为

$$G = \frac{I_{\text{bpov, octree-DPCLGC}} - I_{\text{bpov, compared}}}{I_{\text{bpov, compared}}} \times 100\% \tag{11}$$

式中: $I_{\text{bpov, octree-DPCLGC}}$ 表示 octree-DPCLGC 方法压缩点云后的每占据体素比特数; $I_{\text{bpov, compared}}$ 表示对照组方法压缩点云后的每占据体素比特数。 G 为负数意味着 octree-DPCLGC 压缩后的点云数据更小,即取得了性能提升。反之, G 为正数意味着性能降低。

不同动态点云无损几何编码方法的性能比较如表 2 所示。可以看出, octree-DPCLGC 方法在 MVUB 和 8iVFB 数据集上分别表现出了较为平稳的编码性能,其压缩率为 3% 左右。其中, octree-DPCLGC 方法在点云序列 Soldier 上的性能尤其突出,压缩率为 2.3%,明显优于其他点云序列的压缩率。同时,观察到 P(Full) 和 S4D-CSPRune 方法也

存在同样的现象。根据对点云序列 Soldier 的内容分析可知,该动态点云中人物运动变化较小,有大量点云帧展现了人物站立不动的场景。与运动幅度大的点云序列相比,该点云序列拥有较强的时间相关性,从而提高了帧间上下文的效率。

分析 octree-DPCLGC 方法的相对编码增益可知,octree-DPCLGC 性能明显优于 P(Full)和 G-PCC,其

原因主要在于本文提出的二级概率估计能够更高效地利用邻域和时域信息,显著提升了熵编码效率。与基于二值图像的方法 S4D-CSPRune 相比,octree-DP-CLGC 方法在 MVUB 和 8iVFB 数据集上的平均编码增益分别为-1.5%和-2.8%。具体地,octree-DP-CLGC 方法为 9 个点云序列中的 6 个带来了编码增益,最高的为点云序列 LongDress 的-6.8%。

表 2 不同动态点云无损几何编码方法的性能比较
Table 2 Performance comparison of different dynamic point cloud lossless geometry coding methods

数据集	点云 序列	P(Full)		G-PCC		S4D-CSPRune		octree- DPCLGC		G/%		
		I_{bpov}	$\rho/\%$	I_{bpov}	$\rho/\%$	I_{bpov}	$\rho/\%$	I_{bpov}	$\rho/\%$	对照组方法 为 P(Full)	对照组方法 为 G-PCC	对照组方法为 S4D-CSPRune
MVUB	Andrew	1.37	5.1	1.14	4.2	0.95	3.5	0.95	3.5	-30.9	-17.2	0.0
	David	1.31	4.9	1.07	3.9	0.94	3.5	0.90	3.3	-31.0	-15.2	-4.3
	Phil	1.42	5.3	1.17	4.3	1.02	3.8	1.01	3.7	-28.9	-13.7	-1.0
	Ricardo	1.34	5.0	1.06	3.9	0.90	3.3	0.91	3.4	-32.0	-13.9	1.1
	Sarah	1.37	5.1	1.08	4.0	0.92	3.4	0.89	3.3	-34.8	-17.0	-3.3
	平均值	1.36	5.0	1.10	4.1	0.95	3.5	0.93	3.4	-31.5	-15.4	-1.5
8iVFB	LongDress	1.13	3.8	1.02	3.4	0.88	2.9	0.82	2.7	-27.0	-19.1	-6.8
	Loot	1.02	3.4	0.95	3.2	0.84	2.8	0.81	2.7	-20.7	-15.2	-3.6
	RedAndBlack	1.23	4.1	1.09	3.6	0.94	3.1	0.89	3.0	-27.7	-18.1	-5.3
	Soldier	0.85	2.8	1.03	3.4	0.65	2.2	0.68	2.3	-20.4	-34.5	4.6
	平均值	1.06	3.5	1.02	3.4	0.83	2.8	0.80	2.7	-24.3	-21.8	-2.8
	总平均值	1.23	4.4	1.07	3.8	0.89	3.0	0.87	2.9	-28.7	-18.1	-2.2

注:表中加粗数据为相同点云测试序列下 4 种方法中压缩性能最好的一组。

2.3 时间复杂度分析

在一台配置了 Intel Core i7-7500U CPU @ 2.70 GHz 和 24 GB 内存的个人计算机上测试了本文方法 octree-DPCLGC 的编解码时间。各点云序列从 2~101 帧的平均每帧编解码时间和平均每帧占据体素数量如表 3 所示。分析可知,点云序列的平均每帧编解码时间与平均每帧占据体素数量呈线性正相关。需要说明的是,由于 P(Full)^[9]和 S4D-CSPRune^[20]仅是在 Matlab 上进行了仿真实验,所以本文无法与其进行精确的时间复杂度对比。

3 结 论

本文面向动态点云的无损压缩,提出了一种八叉树结构下的几何信息编码方法。基于动态点云帧内的空间相关性,采用当前八叉树节点在各个方向

表 3 octree-DPCLGC 方法的编解码时间

Table 3 Encoding and decoding times of the octree-DPCLGC

数据集	序列	平均每 帧编码 时间/s	平均每 帧解码 时间/s	平均每帧 占据体素 数量/10 ⁵
MVUB	Andrew	0.19	0.14	2.86
	David	0.21	0.15	3.29
	Phil	0.22	0.16	3.42
	Ricardo	0.15	0.10	2.43
	Sarah	0.17	0.12	2.68
	平均值	0.19	0.13	2.93
8iVFB	LongDress	0.56	0.39	8.33
	Loot	0.49	0.33	7.82
	RedAndBlack	0.48	0.33	7.15
	Soldier	0.74	0.53	10.60
	平均值	0.57	0.40	8.48
	总平均值	0.36	0.25	5.40

上的有限个已编解码的邻居节点和父节点进行帧内上下文建模。针对动态点云帧间的时间相关性,将已编解码的上一帧作为参考帧,并将参考帧内同位置的八叉树节点作为参考节点。基于参考节点及其父节点构建帧间上下文。为解决上下文因数量过于庞大而导致的概率估计不准确和复杂度过高的问题,提出了一种二级概率估计的方法。该方法不仅有效地利用了帧内和帧间上下文,同时也将复杂度控制在可接受的程度。实验结果表明,与现有的八叉树结构下的动态点云无损几何编码方法相比,本文 octree-DPCLGC 方法带来了显著的编码增益。与近年来发表的基于二值图像的动态点云无损几何编码方法相比,octree-DPCLGC 方法带来了大约—2.2%的编码增益。

本文在利用动态点云帧间时间相关性的方面还有所欠缺,下一步将进一步研究帧间上下文的建模。此外,引入点云帧间运动向量进一步去除时间冗余也是一个重点研究方向。

参考文献:

- [1] 李厚强,李礼,李竹. 点云编码综述 [J]. 中兴通讯技术, 2021, 27(1): 4-9.
LI Houqiang, LI Li, LI Zhu. A review of point cloud compression [J]. ZTE Technology Journal, 2021, 27(1): 4-9.
- [2] RUSU R B, COUSINS S. 3D is here: point cloud library (PCL) [C]//2011 IEEE International Conference on Robotics and Automation. Piscataway, NJ, USA: IEEE, 2011: 1-4.
- [3] CAO Chao, PREDA M, ZAHARIA T. 3D point cloud compression: a survey [C]//24th International Conference on 3D Web Technology. New York, NY, USA: Association for Computing Machinery, 2019: 1-9.
- [4] 张卉冉,董震,杨必胜,等. 点云压缩研究进展与趋势 [J]. 武汉大学学报(信息科学版), 2023, 48(2): 192-205.
ZHANG Huirao, DONG Zhen, YANG Bisheng, et al. Progress and perspectives of point cloud compression [J]. Geomatics and Information Science of Wuhan University, 2023, 48(2): 192-205.
- [5] JACKINS C L, TANIMOTO S L. Oct-trees and their use in representing three-dimensional objects [J]. Computer Graphics and Image Processing, 1980, 14(3): 249-270.
- [6] GARCIA D C, DE QUEIROZ R L. Context-based oc-

- tree coding for point-cloud video [C]//2017 IEEE International Conference on Image Processing (ICIP). Piscataway, NJ, USA: IEEE, 2017: 1412-1416.
- [7] DE QUEIROZ R L, GARCIA D C, CHOU P A, et al. Distance-based probability model for octree coding [J]. IEEE Signal Processing Letters, 2018, 25(6): 739-742.
- [8] GARCIA D C, DE QUEIROZ R L. Intra-frame context-based octree coding for point-cloud geometry [C]//2018 25th IEEE International Conference on Image Processing (ICIP). Piscataway, NJ, USA: IEEE, 2018: 1807-1811.
- [9] GARCIA D C, FONSECA T A, FERREIRA R U, et al. Geometry coding for dynamic voxelized point clouds using octrees and multiple contexts [J]. IEEE Transactions on Image Processing, 2020, 29: 313-322.
- [10] 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, 2019, 9(1): 133-148.
- [11] ISO. Information technology: coded representation of immersive media: part 9 geometry-based point cloud compression: ISO/IEC23090-9: 2023 [S]. Vernier, Geneva, Switzerland: ISO, 2023.
- [12] MARPE D, SCHWARZ H, WIEGAND T. Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(7): 620-636.
- [13] ALSHIN A, ALSHINA E, PARK J H. High precision probability estimation for CABAC [C]//2013 Visual Communications and Image Processing (VCIP). Piscataway, NJ, USA: IEEE, 2013: 1-6.
- [14] IM S K, CHAN K H. More probability estimators for CABAC in versatile video coding [C]//2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP). Piscataway, NJ, USA: IEEE, 2020: 366-370.
- [15] KARWOWSKI D. Precise probability estimation of symbols in VVC CABAC entropy encoder [J]. IEEE Access, 2021, 9: 65361-65368.
- [16] SCHWARZ H, COBAN M, KARCZEWICZ M, et al. Quantization and entropy coding in the versatile video coding (VVC) standard [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2021, 31(10): 3891-3906.
- [17] SAYOOD K. Introduction to data compression [M]. 4th ed. Waltham, MA, USA: Morgan Kaufmann, 2012.
- [18] PEIXOTO E. Intra-frame compression of point cloud

- geometry using dyadic decomposition [J]. IEEE Signal Processing Letters, 2020, 27: 246-250.
- [19] PEIXOTO E, MEDEIROS E, RAMALHO E. Silhouette 4d: an inter-frame lossless geometry coder of dynamic voxelized point clouds [C]//2020 IEEE International Conference on Image Processing (ICIP). Piscataway, NJ, USA: IEEE, 2020: 2691-2695.
- [20] RAMALHO E, PEIXOTO E, MEDEIROS E. Silhouette 4D with context selection: lossless geometry compression of dynamic point clouds [J]. IEEE Signal Processing Letters, 2021, 28: 1660-1664.
- [21] CAO Chao, PREDA M, ZAKHARCHENKO V, et al. Compression of sparse and dense dynamic point clouds: methods and standards [J]. Proceedings of the IEEE, 2021, 109(9): 1537-1558.
- [22] XIONG Jian, GAO Hao, WANG Miaohui, et al. Advanced geometry surface coding for dynamic point cloud compression [EB/OL]. (2021-03-11) [2023-02-01]. <https://arxiv.org/abs/2103.06549>.
- [23] SHI Haoyu, LI Fan. Patch re-segmentation and packing for dynamic point cloud compression via back-and-forth structure [J]. IEEE Open Journal of Signal Processing, 2022, 3: 155-168.
- [24] HOLT C C. Forecasting seasonals and trends by exponentially weighted moving averages [J]. International Journal of Forecasting, 2004, 20(1): 5-10.
- [25] LOOP C, CAI Q, ESCOLANO S O, et al. Microsoft voxelized upper bodies: a voxelized point cloud dataset [EB/OL]. (2016-05-25) [2023-02-01]. https://dms.mpeg.expert/doc_end_user/current_document.php?id=55507&id_meeting=0.
- [26] D'EON E, HARRISON B, MYERS T, et al. 8i voxelized full bodies-a voxelized point cloud dataset [EB/OL]. (2017-01-12) [2023-02-01]. https://dms.mpeg.expert/doc_end_user/current_document.php?id=57355&id_meeting=0.

(编辑 陶晴 武红江)