自适应分割的视频点云多模式帧间编码方法

陈 建[1,2] 廖燕俊[1] 王 适[2] 郑明魁[1,2] 苏立超[3]

摘 要 基于视频的点云压缩 (Video based point cloud compression, V-PCC) 为压缩动态点云提供了高效的解决方案, 但 V-PCC 从三维到二维的投影使得三维帧间运动的相关性被破坏, 降低了帧间编码性能. 针对这一问题, 提出一种基于 V-PCC 改进的自适应分割的视频点云多模式帧间编码方法, 并依此设计了一种新型动态点云帧间编码框架. 首先, 为实现更精准的块预测, 提出区域自适应分割的块匹配方法以寻找最佳匹配块; 其次, 为进一步提高帧间编码性能, 提出基于联合属性率失真优化 (Rate distortion optimization, RDO) 的多模式帧间编码方法, 以更好地提高预测精度和降低码率消耗. 实验结果表明, 提出的改进算法相较于 V-PCC 实现了−22.57% 的 BD-BR (Bjontegaard delta bit rate) 增益. 该算法特别适用于视频监控和视频会议等帧间变化不大的动态点云场景.

关键词 点云压缩, 基于视频的点云压缩, 三维帧间编码, 点云分割, 率失真优化

引用格式 陈建, 廖燕俊, 王适, 郑明魁, 苏立超. 自适应分割的视频点云多模式帧间编码方法. 自动化学报, 2023, **49**(8): 1707−1722

DOI 10.16383/j.aas.c220549

An Adaptive Segmentation Based Multi-mode Inter-frame Coding Method for Video Point Cloud

CHEN Jian[1,2] LIAO Yan-Jun[1] WANG Kuo[2] ZHENG Ming-Kui[1,2] SU Li-Chao[3]

Abstract Video based point cloud compression (V-PCC) provides an efficient solution for compressing dynamic point clouds, but the projection of V-PCC from 3D to 2D destroys the correlation of 3D inter-frame motion and reduces the performance of inter-frame coding. To solve this problem, we proposes an adaptive segmentation based multi-mode inter-frame coding method for video point cloud to improve V-PCC, and designs a new dynamic point cloud inter-frame encoding framework. Firstly, in order to achieve more accurate block prediction, a block matching method based on adaptive regional segmentation is proposed to find the best matching block; Secondly, in order to further improve the performance of inter coding, a multi-mode inter-frame coding method based on joint attribute rate distortion optimization (RDO) is proposed to increase the prediction accuracy and reduce the bit rate consumption. Experimental results show that the improved algorithm proposed in this paper achieves −22.57% Bjontegaard delta bit rate (BD-BR) gain compared with V-PCC. The algorithm is especially suitable for dynamic point cloud scenes with little change between frames, such as video surveillance and video conference.

Key words Point cloud compression, video-based point cloud compresion (V-PCC), 3D inter-frame coding, point cloud segmentation, rate distortion optimization (RDO)

Citation Chen Jian, Liao Yan-Jun, Wang Kuo, Zheng Ming-Kui, Su Li-Chao. An adaptive segmentation based multi-mode inter-frame coding method for video point cloud. *Acta Automatica Sinica*, 2023, **49**(8): 1707−1722

点云由三维空间中一组具有几何和属性信息的点集构成, 通常依据点的疏密可划分为稀疏点云和密集点云[1]. 通过相机矩阵或高精度激光雷达采集的密集点云结合 VR 头盔可在三维空间将对象或场景进行 6 自由度场景还原, 相较于全景视频拥有更真实的视觉体验, 在虚拟现实、增强现实和三维物体捕获领域被广泛应用[2−3]. 通过激光雷达反射光束经光电处理后收集得到的稀疏点云可生成环境地

收稿日期 2022-07-05 录用日期 2022-11-29
Manuscript received July 5, 2022; accepted November 29, 2022
国家自然科学基金 (62001117, 61902071), 福建省自然科学基金 (2020J01466), 中国福建光电信息科学与技术创新实验室 (闽都创新实验室) (2021ZR151), 超低延时视频编码芯片及其产业化 (2020 年福建省教育厅产学研专项) 资助
Supported by National Natural Science Foundation of China (62001117, 61902071), Fujian Natural Science Foundation (2020J01466), Fujian Science & Technology Innovation Laboratory for Optoelectronic Information of China (2021ZR151), and Ultra-low Latency Video Coding Chip and its Industrialization (2020 Special Project of Fujian Provincial Education Department for Industry-University Research)
本文责任编委 刘成林
Recommended by Associate Editor LIU Cheng-Lin
1. 福州大学先进制造学院 泉州 362251 2. 福州大学物理与信息工程学院 福州 350116 3. 福州大学计算机与大数据学院/软件学院 福州 350116
1. School of Advanced Manufacturing, Fuzhou University, Quanzhou 362251 2. College of Physics and Information Engineering, Fuzhou University, Fuzhou 350116 3. College of Computer and Data Science/College of Software, Fuzhou University, Fuzhou 350116

图, 以实现空间定位与目标检测等功能, 业已应用于自动驾驶、无人机以及智能机器人等场景[4−7]. 但相较于二维图像, 点云在存储与传输中的比特消耗显著增加[8], 以经典的 8i 动态点云数据集[9] 为例, 在每秒 30 帧时的传输码率高达 180 MB/s, 因此动态点云压缩是对点云进行高效传输和处理的前提.

为了实现高效的动态点云压缩, 近年来, 一些工作首先在三维上进行帧间运动估计与补偿, 以充分利用不同帧之间的时间相关性. 其中, Kammerl 等[10] 首先提出通过构建八叉树对相邻帧进行帧间差异编码, 实现了相较于八叉树帧内编码方法的性能提升; Thanou 等[11] 则提出将点云帧经过八叉树划分后, 利用谱图小波变换将三维上的帧间运动估计转换为连续图之间的特征匹配问题. 然而, 上述方法对帧间像素的运动矢量估计不够准确. 为了实现更精确的运动矢量估计, Queiroz 等[12] 提出一种基于运动补偿的动态点云编码器, 将点云体素化后进行块划分, 依据块相关性确定帧内与帧间编码模式, 对帧间编码块使用提出的平移运动模型改善预测误差; Mekuria 等[13] 则提出将点云均匀分割为 $N \times N \times N$ 的块, 之后将帧间对应块使用迭代最近点 (Iterative closest point, ICP)[14] 进行运动估计, 以进一步提高帧间预测精度; Santos 等[15] 提出使用类似于 2D 视频编码器的 N 步搜索算法 (N-step search, NSS), 在 $3 \times 3 \times 3$ 的三维块区域中迭代寻找帧间对应块, 而后通过配准实现帧间编码. 然而, 上述方法实现的块分割破坏了块间运动相关性, 帧间压缩性能没有显著提升.

为了进一步提高动态点云压缩性能, 一些工作通过将三维点云投影到二维平面后组成二维视频序列, 而后利用二维视频编码器中成熟的运动预测与补偿算法, 实现三维点云帧间预测. 其中, Lasserre 等[16] 提出基于八叉树的方法将三维点云投影至二维平面, 之后用二维视频编码器进行帧间编码; Budagavi 等[17] 则通过对三维上的点进行二维平面上的排序, 组成二维视频序列后利用高效视频编码器 (High efficiency video coding, HEVC) 进行编码. 上述方法在三维到二维投影的过程中破坏了三维点间联系, 重构质量并不理想. 为改善投影后的点间联系, Schwarz 等[18] 通过法线将点映射于圆柱体上确保点间联系, 对圆柱面展开图使用二维视频编码以提高性能. 但在圆柱上的投影使得部分点因遮挡丢失, 影响重构精度. 为尽可能保留投影点数, Mammou 等[19] 根据点云法线方向与点间距离的位置关系, 将点云划分为若干 Patch, 通过对 Patch 进行二维平面的排列以减少点数损失, 进一步提高了重构质量.

基于 Patch 投影后使用 2D 视频编码器进行编码, 以实现二维上的帧间运动预测与补偿的思路取得了最优的性能, 被运动图像专家组 (Moving picture experts group, MPEG) 正在进行的基于视频的点云压缩 (Video-based point cloud compression, V-PCC) 标准[20] 所采纳, 但将 Patch 从三维到二维的投影导致三维运动信息无法被有效利用, 使得帧间压缩性能提升受到限制. 针对这一问题, 一些工作尝试在 V-PCC 基础上实现三维帧间预测, 其中, Li 等[21] 提出了一种三维到二维的运动模型, 利用 V-PCC 中的几何与辅助信息推导二维运动矢量以实现帧间压缩性能改善, 但通过二维推导得到的三维运动信息并不完整, 导致运动估计不够准确. Kim 等[22] 提出通过点云帧间差值确定帧内帧与预测帧, 帧内帧用 V-PCC 进行帧内编码, 预测帧依据前帧点云进行运动估计后对残差进行编码以实现运动补偿, 但残差编码依旧消耗大量比特. 上述方法均在 V-PCC 基础上实现了三维点云的帧间预测, 但无论是基于二维的三维运动推导还是帧间残差的编码, 性能改善都比较有限.

在本文的工作中, 首先, 为了改善三维上实现运动估计与补偿中, 块分割可能导致的运动相关性被破坏的问题, 本文引入了 KD 树 (K-dimension tree, KD Tree) 思想, 通过迭代进行逐层深入的匹配块分割, 并定义分割块匹配度函数以自适应确定分割的迭代截止深度, 进而实现了更精准的运动块搜索; 另外, 针对 V-PCC 中二维投影导致三维运动信息无法被有效利用的问题, 本文提出在三维上通过匹配块的几何与颜色两种属性进行相似性判别, 并设计率失真优化 (Rate distortion optimization, RDO) 模型对匹配块分类后进行多模式的帧间编码, 实现了帧间预测性能的进一步改善. 实验表明, 本文提出的自适应分割的视频点云多模式帧间编码方法在与最新的 V-PCC 测试软件和相关文献的方法对比中均取得了 BD-BR (Bjontegaard delta bit rate) 的负增益. 本文的主要贡献如下:

1) 提出了针对动态点云的新型三维帧间编码框架, 通过自动编码模式判定、区域自适应分割、联合属性率失真优化的多模式帧间编码、结合 V-PCC 实现了帧间编码性能的提升;

2) 提出了一种区域自适应分割的块匹配方法, 以寻找帧间预测的最佳匹配块, 从而改善了均匀分割和传统分割算法导致运动相关性被破坏的问题;

3) 提出了一种基于联合属性率失真优化模型的多模式帧间编码方法, 在改善预测精度的同时显著减少了帧间编码比特.

1 基于视频的点云压缩及其问题分析

本文所提出的算法主要在 V-PCC 基础上进行

三维帧间预测改进, 因此本节对 V-PCC 的主要技术做简要介绍, 并分析其不足之处. 其中, V-PCC 编码框架如图 1 所示.

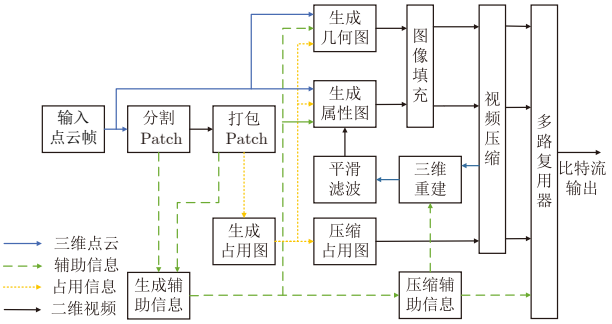
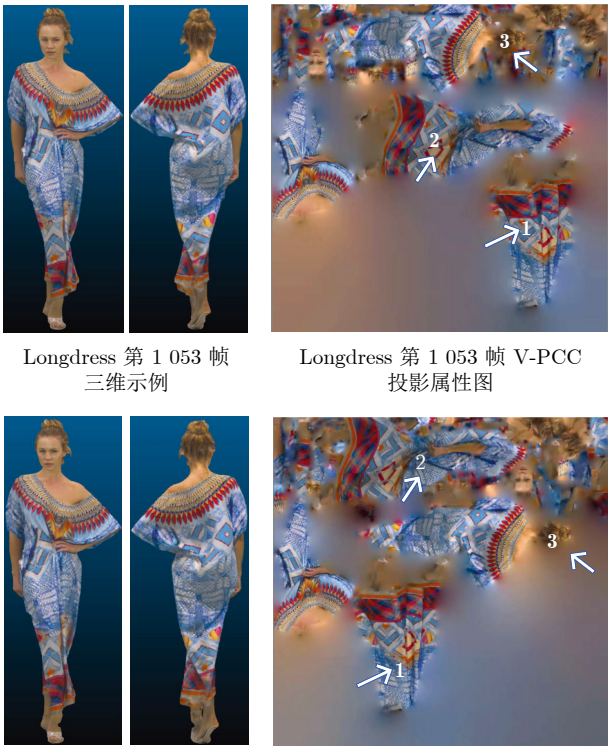


图 1　V-PCC 编码器框架

Fig. 1　V-PCC encoder diagram

首先, V-PCC 计算 3D 点云中每个点的法线以确定最适合的投影面, 进而将点云分割为多个 Patch[23]. 接着, 依据对应 Patch 的位置信息, 将其在二维平面上进行紧凑排列以完成对 Patch 的打包. 之后, 依据打包结果在二维上生成对应的图像, 并使用了几何图、属性图和占用图分别表示各点的坐标、颜色及占用信息. 鉴于 Patch 在二维的排列不可避免地存在空像素点, 因此需要占用图表示像素点的占用与否[24]; 由于三维到二维的投影会丢失一个维度坐标信息, 因此使用几何图将该信息用深度形式进行表示; 为了实现动态点云的可视化, 还需要一个属性图用于表示投影点的颜色属性信息. 最后, 为了提高视频编码器的压缩性能, 对属性图和几何图的空像素进行了填充和平滑处理以减少高频分量; 同时, 为了缓解重构点云在 Patch 边界可能存在的重叠或伪影, 对重构点云进行几何和属性上的平滑滤波处理[25]. 通过上述步骤得到二维视频序列后, 引入二维视频编码器 (如 HEVC) 对视频序列进行编码.

V-PCC 将动态点云帧进行二维投影后, 利用成熟的二维视频编码技术实现了动态点云压缩性能的提升. 但是, V-PCC 投影过程将连续的三维物体分割为多个二维子块, 丢失了三维上的运动信息, 使得三维动态点云中存在的时间冗余无法被有效利用. 为了直观展示投影过程导致的运动信息丢失, 图 2 以 Longdress 数据集为例, 展示了第 1053 和第 1054 两相邻帧使用 V-PCC 投影得到的属性图. 观察图 2 可以发现, 部分在三维上高度相似的区域, 如图中标记位置 1、2 与 3 所对应 Patch, 经二维投影后呈现出完全不同的分布, 该结果使得二维视频编码器中帧间预测效果受到限制, 不利于压缩性能的进一步提升.



Longdress 第 1 053 帧 三维示例　　Longdress 第 1 053 帧 V-PCC 投影属性图

Longdress 第 1 054 帧 三维示例　　Longdress 第 1 054 帧 V-PCC 投影属性图

图 2　V-PCC 从三维到二维投影 (属性图)

Fig. 2　V-PCC projection from 3D to 2D (Attribute map)

2　改进的动态点云三维帧间编码

为了在 V-PCC 基础上进一步降低动态点云的时间冗余性, 在三维上进行帧间预测和补偿以最小化帧间误差, 本文提出了一个在 V-PCC 基础上改进的针对动态点云的三维帧间编码框架, 如图 3 所示. 下面对该框架基本流程进行介绍.

首先, 在编码端, 我们将输入的点云序列通过模块 (a) 进行编码模式判定, 以划分帧内帧与预测帧. 其思想与二维视频编码器类似, 将动态点云划分为多组具有运动相似性的图像组 (Group of pictures, GOP) 以分别进行编码. 其中图像组中的第一帧为帧内帧, 后续帧均为预测帧, 帧内帧直接通过 V-PCC 进行帧内编码; 预测帧则通过帧间预测方式进行编码. 合理的 GOP 划分表明当前图像组内各相邻帧均具有较高运动相关性, 因此可最优化匹配块预测效果以减少直接编码比特消耗, 进而提高整体帧间编码性能. 受文献 [22] 启发, 本文通过对当前帧与上一帧参考点云进行几何相似度判定, 以确定当前帧的编码方式进行灵活的图像组划分. 如式 (1) 所示.

$$mode = \max\left\{\mathrm{sign}\left[E^{G}_{cur,ref} - \Omega\right], 0\right\} \tag{1}$$

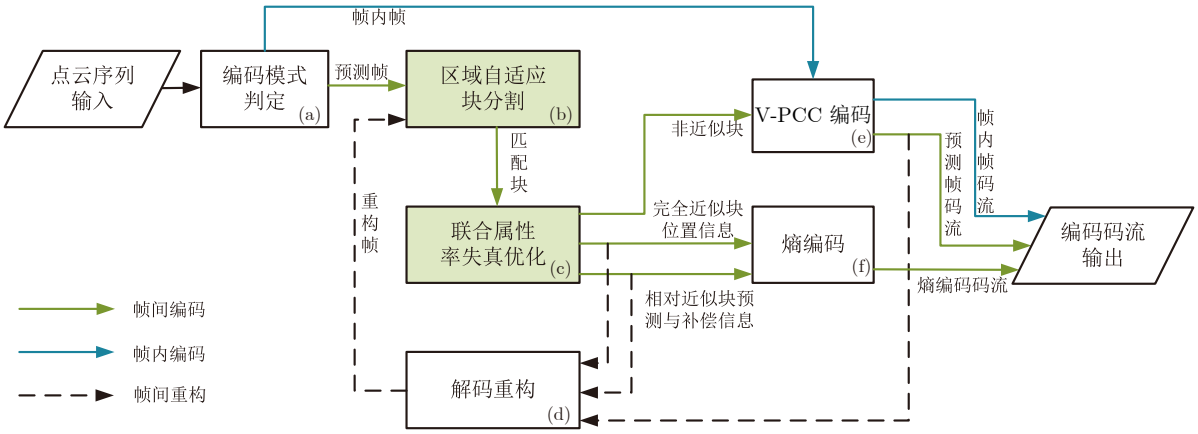


图 3 改进的三维帧间编码框架

Fig. 3 Improved 3D inter-frame coding framework

其中, cur 为当前帧点云, ref 为前帧参考点云, $E^{G}_{cur,\,ref}$ 表示两相邻帧点云的几何偏差, Ω 为编码模式判定阈值. 当 $mode$ 值为 1 时表示当前帧差异较大, 应当进行帧内模式编码; 当 $mode$ 值为 0 时则表示两帧具有较大相似性, 应当进行帧间模式编码. 另外, 在动态点云重构误差 E 的计算中, 使用原始点云 O 中各点与重构点云 R 在几何和属性上的误差均值表示, 即式 (2) 所示.

$$E_{O,\,R} = \frac{1}{N_O}\sum_{i=1}^{N_O}\|O(i) - R(i')\|_2^2 \tag{2}$$

其中, N_O 为原始点云点数, $O(i)$ 和 $R(i')$ 分别表示原始点云第 i 点与对应重构点云 i' 点的几何或属性值, $E_{O,\,R}$ 即为原始点云 O 与重构点云 R 间误差值.

接着, 在进行帧间编码模式判断后, 通过模块 (b) 进行预测帧的区域自适应块分割. 块分割的目的在于寻找具有帧间运动一致性的匹配块以进行运动预测和补偿. 不同于 $N\times N\times N$ 等分或 K 均值聚类, 所提出的基于 KD 树思想的区域自适应块匹配从点云质心、包围盒和点数三个角度, 判断分割块的帧间运动程度以进行分割深度的自适应判定, 最终实现最佳匹配块搜索.

之后, 对于分割得到的匹配块, 通过模块 (c) 进行基于联合属性率失真优化的帧间预测. 在该模块中, 我们通过帧间块的几何与颜色属性联合差异度, 结合率失真优化模型对匹配块进行分类, 分为几乎无差异的完全近似块 (Absolute similar block, ASB)、差异较少的相对近似块 (Relative similar block, RSB) 以及存在较大差异的非近似块 (Non similar block, NSB). 完全近似块认为帧间误差可忽略不计, 仅需记录参考块的位置信息; 而相对近似块则表示存在一定帧间误差, 但可通过 ICP 配准和属性补偿来改善几何与属性预测误差, 因此除了块位置信息, 还需记录预测与补偿信息; 而对于非近似块, 则认为无法实现有效的帧间预测, 因此通过融合后使用帧内编码器进行编码.

最后, 在完成帧间模式分类后, 为了在编码端进行当前帧的重构以作为下一帧匹配块搜索的参考帧, 通过模块 (d) 对相对近似块进行几何预测与属性补偿, 而后将几何预测与属性补偿后的相对近似块、完全近似块、非近似块进行融合得到重构帧. 为了在解码端实现帧间重构, 首先需要组合预测帧中的所有非近似块, 经由模块 (e) 的 V-PCC 编码器进行帧内编码, 并且, 还需要对完全近似块的位置信息、相对近似块的位置与预测补偿信息通过模块 (f) 进行熵编码以实现完整的帧间编码流程.

至此, 整体框架流程介绍完毕, 在接下来的第 3 节与第 4 节中, 我们将对本文提出的区域自适应分割的块匹配算法与联合属性率失真优化的多模式帧间编码方法进行更为详细的介绍, 并在第 5 节通过实验分析进行算法性能测试.

3 区域自适应分割的块匹配

相较于二维视频序列, 动态点云存在大量空像素区域, 帧间点数也往往不同. 因此, 对一定区域内的点集进行帧间运动估计时, 如何准确找到匹配的邻帧点集是一个难点. 假设对当前帧进行帧间预测时共分割为 N_B 个子点云块, 第 j 块子点云 cur_j 与其对应参考帧匹配块 ref_j 间存在几何与属性综合误差 $\Delta E_{cur_j,\,ref_j}$. 由于重构的预测帧实质上是通过组合相应的参考帧匹配块而估计得到的, 因此精准的帧间块匹配尝试最小化每个分割块的估计误差, 以提高预测帧整体预测精度, 如式 (3) 所示:

$$\min \Delta E_{cur,\,ref} = \min \frac{1}{N_B}\sum_{j=1}^{N_B}\Delta E_{cur_j,\,ref_j} \tag{3}$$

为了充分利用帧间相关性以降低时间冗余, 一些工作尝试对点云进行分割后寻找最佳匹配块以实现帧间预测. Mekuria 等[13] 将动态点云划分为若干个大小相同的宏块, 依据帧间块点数和颜色进行相似性判断, 对相似块使用迭代最近点算法计算刚性变换矩阵以实现帧间预测. 然而, 当区域分割得到的对应匹配块间存在较大偏差时, 预测效果不佳. 为了减少匹配块误差以提高预测精度, Xu 等[26] 提出使用 K 均值聚类将点云分为多个簇, 在几何上通过 ICP 实现运动预测, 在属性上则使用基于图傅里叶变换的模型进行运动矢量估计. 但基于 K 均值聚类的点云簇分割仅在预测帧中进行, 没有考虑帧间块运动相关性, 匹配精度提升受到限制. 为了进一步提高匹配精度, Santos 等[15] 受到二维视频编码器中 N 步搜索算法的启发, 提出了一种 3D-NSS 方法实现三维上的匹配块搜索, 将点云分割为 $N\times N\times N$ 的宏块后进行 3D-NSS 以搜索最优匹配块, 而后通过 ICP 进行帧间预测.

上述分割方法均实现了有效的块匹配, 但是, 基于宏块的均匀块分割与基于传统 K 均值聚类的块划分均没有考虑分割块间可能存在的运动连续性, 在分割上不够灵活. 具体表现为分割块过大无法保证块间匹配性, 过小又往往导致已经具有运动连续性的预测块被过度细化, 出现相同运动预测信息的冗余编码. 为了避免上述问题, 本文引入 KD 树思想, 提出了一种区域自适应分割算法, 该算法通过迭代进行逐层深入的二分类划分, 对各分割深度下块的运动性质与匹配程度进行分析, 确定是否需要继续分割以实现精准运动块匹配. 算法基本思想如图 4 所示, 若满足分割条件则继续进行二分类划分, 否则停止分割.

其中, 准确判断当前分割区域是否满足运动连续性条件下的帧间运动, 是避免过度分割以实现精准的运动块搜索的关键, 本文通过定义分割块匹配函数 $\Psi(l,\,n)$ 来确定截止深度, 如式 (4) 所示:

$$\Psi(l,\,n) = \rho(n)\cdot\xi(l) \tag{4}$$

其中, $\rho(n) = \max\left[\mathrm{sign}\left(n - N_D\right),\,0\right]$ 为点数判定函数, 当点数 n 大于最小分割块点数阈值 N_D 时, $\rho(n) = 1$, 表示满足深入分割的最小点数要求, 否则强制截止; $\xi(l)$ 为当前深度 l 下的块运动偏移度, 通过衡量匹配块间的运动变化分析是否需要进一步分割.

提出的 ξ 函数分别通过帧间质心偏移度 ξ_w 估计匹配块间运动幅度, 帧间包围盒偏移度 ξ_u 进行匹配块间几何运动一致性判定, 点数偏移度 ξ_n 进行点云分布密度验证, 最后通过 ξ_w、ξ_u 与 ξ_n 累加值与分割截止阈值 T 的比值来整体衡量当前块的运动程度与一致性. 即对于当前分割深度 l, $\xi(l)$ 可进一步细化为式 (5):

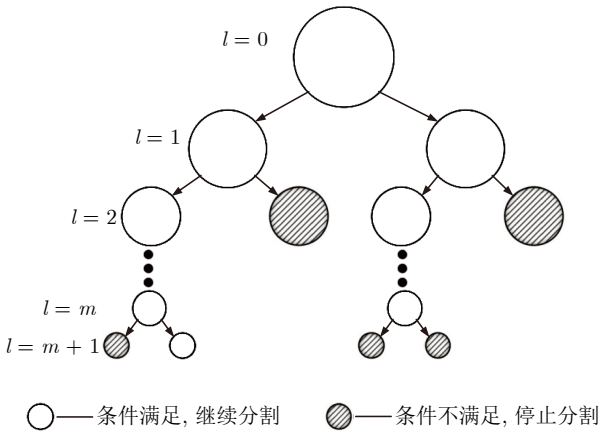


图 4　区域自适应分割块匹配方法示意图

Fig. 4　Schematic diagram of region adaptive segmentation based block matching method

$$\xi(l) = \frac{\xi_w(l) + \xi_u(l) + \xi_n(l)}{T} \tag{5}$$

其中,

$$\begin{cases} \xi_w(l) = \dfrac{\|w_{cur}(l) - w_{ref}(l)\|_2}{\|P_{\mathrm{Max}}(l) - P_{\mathrm{Min}}(l)\|_2} \\ \xi_u(l) = \dfrac{\|u_{cur}(l) - u_{ref}(l)\|_2}{u_{cur}(l)} \\ \xi_n(l) = \dfrac{\|n_{cur}(l) - n_{ref}(l)\|_2}{n_{cur}(l)} \end{cases}$$

并且, w_{cur}、w_{ref}、u_{cur}、u_{ref}、n_{cur} 与 n_{ref} 分别表示当前分割深度 l 下该区域与其前帧对应区域的质心、包围盒与点数, P_{Max} 和 P_{Min} 分别为当前块对角线对应点.

在 $\rho(n) = 1$ 的前提下, $\xi(l)$ 值反映当前 KD 树分割深度 l 下该区域点云的帧间运动情况. ξ 值越大帧间运动越显著, 当 ξ 值大于 1 时, 需对运动块进行帧间运动补偿, 如果继续分割将导致块的运动一致性被破坏或帧间对应块无法实现有效匹配, 从而导致帧间预测失败; ξ 值越小说明当前区域点云整体运动变化越小, 当 ξ 值小于 1 时, 需进一步分割寻找可能存在的运动区域.

对于需要进一步分割的点云块, 为了尽可能均匀分割以避免分割后匹配块间误差过大, 将待分割匹配块质心均值作为分割点, 通过以包围盒最长边作为分割面来确定 $l+1$ 深度下的分割轴 d, 分割轴

如式 (6) 所示:

$$d^* = \arg\max_{d=x,y,z} (Edge_{d,\max} - Edge_{d,\min}) \quad (6)$$

其中, $Edge_{d,\max}$ 和 $Edge_{d,\min}$ 分别为待分割块在 d 维度的最大值和最小值.

总结上文所述, 我们将提出的区域自适应分割的块匹配算法归纳为算法 1.

算法 1. 区域自适应分割的块匹配

输入. 当前帧点云 cur 与前帧参考点云 ref

输出. 当前帧与参考帧对应匹配块

1) **For** $j = 1$ to N_B **Do**
2) 初始化分割深度 $l = 0$;
3) **Do**
4) 选取待分割块 cur_j 和对应待匹配块 ref_j;
5) 计算质心 w、包围盒 u 与块点数 n;
6) 根据式 (5) 计算运动块偏移度 $\xi(l)$;
7) 根据函数 $\rho(n)$ 判定当前分割块点数;
8) 根据式 (4) 计算分割块匹配函数 $\Psi(l, n)$;
9) **If** $\Psi(l, n)$ 满足匹配块分割条件:
10) 根据式 (6) 确定分割轴 d;
11) 对 cur_j 与 ref_j 进行分割;
12) 保存分割结果;
13) 分割深度 $l + 1$;
14) **Else** $\Psi(l, n)$ 不满足匹配块分割条件:
15) 块分割截止;
16) 保存匹配块;
17) **End of if**
18) **While** 所有块均满足截止条件;
19) **End of for**

图 5 展示了本文提出的区域自适应分割的块匹配算法对帧 Longdress_0536 和其参考帧 Longdress_0535 进行分割后的块匹配结果. 在该序列当前帧下, 人物进行上半身的侧身动作. 观察图 5 可发现, 在运动变化较大的人物上半身, 算法在寻找到较大的对应匹配块后即不再分割; 而人物下半身运动平缓, 算法自适应提高分割深度以实现帧间匹配块的精确搜索, 因而下半身的分块数目大于上半身.

4 联合属性率失真优化的多模式帧间编码

在动态点云的帧间编码中, 常对相邻帧进行块分割或聚类后依据匹配块相似性实现帧间预测, 并利用补偿算法减少预测块误差以改善帧间编码质量. 其中迭代最近点算法常用于帧间运动估计中, 其通过迭代更新待配准点云 P 相较于目标点云 Q

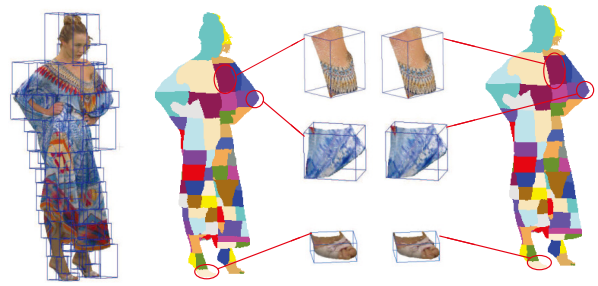


图 5 区域自适应分割的块匹配方法分割示例

Fig. 5 Example of block matching method based on adaptive regional segmentation

间的旋转矩阵 S 和平移向量 $\boldsymbol{t}$, 进而实现误差 $E(S, \boldsymbol{t})$ 最小化, 如式 (7) 所示:

$$(S^*, \boldsymbol{t}^*) = \underset{S,\boldsymbol{t}}{\operatorname{argmin}}\ E(S, \boldsymbol{t}) = \underset{S,\boldsymbol{t}}{\operatorname{argmin}} \sum_{i=1}^{N_p} \|Sp_i + \boldsymbol{t} - q_{i'}\|_2^2 \quad (7)$$

其中 N_p 为待配准点云点数, p_i 为待配准点云 P 的第 i 个点, $q_{i'}$ 为目标点云 Q 中与 p_i 相对应的点.

但是, 完全依据 ICP 配准进行动态点云的三维帧间预测存在两个问题: 首先, ICP 仅在预测块上逼近几何误差的最小化而没考虑到颜色属性偏差引起的匹配块差异, 影响了整体预测精度; 其次, 从率失真角度分析, 对运动变化极小的匹配块进行 ICP 配准实现的运动估计是非必要的, 该操作很难改善失真且会增加帧间编码比特消耗.

为改善上述问题, 提出了联合属性率失真优化的多模式帧间编码方法. 提出的方法首先在确保几何预测精度的同时, 充分考虑了可能的属性变化导致的预测精度下降问题, 而后通过率失真优化模型, 对块依据率失真代价函数得到的最优解进行分类后, 应用不同的编码策略以优化帧间编码方案, 旨在有限的码率约束下最小化编码失真, 即式 (8) 所示:

$$\min \sum_{j=1}^{N_B} D_j, \ \text{s.t.} \sum_{j=1}^{N_B} R_j < R_C \quad (8)$$

其中, R_j 和 D_j 分别表示第 j 个点云块的编码码率和对应的失真; N_B 是当前帧编码块总数; R_C 表示总码率预算. 引入拉格朗日乘子 λ, 式 (8) 所示的带约束优化问题可以转换为无约束的最优化问题, 即式 (9) 所示:

$$\min\{J\},\ J = \sum_{j=1}^{N_B} D_j + \lambda \sum_{j=1}^{N_B} R_j \quad (9)$$

其中, J 称作率失真代价函数, λ 称为拉格朗日乘子, 较小的 λ 使失真较小, 但会消耗较多的比特数, 反之则失真较大而消耗比特数较少.

具体到匹配块不同的编码方式选择上, 本文提出依据匹配块间差异程度分为 3 类方式编码.

首先, 第一类为完全近似块, 认为在帧间编码中可由前帧匹配块直接复用而几乎不影响重构精度, 因此在码率消耗上仅需要记录块位置信息, 即包围盒信息 $(X, Y, Z)_{\min}$ 与 $(X, Y, Z)_{\max}$ 共 6 个整型数据, 每个整型需要 16 bits 信息, 则包围盒位置信息编码所需总比特 R_{ASB} 如式 (10) 所示:

$$R_{\mathrm{ASB}} = (6 \times 16)\ \mathrm{bits} = 96\ \mathrm{bits} \tag{10}$$

其失真度 D_{ASB} 可表示为当前帧分割得到的第 j 块原始点云块与前帧重构得到的匹配块之间几何与属性误差累加值, 即式 (11) 所示:

$$D_{\mathrm{ASB}} = E^{G}_{cur_j^O, ref_j^R} + E^{A}_{cur_j^O, ref_j^R} \tag{11}$$

而后, 第二类为相对近似块, 认为是与前帧匹配块间在几何上出现某个方向、角度的运动或在颜色属性上由于光影变化出现偏移, 无法直接通过前帧匹配块进行复用的预测块. 相对近似块需通过 ICP 实现配准并进行属性补偿以减少块间差异度, 从而满足帧间预测的误差要求.

相对近似块在帧间编码中除了记录位置的包围盒信息, 还需要编码由 ICP 配准得到的 3×3 旋转矩阵 S、3×1 平移向量 $\boldsymbol{t}$ 以及块整体的属性补偿值 C, 其中旋转和平移的各个参数均需要 32 bits 浮点型数据, 属性补偿值的 (R, G, B) 分量各需要 16 bits 整型. 这类近似块进行帧间编码所需消耗的编码比特数统计如表 1 所示.

表 1　相对近似块帧间编码比特占用
Table 1　RSB inter coding bits occupancy

编码类型	编码信息	数据类型	比特占用 (bits)
位置信息	$(X, Y, Z)_{\min, \max}$	Int	96
旋转矩阵	3×3 矩阵	Float	288
平移向量	3×1 向量	Float	96
属性偏移	(R, G, B)	Int	48
总编码消耗	R_{RSB}	Int、Float	528

因此相对近似块编码码率 R_{RSB} 如式 (12) 所示:

$$R_{\mathrm{RSB}} = 528\ \mathrm{bits} \tag{12}$$

其失真度 D_{RSB} 可表示为经过 ICP 配准与补偿后 cur_j^O 与 ref_j^R 之间几何与属性误差累加值, 即式 (13) 所示:

$$D_{\mathrm{RSB}} = E'^{G}_{cur_j^O, ref_j^R}(S, \boldsymbol{t}) + E'^{A}_{cur_j^O, ref_j^R}(C) \tag{13}$$

其中, $E'^{G}_{cur_j^O, ref_j^R}(S, \boldsymbol{t})$ 为经过配准后的当前块点云与重构得到的前帧参考块点云间的几何误差, S、$\boldsymbol{t}$ 分别为配准计算中得到的旋转矩阵与平移向量, $E'^{A}_{cur_j^O, ref_j^R}(C)$ 为经过属性补偿后当前块点云与重构得到的前帧参考块点云间属性误差, C 为属性补偿值.

具体的, 对于需要进行配准与补偿的相对近似块, 我们的方案提出联合几何与属性两个角度改善其预测误差. 在几何上, 通过式 (7) 的 ICP 配准算法计算相对近似块与其前帧匹配块的最优解 $(S^*, \boldsymbol{t}^*)$ 以获得几何预测误差, 如式 (14) 所示:

$$E'^{G}_{cur_b^O, ref_b^R}(S, \boldsymbol{t}) = \frac{1}{n}\sum_{i=1}^{n}\left\| S^* \cdot cur_j^{O,G}(i) + \boldsymbol{t}^* - ref_j^{R,G}(i') \right\|_2^2 \tag{14}$$

其中, n 为相对近似块的点数, $cur_j^{O,G}(i)$ 为相对近似块中第 i 点的几何坐标, $ref_j^{R,G}(i')$ 则为匹配块中与该点相对应的第 i' 点的几何坐标.

属性上, 通过计算相对近似块与对应匹配块的属性均值, 对整体属性偏移量 C 进行补偿以改善属性误差.

具体的, 属性偏移量 C 由当前帧待补偿的第 j 块点云 cur_j^O 与前帧重构得到的匹配块 ref_j^R 各对应点属性之差的均值定义, 属性偏移量计算与改善后的属性误差分别如式 (15)、式 (16) 所示:

$$C = \frac{1}{n}\sum_{i=1}^{n}\left[cur_j^{O,A}(i) - ref_j^{R,A}(i') \right] \tag{15}$$

$$E'^{A}_{cur_j^O, ref_j^R}(C) = \frac{1}{n}\sum_{i=1}^{n}\left\| cur_j^{O,A}(i) - ref_j^{R,A}(i') - C \right\|_2^2 \tag{16}$$

其中, n 为当前块点数, $cur_j^{O,A}(i)$、$ref_j^{R,A}(i')$ 分别为当前块的第 i 点与前帧相应匹配块对应的第 i' 点属性值.

最后, 第三类为非近似块, 认为与前帧匹配块在几何上的运动或在颜色属性的变化较为显著, 通过 ICP 配准与属性补偿后依旧无法满足帧间编码的误差要求, 而需要进行帧内编码的块. 帧内编码通过 V-PCC 编码器进行, 其编码码率 R_{NSB} 是由当前块点数 n 与 V-PCC 帧内编码中对应的每点编码比特消耗 (Bits per point, Bpp) 所决定, 如式 (17)

所示:

$$R_{\mathrm{NSB}} = n \times Bpp \tag{17}$$

其中, n 大于最小块分割点数阈值 N_D.

在失真度量上, 由于非近似块采用 V-PCC 帧内编码, 失真 D_{NSB} 即为 V-PCC 编解码中固有的帧内失真, 如式 (18) 所示:

$$D_{\mathrm{NSB}} = E'^{G}_{cur_j^O,\, cur_j^R} + E'^{A}_{cur_j^O,\, cur_j^R} \tag{18}$$

其中, cur_j^R 为解码得到的第 j 个帧内重构块, $E'^{G}_{cur_j^O,\, cur_j^R}$ 与 $E'^{A}_{cur_j'^O,\, cur_j^R}$ 分别表示为当前分割块与其帧内解码重构块的几何与属性误差.

鉴于自适应分割得到的分割块在不同编码方式中的码率 R 和失真 D 不同, 由式 (9) 计算的整体率失真代价 J 也必然存在差异, 进而通过对比 ASB、RSB 与 NSB 三种模式的 J 可确定最优的编码方式. 又由于动态点云中帧内编码为帧间预测提供了至关重要的参考帧, 其失真度 D_{NSB} 必然需小于帧间编码的 D_{ASB} 和 D_{RSB} 的失真度. 进而由参考文献 [21−22] 的实验结果可以推断, V-PCC 在不同的量化等级 (R1 ~ R5) 下, 每点需要消耗的最小编码比特约为 0.13 bits 至 1.15 bits 不等, 因此通过限定分割块的最小块点数 N_D, 使帧内编码块的码率 R_{NSB} 始终大于式 (10)、式 (12) 所估算的两类帧间预测方式, 确保帧间编码不会出现码率高于帧内编码的情形. 其中 N_D 的取值将在第 5.1 节评价指标与参数配置中给出.

鉴于上述三类分割块的 R-D 分析, 三类不同编码块的率失真分布可分为两种情形, 如图 6 中黑实折线与虚折线所示: 第一种情形为 ICP 配准与补偿能够改善匹配块的帧间预测误差, 此时完全近似块的帧间编码具有最小的码率和最大的失真, 相对近似块的帧间编码的码率和失真居中, 非近似块帧内编码的失真最小但消耗最大码率; 第二种情形则是由于帧间变化很小或 ICP 算法配准不能减小帧间预测误差, 导致完全近似块编码与相对近似块编码具有相同的失真, 但完全近似块编码中码率消耗较少, 而非近似块编码由于分割中块点数的限制, 在失真最小的情况下依旧需要最高的码率消耗.

为了获得上述帧间率失真优化模型的拉格朗日乘子 λ, 借鉴文献 [15] 的方法, 通过计算 ASB-RSB 和 RSB-NSB 两条折线中点连线的负斜率来估算 λ, 图 6 中灰虚线对 λ 的计算方式进行了直观的图形表达, 具体计算如式 (19) 所示:

$$\lambda = \frac{D_{\mathrm{ASB}} - D_{\mathrm{NSB}}}{R_{\mathrm{NSB}} - R_{\mathrm{ASB}}} \tag{19}$$

结合式 (9) ~ 式 (19), 可对当前待编码块进行率失真分析以确定最终编码方式, 从而实现最优化编码, 即在确定 R、D 与 λ 后取代价函数 J 最小值的编码方式进行块编码, 如式 (20) 所示:

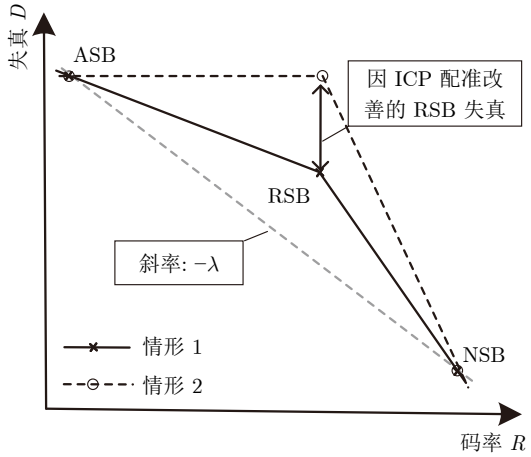


图 6 率失真折线与 λ 示意图

Fig. 6 Rate distortion polyline and λ diagram

$$J^* = \arg\min\left(J_{\mathrm{ASB}},\, J_{\mathrm{RSB}},\, J_{\mathrm{NSB}}\right) \tag{20}$$

其中, 当率失真曲线满足情形 2 时, 仅需进行一次比较, 当 $J_{\mathrm{NSB}} < J_{\mathrm{ASB}}$ 时定义为非近似块, 采用 V-PCC 帧内编码, 否则定义为完全近似块, 仅需对匹配块的位置信息进行帧间编码; 而满足情形 1 时则需进行两次比较, 当 $J_{\mathrm{NSB}} < J_{\mathrm{RSB}}$ 且 $J_{\mathrm{NSB}} < J_{\mathrm{ASB}}$ 时, 进行 V-PCC 帧内编码, 否则, 采用帧间方式编码; 而后, 若 $J_{\mathrm{RSB}} > J_{\mathrm{ASB}}$, 则定义为完全近似块, 对其匹配块位置信息进行帧间编码, 否则定义为相对近似块, 进行配准、补偿与位置信息的帧间编码.

根据上文描述, 可以总结如算法 2 所述:

算法 2. 联合属性率失真优化的多模式帧间编码

输入. 当前帧 N_B 个分割块与参考帧对应匹配块

输出. ASB 与 RSB 预测信息、NSB 融合点云编码码流

1) **For** $j = 1$ to N_B **Do**
2) 根据式 (11) 计算 cur_j 的失真 D_{ASB};
3) 根据式 (7)、式 (15) 计算 S、$\boldsymbol{t}$ 与 C;
4) 根据式 (13)、式 (14) 和式 (16) 计算 cur_j 的失真 D_{RSB};
5) 根据式 (17) 计算 R_{NSB};
6) 根据式 (18) 计算 cur_j 的失真 D_{NSB};
7) 根据式 (19) 计算 λ;
8) 根据式 (20) 计算最佳编码模式代价 J^*;
9) **If** $J^* = J_{\mathrm{ASB}}$
10) ASB 位置信息熵编码;
11) **Else if** $J^* = J_{\mathrm{RSB}}$
12) RSB 位置、预测与补偿信息熵编码;
13) **Else if** $J^* = J_{\mathrm{NSB}}$

14)　NSB 融合并 V-PCC 帧内编码;

15)　**End of if**

16) **End of for**

5　实验分析

5.1　评价指标与参数配置

根据 V-PCC 通用测试条件 (Common test condition, CTC)[27] 的测试要求, 测试的动态点云序列包括 Soldier、Queen、Longdress、Loot 和 Redandblack 5 组. 此外, 几何和颜色属性的客观重构质量通过计算原始点云与解码重构点云间点到点和点到面的峰值信噪比 PSNR (D1、D2), 亮度 (Luma) 以及色度 (Cb、Cr) 分量的 PSNR 来体现. 具体的, PSNR 通过对点云坐标或颜色属性的最大值 p 与原始点云 O、重构点云 R 间均方误差的比值取对数后进行表示, 如式 (21) 和式 (22) 所示.

$$PSNR_G = 10\lg\frac{3p^2}{E^G_{O,R}} \tag{21}$$

$$PSNR_A = 10\lg\frac{p^2}{E^A_{O,R}} \tag{22}$$

其中, $PSNR_G$ 和 $PSNR_A$ 以及 $E^G_{O,R}$ 与 $E^A_{O,R}$ 分别为原始点云与重构点云在几何与属性的峰值信噪比和均方误差.

另外, V-PCC 定义了从 R1 至 R5 这 5 个量化等级进行不同码率下的编码性能测试, 编码所需比特消耗和重构质量随量化等级递增. 与 V-PCC 中定义的 5 个量化等级相对应, 在提出的算法中, 需对编码模式判定阈值 Ω、分割块最小点数 N_D 与分割截止阈值 T 3 个参数做对应调整, 配置情况分析如下:

1) 低码率下: 对分割块的帧内编码所需比特较少, 帧间编码中块位置索引与预测和补偿信息的编码比特占总编码比特的比例较高, 并且低码率下帧内编码本身的重构精度较差, 使得预测精度要求相对降低, 所以提高块最小块点数 N_D 与编码模式判定阈值 Ω, 降低分割截止阈值 T, 以减少分割块数和预测信息编码码流.

2) 高码率下: 与低码率相反, 对分割块帧内编码所需比特显著提高, 而帧间编码比特占总比特的比例很小, 并且帧内编码重构精度很高的情况下对预测精度要求也较高, 所以需降低最小块点数 N_D 与编码模式判定阈值 Ω, 提高分割截止阈值 T 以实现更细化的块分割来提高预测补偿精度.

在经过实验总结后, 定义从 R1 到 R5 这 5 个量化等级下相对应的 Ω、N_D 与 T 3 个参数的具体取值如表 2 所示.

表 2　提出的改进算法量化参数

Table 2　Quantization parameters of proposed improved algorithm

量化参数	编码模式判定 Ω	最小块阈值 N_D	分割截止阈值 T
R1	3.5	14000	2.0
R2	3.0	12000	2.2
R3	2.5	10000	2.5
R4	2.0	8000	2.8
R5	1.5	6000	3.0

在硬件配置上, 所有实验基于华硕主板, 英特尔 i5-10400 型处理器、8 GB 内存的台式机, 运行环境为 Windows10 操作系统和 VisualStudio2017. 图 7 展示了实验中需要测试的 5 个典型的 8i 动态点云序列示例.

Longdress　Soldier　Queen　Redandblack　Loot

图 7　8i 动态点云序列示例

Fig. 7　Example of 8i dynamic point cloud sequence

5.2　消融实验

5.2.1　区域自适应分割的块匹配

为验证本文提出的算法的优越性, 分别以均匀块分割实现的块匹配方法和提出的方法进行对照, 以文献 [13] 提出的基于误差与 ICP 配准的帧间预测方法作为共同的帧间预测方式, 在相同测试环境下, 分别对图 7 所示的 5 个测试序列进行 50 帧测试实验, 并使用基于 V-PCC 的标准测试软件 TMC2-v14.0[28] 进行 BD-BR 增益对比, 最终结果如表 3 所示.

通过表 3 可以观察到, 整体上, 相较于 V-PCC 锚点, 在测试的 5 个动态点云序列中, 两种分割算法均实现了一定的 BD-BR 增益. 在两种分割算法的对比中, 区域自适应分割相较于均匀块分割在几何的帧间压缩中具有明显优越性, 几何分量 D1、D2 分别实现了 −4.91%、−6.38% 的相对 BD-BR

表 3 区域自适应分割的块匹配方法性能测试

Table 3 Performance test of block matching method based on adaptive regional segmentation

	Name	D1 (%)	D2 (%)	Luma (%)	Cb (%)	Cr (%)
均匀块分割	Loot	33.93	49.67	4.48	−18.07	−15.08
	Redandblack	13.51	19.42	13.41	−3.01	10.79
	Soldier	−34.75	−34.66	−40.82	−42.91	−44.52
	Queen	−33.64	−33.33	−18.33	−28.58	−12.74
	Longdress	11.97	13.15	27.48	−4.88	11.13
	Average	−1.80	−2.85	−2.76	−18.51	−10.08
改进的自适应块分割	Loot	27.10	39.28	2.22	−24.14	−21.19
	Redandblack	6.03	8.14	16.06	4.14	13.72
	Soldier	−37.69	−37.57	−42.87	−45.10	−46.42
	Queen	−35.72	−33.94	−9.36	−12.93	−14.92
	Longdress	6.74	6.46	4.83	5.02	12.49
	Average	−6.71	−3.53	−5.82	−14.60	−11.26
相对 BD-BR 增益		−4.91	−6.38	−3.06	3.91	−1.18

增益, 亮度分量 Luma、色度分量 Cr 分别实现了 −3.06%、−1.18% 的相对 BD-BR 增益, 但色度分量 Cb 却出现了 BD-BR 增加的情况.

具体到单个测试序列, 本文提出的分割算法在几何分量 D1、D2 上均实现了负增益, 但存在少部分色度分量正增益的情况. 其中 Longdress、Loot 与 Redandblack 序列几何分量改善较为显著, 均实现了 −6% 以上的 D1、D2 的 BD-BR 增益. 原因在于, 此类序列人物服饰纹理丰富、动作变化频繁且无固定方向运动, 因此均匀块分割算法针对该类点云的帧间运动预测困难, 而本文提出区域自适应分割的块匹配方法, 可对分割区域与深度进行自适应判别, 通过待分割块在几何与属性上的联合匹配度判定当前块间误差, 合理调节分割深度以找到最佳匹配的块, 提高了匹配块预测精度, 进而改善了整体帧间预测性能.

5.2.2 联合属性率失真优化的多模式帧间编码

为验证提出的基于联合属性率失真优化的多模式帧间编码方法的优越性, 针对所有匹配块进行的完全帧间运动预测, 同提出的基于联合属性率失真优化模型的帧间预测方法进行对照, 并以均匀块分割算法作为匹配块的共同分割方法, 在相同测试环境下, 分别对图 7 所示的 5 个测试序列进行 50 帧测试实验, 并使用 V-PCC 标准测试软件 TMC2-v14.0 进行 BD-BR 增益对比, 最终结果如表 4所示.

通过表 4 可以观察到, 整体上, 相较于 V-PCC 锚点, 在测试的 5 个动态点云序列中, 两种预测算法均实现了一定的性能改善. 在两种算法的对比中, 基于联合属性率失真优化的帧间预测相较于针对所有匹配块进行的完全帧间运动预测, 在几何的帧间压缩中具有显著的 BD-BR 负增益, 其中几何分量 D1、D2 分别实现了 −17.77%、−19.41% 的相对 BD-BR 增益, 另外, 亮度分量 Luma 也有 −6.80% 的 BD-BR 增益. 但色度分量 Cb 和 Cr, 出现了 BD-BR 增加的情况.

具体到单个测试序列, 提出的帧间预测算法的 BD-BR 负增益主要体现在几何分量 D1、D2 与亮度分量 Luma 上, 在色度分量中, 仅在 Queen 测试序列中实现了显著的性能改善. 具体的, 在 Longdress、Loot 与 Redandblack 3 个帧间运动预测困难的序列中, 提出的方法相对于完全帧间运动预测具有明显优势, 均实现了 D1、D2两个几何分量 −10% 以上相对 BD-BR 增益. 原因在于提出的帧间预测方法, 通过对匹配块进行基于率失真优化模型的分类以应用多模式编码方法, 可以在确保质量的同时, 减少几何预测与属性补偿的非必要编码以降低整体比特消耗, 进而有效提高了帧间预测性能.

5.2.3 相互消融实验分析

为验证本文提出的两个创新点间的耦合关系, 本节在整体框架基础上, 分别将仅使用自适应分割的块匹配方法与仅使用多模式帧间运动预测方法以及完整算法框架进行综合比对和性能分析, 实验数据集选取自图 7 所示的 5 个测试序列. 表 5 给出了以 V-PCC 为参考基准, 应用不同编码方法的 BD-BR

表 4　联合属性率失真优化的多模式帧间编码性能测试

Table 4　Performance test of multi-mode inter-frame coding based on joint attribute rate distortion optimization

	Name	D1 (%)	D2 (%)	Luma (%)	Cb (%)	Cr (%)
完全帧间运动预测	Loot	32.31	42.75	11.16	−19.63	−16.09
	Redandblack	14.16	15.52	7.88	−11.11	6.35
	Soldier	−31.21	−31.21	−37.47	−41.07	−42.48
	Queen	−29.34	−30.23	−12.47	−17.84	−6.77
	Longdress	12.28	14.79	17.74	−13.26	0.79
	Average	−0.36	2.32	−2.63	−20.58	−13.70
改进的多模式帧间运动预测	Loot	−0.06	0.70	10.07	5.44	5.80
	Redandblack	−3.31	−2.09	11.73	1.66	9.45
	Soldier	−35.88	−34.59	−36.31	−36.28	−38.70
	Queen	−39.55	−39.37	−39.16	−44.46	−44.97
	Longdress	−0.64	1.28	6.50	3.10	10.23
	Average	−18.13	−17.09	−9.43	−16.06	−11.64
相对 BD-BR 增益		−17.77	−19.41	−6.80	4.52	2.06

表 5　相较于 V-PCC 相互消融性能 (平均 BD-BR)

Table 5　Mutual ablation performance compared to V-PCC (Average BD-BR)

应用算法	D1 (%)	D2 (%)	Luma (%)	Cb (%)	Cr (%)
多模式帧间编码	−18.13	−17.09	−9.43	−16.06	−11.64
自适应块分割	−6.71	−3.53	−5.82	−14.60	−11.26
完整算法框架	−22.57	−20.94	−22.01	−23.67	−21.90

增益情况.

由表 5 可知, 相较于 V-PCC 锚点, 多模式帧间编码与自适应块分割算法整体上均实现了 BD-BR 负增益, 但均不及两者同时应用的完整算法框架下的测试结果. 从提出的两个创新点的相互消融实验结果可以发现, 多模式帧间编码在各分量上的性能改善均高于自适应块分割算法, 尤其在几何分量上最为突出, 原因在于自适应块分割实现的匹配块搜索有效改善了匹配块间基于 ICP 配准的失真, 而属性分量的 BD-BR 负增益则主要源于帧间编码所减少的码流消耗, 因此区别较小.

为了更直观地分析本文提出的自适应块分割算法与多模式帧间编码方法在不同码率时的性能差异, 图 8 展示了 Queen 和 Soldier 序列在 R1 至 R5 这 5 个量化等级下, 分别应用自适应块分割方法、多模式帧间编码方法以及应用完整算法框架时, 相对于 V-PCC 锚点在几何分量 D1 与属性分量 Luma 上的 BD-BR 变化趋势.

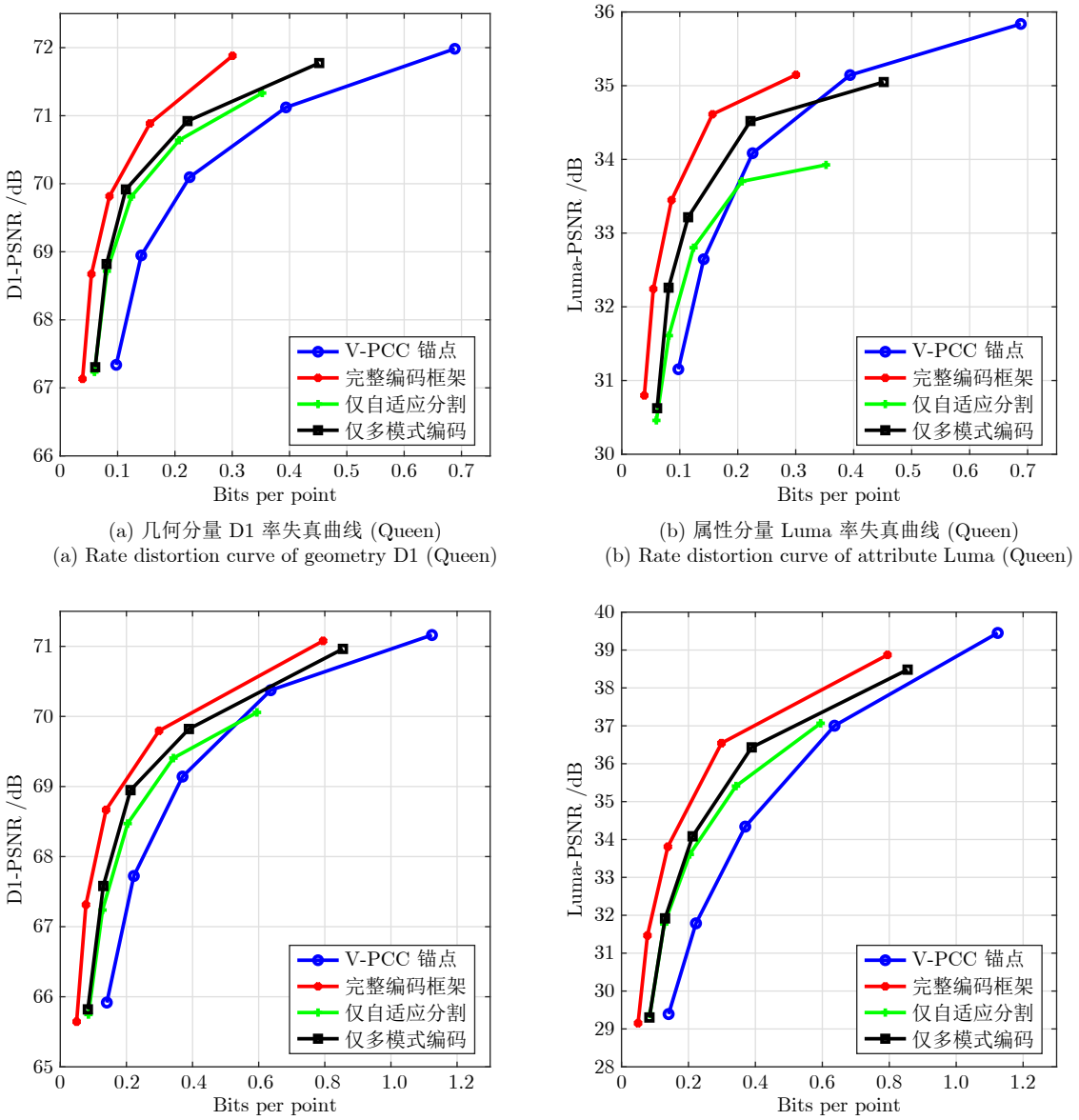
观察图 8 对 Queen 以及 Soldier 数据集的相互消融实验所绘制的率失真曲线可发现, 其整体趋势与表 5 基本一致, 完整算法框架整体性能始终高于仅使用自适应块分割算法与仅使用多模式帧间编码算法的情形. 具体的, 随着码率增大, PSNR 亦随之提高, 其中低码率下自适应分割算法与多模式帧间编码对编码性能的贡献差距较小, 但随着码率的提升, 多模式帧间编码的贡献逐渐提高, 原因在于高码率下帧间匹配块编码失真要求较高, 多模式编码可有效减少帧间编码误差, 因此取得了相较于仅使用自适应分割方法更高的 PSNR.

5.3　综合性能

5.3.1　率失真性能分析

由第 5.2 节消融实验可以看出, 本文提出的区域自适应分割的块匹配方法与联合属性预测率失真优化多模式帧间编码方法, 相较于传统帧间块分割与预测方法, 均降低了 BD-BR. 为验证算法的综合性能, 以基于 V-PCC 标准的最新测试软件 TMC2-v14.0 为锚点, 以第 5.1 节中设置的实验配置与编码参数作为测试条件, 针对 5 个动态点云序列的平均 BD-BR 和编码耗时情况进行实验, 结果见表 6.

由表 6 可知, 提出的方法与 V-PCC 测试软件 TMC2-v14.0 相比, 融合了自适应块匹配和联合属性率失真优化的多模式帧间编码方法使得帧间预测更高效, 实现了性能改善. 具体的, 在几何性能测评中 D1 与 D2 的 BD-BR 分别节省了 22.57% 与 20.94%; 在属性评测中, Luma、Cb 和 Cr 的改善更为明显, 分



(a) 几何分量 D1 率失真曲线 (Queen)
(a) Rate distortion curve of geometry D1 (Queen)

(b) 属性分量 Luma 率失真曲线 (Queen)
(b) Rate distortion curve of attribute Luma (Queen)

(c) 几何分量 D1 率失真曲线 (Soldier)
(c) Rate distortion curve of geometry D1 (Soldier)

(d) 属性分量 Luma 率失真曲线 (Soldier)
(d) Rate distortion curve of attribute Luma (Soldier)

图 8 消融实验率失真曲线图

Fig. 8 Rate distortion curve of ablation experiment

表 6 提出的改进算法相较于 V-PCC 的性能比较

Table 6 Performance of proposed improved algorithm compared with V-PCC

Name	D1 (%)	D2 (%)	Luma (%)	Cb (%)	Cr (%)
Loot	−2.64	−0.69	−3.57	−4.74	−4.85
Redandblack	−1.65	−0.95	−1.75	−2.01	−2.12
Soldier	−51.53	−46.99	−54.12	−55.65	−55.95
Queen	−56.58	−56.45	−51.62	−55.12	−46.27
Longdress	−0.44	0.40	0.99	−0.80	−0.30
Overall	−22.57	−20.94	−22.01	−23.67	−21.90

别达到了 22.01%、23.67%、21.90% 的 BD-BR 降低.

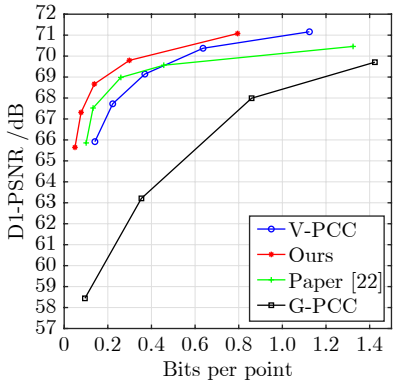
对表 6 各测试序列进行单独分析可发现, 本文提出的方法对于 Soldier 与 Queen 两序列的性能增益更为明显, 平均约有 50% 的 BD-BR 降低; 而在 Redandblack 与 Loot 两序列上则性能提升有限, 平均实现了约 −3% 的几何与属性 BD-BR 增益; 在 Longdress 数据集上则仅在几何分量 D1 与属性分量 Cb、Cr 上略微改善, 在几何分量 D2 与属性分量 Luma 上则出现 BD-BR 增加的情况. 针对上述结果, 对 5 个测试序列进行可视化分析可知, Soldier 与 Queen 序列帧间变化较小, 在帧间编码中

相对近似块与完全近似块的占比较高, 因此在帧间编码中具有更高增益; 而 Loot, Red and black 与 Long dress 序列的帧间运动变化相对较大, 在帧间编码中因匹配块差异过大导致帧间预测效果较差, 使得帧间编码性能增益比较有限.

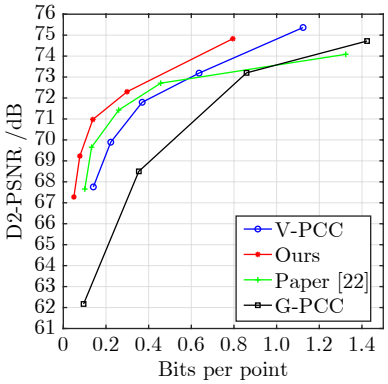
为了更直观地展示本文算法的率失真性能, 图 9 以 Soldier 序列为例, 展示了 R1 至 R5 这 5 个量化等级下, 本文提出的算法相较于 V-PCC 锚点、文献 [22] 提出的帧间改进算法以及 MPEG 制定的基于几何的点云压缩 (Geometry Point Cloud Compression, G-PCC)[26] 压缩算法在几何和颜色属性的 PSNR 随码率变化的曲线. 横轴为平均每点比特消耗, 纵轴为几何与属性各指标 PSNR, 两者均随量化级数的提高呈增长趋势. 从图中可得出以下结论: 首先, 与 V-PCC 相比, 受益于精准的块匹配与以率失真优化为目的进行的分类帧间编码, 提出的帧间算法在不同的码率下均取得了明显的 PSNR增益; 其次, 在低码率配置下 (R1 ~ R3), 帧间算法性能改善比高码率配置下 (R4、R5) 的性能改善更为明显, 原因在于同样减少了码率的情况下, 较低码率下的整体失真较大, 帧间编码额外引入的失真对整体影响反而较小; 再者, 受益于合理的运动块搜索和帧间自适应编码, 整体 BD-BR 性能同样优于文献 [22] 的帧间预测算法和 MPEG 的 G-PCC 压缩标准.

为更全面地比较帧间编码性能, 选取具有相同测试条件下的动态点云三维帧间改进方法[15, 21−22] 进行对比, 结果如表 7 所示.

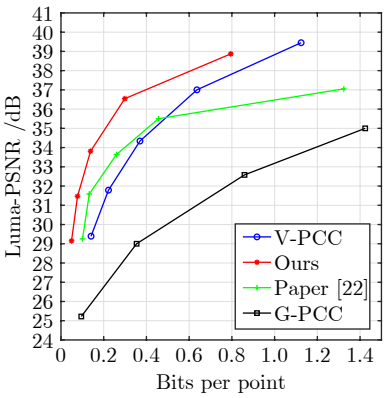
观察表 7 可以发现, 提出的方法在几何与属性上的增益均明显优于文献 [15, 21−22] 提出的方法. 原因在于 Santos 等[15] 提出的三维 N 步匹配块搜索提高了帧间分割块匹配精度, 帧间编码性能有了明显改善, 但没有考虑均匀块分割对预测块的运动连续性的破坏, 导致具有相同运动信息分割块帧间编码的冗余; Li 等[21] 提出的三维帧间估计是通过二维



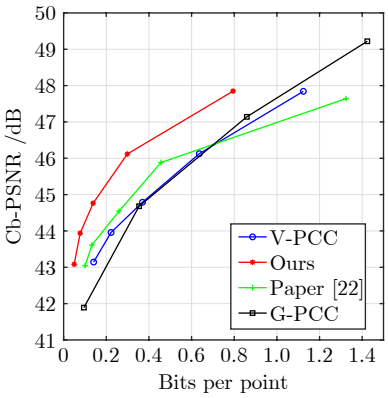
(a) 几何分量 D1 率失真曲线 (Soldier)
(a) Rate distortion curve of geometry Dl (Soldier)



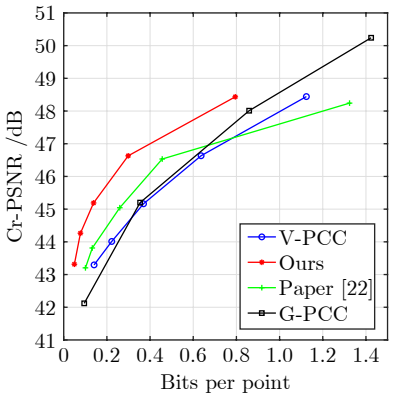
(b) 几何分量 D2 率失真曲线 (Soldier)
(b) Rate distortion curve of geometry D2 (Soldier)



(c) 属性分量 Luma 率失真曲线 (Soldier)
(c) Rate distortion curve of attribute Luma (Soldier)



(d) 属性分量 Cb 率失真曲线 (Soldier)
(d) Rate distortion curve of attribute Cb (Soldier)



(e) 属性分量 Cr 率失真曲线 (Soldier)
(e) Rate distortion curve of attribute Cr (Soldier)

图 9　Soldier 序列率失真曲线图

Fig. 9　Rate distortion curve of Soldier sequence

表 7 提出的改进算法同文献 [15, 21−22] 的性能对比
Table 7 Performance comparisons between the proposed improved algorithm and references [15, 21−22]

Name	D1 (%)	D2 (%)	Luma (%)	Cb (%)	Cr (%)
文献 [15]	−21.02	−15.68	−3.54	—	—
文献 [21]	−4.70	−4.70	−9.70	−16.30	−14.50
文献 [22]	−14.80	−19.10	−2.20	−16.20	−19.90
提出的算法	−22.57	−20.94	−22.01	−23.67	−21.90

上分割块的位置与运动信息建立, 由二维到三维的运动模型来实现的, 因此需要对二维视频编码器进行适用于三维运动估计的更改, 且不能完全反映真实的三维运动; Kim 等[22] 通过计算帧间误差确定预测帧后依据前后帧点云的残差编码实现帧间运动估计, 但对于运动频繁的动态点云的残差编码需要消耗大量的比特使得性能提升受到限制. 相较于上述方法, 本文提出的通过区域自适应分割的块匹配方法能够对匹配块的三维运动进行更精确的预测, 并且, 基于率失真优化模型进行的选择性预测和补偿方法进一步减少了预测编码的比特消耗, 因此有更高的三维帧间压缩性能.

5.3.2 编码开销与复杂度分析

在本节中, 将进一步对提出的算法在编码比特开销与时间复杂度上的表现展开分析. 其中, 提出的改进算法的编码开销体现为 3 个部分, 分别为关键帧的 V-PCC 自身帧内编码比特开销、预测帧的非近似块的 V-PCC 编码比特开销、以及预测帧进行预测块定位、几何与属性补偿中需引入的部分帧间编码比特开销. 以 Soldier 和 Queen 数据集为例, 在通用测试条件下测得各测试序列的平均编码开销和编码用时, 并同 V-PCC 的帧内编码进行编码开销与相对编码用时的对比, 最终结果如表 8所示.

观察表 8, 提出的视频点云编码算法在 Soldier 和 Queen 数据集中, 码流消耗在各量化参数下均相较于 V-PCC 编码实现了明显减少, 平均开销仅为 V-PCC 的 49.56%, 原因在于提出的基于 V-PCC 改进帧间编码方法, 通过对预测帧点云进行合理的块分割并分类编码, 在尽可能保证编码质量的前提下减少了直接帧内编码的点云点数, 降低了总体码率消耗, 并且, 受益于编码点数的减少, 整体编码时间实现了平均 14.45% 的降低. 具体到不同量化等级, 随着量化等级的提升, 预测编码码流亦随之提高, 这是为了尽可能提高预测精度, 较高的量化等级下分割块总数提升也较高, 因而帧间预测编码中比特消耗和编码用时的改善不及低量化等级.

表 8 相较于 V-PCC 的编码开销与时间复杂度
Table 8 Encoding overhead and time complexity compared to V-PCC

	量化等级	预测编码 (bits)	NSB 编码 (bits)	V-PCC 帧内编码 (bits)	相对编码开销 (%)	相对编码时间 (%)
Queen	R1	5722	33065	97363	39.83	85.01
	R2	6099	48154	140968	38.49	84.74
	R3	6841	78773	225197	38.02	84.99
	R4	6797	149308	392872	39.71	85.77
	R5	7494	292382	686758	43.67	92.08
Soldier	R1	4986	47469	149301	35.13	82.17
	R2	5692	76517	236022	34.83	81.79
	R3	7986	138810	392283	37.42	81.27
	R4	11836	305130	675218	46.94	83.38
	R5	16182	826968	1192585	70.69	94.29
Average		7955	199658	418857	49.56	85.55

6 总结

本文主要针对 V-PCC 在动态点云压缩中从三维到二维的投影使得三维帧间运动的相关性被破坏, 降低了帧间编码性能的问题, 提出了一种针对动态点云的三维帧间预测算法, 并设计了一个新型的动态点云帧间编码框架. 为了兼顾分割块的运动相关性的同时避免过度分割, 提出了区域自适应分割的块匹配方法, 以实现更精准的帧间运动估计; 为了确保压缩效率的前提下最大限度减少比特消耗, 提出了联合属性率失真优化模型的多模式帧间编码方法. 实验表明, 本文提出的方法有效利用了时间冗余, 改善了动态点云的三维帧间压缩性能, 总体上相较于 V-PCC 取得了 −22.57% 的 BD-BR 增益, 并与其他帧间改进算法的对比中均有不同程度的性能提升. 但需要说明的是, 本文的方法在面对帧间运动剧烈且纹理丰富的点云数据时帧间预测效果较为有限, 原因在于基于 ICP 的帧间配准只是在匹配块基础上进行刚性变换, 难以对具有复杂纹理的块进行帧间预测, 后续研究工作中, 我们将针对这类点云引入更合适的帧间预测算法.

References

1 Li Hou-Qiang, Li Li, Li Zhu. A review of point cloud compression. *ZTE Technology Journal*, 2021, **27**(1): 5−9
(李厚强, 李礼, 李竹. 点云编码综述. 中兴通讯技术, 2021, **27**(1): 5−9)

2 Ishikawa Y, Hachiuma R, Ienaga N, Kuno W, Sugiura Y, Saito H. Semantic segmentation of 3D point cloud to virtually manipulate real living space. In: Proceedings of the Asia Pacific Work-

shop on Mixed and Augmented Reality. Ikoma, Japan: IEEE, 2019. 1–7

3 Deng X M, Zhu Y Y, Zhang Y D, Cui Z P, Tan P, Qu W, et al. Weakly supervised learning for single depth-based hand shape recovery. *IEEE Transactions on Image Processing*, 2021, **30**: 532–545

4 Tian Yong-Lin, Shen Yu, Li Qiang, Wang Fei-Yue. Parallel point clouds: Point clouds generation and 3D model evolution via virtual-real interaction. *Acta Automatica Sinica*, 2020, **46**(12): 2572–2582
(田永林, 沈宇, 李强, 王飞跃. 平行点云: 虚实互动的点云生成与三维模型进化方法. 自动化学报, 2020, **46**(12): 2572–2582)

5 Sun P P, Zhao X M, Xu Z G, Wang R M, Min H G. A 3D LiDAR data-based dedicated road boundary detection algorithm for autonomous vehicles. *IEEE Access*, 2019, **7**: 29623–29638

6 Huang X Y, Wang P, Cheng X J, Zhou D F, Geng Q C, Yang R G. The apolloscape open dataset for autonomous driving and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, **42**(10): 2702–2719

7 Gan Z P, Xu H R, He Y R, Cao W, Chen G H. Autonomous landing point retrieval algorithm for UAVs based on 3D environment perception. In: Proceedings of the IEEE 7th International Conference on Virtual Reality. Foshan, China: IEEE, 2021. 104–108

8 Cao Cheng-Kun, Zhang Cong-Yi, Wang Guo-Ping. Precision controllable point clouds compression using global similarity in dictionary. *Journal of Computer-Aided Design & Computer Graphics*, 2019, **31**(6): 869–877
(曹成坤, 张琮毅, 汪国平. 精度可控的字典全局相似性点云压缩. 计算机辅助设计与图形学学报, 2019, **31**(6): 869–877)

9 Maja K, Chou P A, Savill P. 8i voxelized surface light field (8iVSLF) dataset. ISO/IEC JTC1/SC29/WG11 MPEG, input document m42914, Ljubljana, Slovenia, 2018

10 Kammerl J, Blodow N, Rusu R B, Gedikli S, Beetz M, Steinbach E. Real-time compression of point cloud streams. In: Proceedings of the IEEE International Conference on Robotics and Automation. Saint Paul, USA: IEEE, 2012. 778–785

11 Thanou D, Chou P A, Frossard P. Graph-based compressionof dynamic 3D point cloud sequences. *IEEE Transactions on Image Processing*, 2016, **25**(4): 1765–1778

12 Queiroz R L, Chou P A. Motion-compensated compression of dynamic voxelized point clouds. *IEEE Transactions on Image Processing*, 2017, **26**(8): 3886–3895

13 Mekuria R, Blom K, Cesar P. Design, implementation, and evaluation of a point cloud codec for tele-immersive video. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017, **27**(4): 828–842

14 Besl P J, McKay N D. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992, **14**(2): 239–256

15 Santos C, Goncalves M, Correa G, Porto M. Block-based interframe prediction for dynamic point cloud compression. In: Proceedings of the IEEE International Conference on Image Processing. Anchorage, USA: IEEE, 2021. 3388–3392

16 Lasserre S, Llach J, Guede C, Ricard J. Technicolor′s response to the CfP for point cloud compression. ISO/IEC JTC1/SC29/WG11 MPEG, input document m41822, Macau, China, 2017

17 Budagavi M, Faramarzi E, Ho T, Najaf-Zadeh H, Sinharoy I. Samsung′s response to CfP for point cloud compression (Category2). ISO/IEC JTC1/SC29/WG11 MPEG, input document m41808, Macau, China, 2017

18 Schwarz S, Sheikhipour N, Sevom V F, Hannuksela M M. Video coding of dynamic 3D point cloud data. *APSIPA Transactions on Signal and Information Processing*, DOI: 10.1017/ATSIP.2019.24

19 Mammou K, Tourapis A M, SINGER D. Video-based and hierarchical approaches point cloud compression. ISO/IEC JTC1/SC29/WG11 MPEG, input document m41779, Macua, China, 2017

20 Graziosi D, Nakagami O, Kuma S, Zaghetto A, Suzuki T, Tabatabai A. 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*, DOI: 10.1017/ATSIP.2020.12

21 Li L, Zhu L, Zakharchenko V, Chen J L, Li H Q. Advanced 3D motion prediction for video-based dynamic point cloud compression. *IEEE Transactions on Image Processing*, 2020, **29**: 289–302

22 Kim J, Im J, Rhyu S, Kim K. 3D motion estimation and compensation method for video-based point cloud compression. *IEEE Access*, 2020, **8**: 83538–83547

23 Costa A, Dricot A, Brites C, Ascenso J, Pereira F. Improved patch packing for the MPEG V-PCC standard. In: Proceedings of the IEEE 21st International Workshop on Multimedia Signal Processing. Kuala Lumpur, Malaysia: IEEE, 2019. 1–6

24 Li L, Li Z, Liu S, Li H Q. Occupancy-map-based rate distortion optimization and partition for video-based point cloud compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020, **31**(1): 326–338

25 Li L, Li Z, Liu S, Li H Q. Efficient projected frame padding for video-based point cloud compression. *IEEE Transactions on Multimedia*, 2021, **23**: 2806–2819

26 Xu Y Q, Hu W, Wang S S, Zhang X F, Guo W. Predictive generalized graph fourier transform for attribute compression of dynamic point clouds. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021, **31**(5): 1968–1982

27 Schwarz S, Martin-Cocher G, Flynn D, Budagavi M. Common test conditions for point cloud compression. ISO/IEC JTC1/SC29/WG11 MPEG, input docment w17766, Ljubljana, Slovenia, 2020

28 Jang E S, Preda M, Mammou K, Tourapis A M, Kim J, Graziosi, et al. Video-based point-cloud-compression standard in MPEG: From evidence collection to committee draft. *IEEE Signal Processing Magazine*, 2019, **36**(3): 118–123

陈 建 福州大学物理与信息工程学院副教授. 主要研究方向为视频编码, 压缩感知, 点云压缩和目标跟踪.
E-mail: chenjian-fzu@163.com
(**CHEN Jian** Associate professor at the College of Physics and Information Engineering, Fuzhou University. Her research interest covers video coding, compression sensing, point cloud compression and target tracking.)

廖燕俊 福州大学先进制造学院硕士研究生. 主要研究方向为点云分割和视频点云压缩.
E-mail: liao.yanjun@foxmail.com
(**LIAO Yan-Jun** Master student at the School of Advanced Manufacturing, Fuzhou University. His research interest covers point cloud segmentation and dynamic point cloud compression.)

王 适 福州大学物理与信息工程学院硕士研究生. 主要研究方向为多媒体技术.
E-mail: wang_kuo@cib.com.cn
(**WANG Kuo** Master student at the College of Physics and Information Engineering, Fuzhou University. His main research interest is multimedia technology.)

郑明魁 福州大学物理与信息工程学院副教授. 主要研究方向为计算机视觉, 点云与视频编码. 本文通信作者.
E-mail: zhengmk@fzu.edu.cn
(**ZHENG Ming-Kui** Associate professor at the College of Physics and Information Engineering, Fuzhou University. His research interest covers computer vision, video and point cloud coding. Corresponding author of this paper.)

苏立超 福州大学计算机与大数据学院/软件学院讲师. 主要研究方向为多媒体信息安全.
E-mail: fzu-slc@fzu.edu.cn
(**SU Li-Chao** Lecturer at the College of Computer and Data Science/College of Software, Fuzhou University. His main research interest is multimedia infomation security.)