深度上下文动态点云几何压缩

江照意, 郑晟豪, 杨柏林*

(浙江工商大学计算机科学与技术学院 杭州 310018)
(ybl@mail.zjgsu.edu.cn)

摘 要: 现有的动态点云压缩方法采用预测编码的框架, 通过对当前帧点云进行预测计算预测帧与当前帧的残差并进行编码, 但由于使用残差编码消除帧间冗余, 在运动比较剧烈和内容细节丰富的区域压缩效果欠佳. 针对此缺陷, 使用条件编码替代传统的预测编码, 提出一个基于条件编码的深度上下文动态点云几何压缩方法. 首先构建一个特征空间多尺度场景流网络, 用于计算动态点云的运动向量, 以提高运动估计的精度; 然后通过上下文生成模块构造上下文信息, 并且直接以此信息为条件对当前帧点云进行编码和解码; 最后融合时间先验与超先验信息, 设计一个基于上下文信息的熵模型, 用于估计点云的条件概率分布, 提高熵编码效率. 以率失真曲线的 BD-rate 作为性能指标, 与 D-DPCC 相比, 在 8iVFB 数据集上, 所提方法的平均比特率降低 13.62%; 在 MVUB 数据集上, 该方法的平均比特率降低 15.15%; 压缩性能得到显著提升.

关键词: 上下文信息; 条件编码; 动态点云压缩; 场景流估计
中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2023-00351

Deep Contextual Dynamic Point Cloud Geometry Compression

Jiang Zhaoyi, Zheng Shenghao, and Yang Bailin*

(*School of Computer Science and Technology, Zhejiang Gongshang University, Hangzhou* 310018)

Abstract: Existing dynamic point cloud compression methods adopt a predictive coding framework, where the residual between a predicted frame and the current frame is calculated and encoded based on the prediction of the current frame's point cloud. However, due to the use of residual coding to eliminate inter-frame redundancy, the compression performance in regions with intense motion and rich content details is subpar. To address this limitation, this paper proposes a depth context-based dynamic point cloud geometry compression method using conditional coding instead of traditional predictive coding. Firstly, a feature space multi-scale scene flow network is constructed to calculate the motion vectors of the dynamic point cloud, enhancing the accuracy of motion estimation. Secondly, a context generation module is utilized to construct context information, which is then directly used as a condition for encoding and decoding the current frame's point cloud. Lastly, by fusing temporal priors with hyperpriors, a context-based entropy model is designed to estimate the conditional probability distribution of the point cloud, thereby improving the efficiency of entropy coding. Evaluated using the BD-rate of rate-distortion curves as the performance metric, the proposed method achieves an average bitrate reduction of 13.62% compared to D-DPCC on the 8iVFB dataset and 15.15% on the MVUB dataset, significantly enhancing compression performance.

Key words: context information; conditional encoding; dynamic point cloud compression; scene flow estimation

收稿日期: 2023-06-18; 修回日期: 2024-08-20. **基金项目**: 国家自然科学基金(62172366); 浙江省自然科学基金(LY21F020013).
江照意(1975—), 男, 博士, 副教授, 硕士生导师, CCF 会员, 主要研究方向为计算机图形学、数字几何处理、机器学习、计算机视觉; **郑晟豪**(1999—), 男, 硕士研究生, 主要研究方向为数字几何处理、深度学习; **杨柏林**(1974—), 男, 博士, 教授, 博士生导师, CCF 会员, 论文通信作者, 主要研究方向为计算机图形学、虚拟现实、机器学习.

动态点云作为重要的 3D 数据表示形式，广泛应用于各种新兴领域中，如自动驾驶、虚拟现实、地图测绘等[1-2]. 动态点云的数据量巨大，对其进行存储和传输是非常困难的，需要进行高效的压缩. 然而，与图像和视频压缩不同，动态点云具有不均匀分布、无序性等特性，帧间冗余更加难以消除. 与传统的压缩基于预测编码的方法不同，本文通过构造上下文信息，采用条件编码的方式压缩动态点云.

动态点云压缩方法分为传统方法和深度学习方法. MPEG[1,3]提出 2 种传统的点云压缩标准：基于视频的 V-PCC 和基于几何的 G-PCC. V-PCC 先将点云序列转换成视频，利用传统视频压缩的方法进行压缩；G-PCC 利用高效的八叉树结构表示消除信息冗余，达到压缩的目的. 然而，传统方法依赖手工设计的特征，适用范围有限. 随着深度学习方法在图像和视频压缩的广泛应用，深度学习在动态点云压缩方面的应用也逐渐受到关注. Fan 等[4]首次提出一个端到端的深度动态点云压缩(deep dynamic point cloud compression, D-DPCC)方法，在特征空间中进行运动估计和运动补偿，但该方法将 2 帧点云特征简单拼接，用于估计运动向量，导致运动补偿生成的预测帧不够准确. 动态点云压缩方法都是采用残差编码的方式压缩动态点云，将使用手工制作或是学习得到的特征通过运动估计和运动补偿生成预测帧，对当前帧和预测帧的残差进行压缩；但是，在某些情况下，如帧间运动过于剧烈或者前帧内容在后帧中消失时，基于残差编码的压缩质量会降低. 理论上，残差编码的信息熵大于等于条件编码的信息熵[5]. 残差编码使用简单的减法操作获得残差信息，消除帧间冗余不充分. 因此，生成更加有效的条件信息用于编码以提高编码效率是一种新的探索. 在视频压缩领域，Li 等[5]采用条件编码的方式压缩视频，与基于残差编码[6]的方式不同，他们首先利用光流估计[7]计算运动向量，通过 MEMC(motion estimation and motion compensation)生成丰富的上下文信息，然后借助上下文信息直接对当前帧进行编码，与基于残差的压缩方法比较，实验结果显示编码效率有了显著的提高.

本文提出一种基于条件编码的深度上下文动态点云几何压缩方法. 针对 D-DPCC 中运动向量计算精度低的缺点，在特征空间执行多尺度场景流估计，以获得更加精确的 2 帧运动流；对于残差编码的局限性，设计一个上下文生成模块，将模块输出的上下文信息作为条件对当前帧进行编码和解码；最后提出一个基于上下文信息的熵模型，包含时间先验和空间超先验，将上下文信息作为时间先验模块的输入，利用上下文信息帮助熵模型更准确地估计当前帧的条件概率分布.

(1) 设计一种基于条件编码的深度上下文动态点云几何压缩框架，可以获得比传统基于残差编码的方法更好的压缩效果. 同时，在特征空间执行多尺度场景流估计以获取更高精度的运动向量.

(2) 提出一种简单有效的方法，使用学习的上下文信息来帮助动态点云的压缩. 将学习得到的上下文信息用于点云的编码、解码以及熵模型的构建.

(3) 融合时间先验与超先验信息设计一个基于上下文信息的熵模型，用于估计点云的条件概率分布，提高熵编码效率. 为了证明本文方法的有效性，在 MPEG 推荐的 8iVFB 数据集[8]和 MVUB 数据集[9]上进行实验，结果表明，在点到点峰值信噪比(point-to-point geometry peak signal-to-noise ratio, D1-PSNR)[10]上，该方法分别以 13.62%和 15.15%的 BD-Rate(bjontegaard delta rate)[11]增益优于 D-DPCC.

1 相关工作

1.1 静态点云压缩

传统的静态点云压缩方法使用八叉树结构[12-13]有效地组织点云数据，然后利用算术编码进行压缩. 当前，基于深度学习的静态点云压缩方法层出不穷，一些方法使用有效的上下文信息[14]得到更好的压缩效率. Huang 等[15]设计树形结构的条件熵模型，并将单节点和其多个祖先节点的依赖关系建模为上下文信息，得到了不错的压缩效率；考虑点云的稀疏性，Wang 等[16]将稀疏卷积[17]运用于点云压缩中，提出一种多尺度的端到端的深度点云压缩方法，通过渐进式重采样方法以分层重建 3D 点云几何，使用八叉树编解码器对几何占用进行无损压缩，使用学习到的概率上下文模型对特征属性进行有损压缩. 受此启发，本文方法也大量使用稀疏卷积.

1.2 动态点云压缩

V-PCC[1,3]是一种基于 2D 视频的传统动态点云压缩方法，其将 3D 点云投影到 2D，执行基于视频的运动估计和运动补偿；文献[18-19]模仿传统的视频压缩框架，并设计一些帧间预测方法：然而这些方法依赖于手工制作的特征，估计的运动向

量通常是不可靠的，影响后续编码的效率. 最近，研究人员提出基于深度学习的动态点云压缩方法. D-DPCC[4]在特征空间中执行帧间预测，使用基于有损自动编码器的网络压缩运动向量和 2 帧残差，得到了不错的压缩效率；Wang 等[20]利于多尺度时间先验，将多尺度静态点云压缩方法[16]扩展到动态点云压缩领域，也取得了不错的压缩效果.

1.3　熵模型

在压缩领域，熵模型的作用是将待编码变量的概率分布进行建模以得到准确估计，从而提高压缩质量[21]. 在图像压缩方面，Ballé 等[22]提出一种基于可变自动编码器端到端图像压缩方法，其结合一个超先验编码器，用于捕捉丰富的空间信息；在视频压缩方面，Sun 等[23]使用隐式网络替代运动预测模块来降低网络复杂度，结合时空熵模型减少时空冗余；在点云压缩方面，Biswas 等[24]设计一种新的条件熵模型，考虑粗级别几何形状和先前扫描的几何强度信息，来模拟八叉树符号的概率. 受到文献[25]的启发，本文提出融合超先验和时间先验的熵模型，可以更好地对潜在变量进行概率估计.

1.4　视频压缩

随着深度学习的广泛应用，针对视频压缩也出现一些新颖的方法，文献[26-28]使用深度学习构建压缩框架；Lu 等[29]提出第 1 个端到端的深度学习视频压缩框架 DVC，运用传统视频压缩方法将残差编码与深度学习的强大非线性表示能力相结合，联合优化了视频压缩的所有组件；Hu 等[30]优化 DVC，在特征空间中进行运动估计、压缩补偿和残差压缩，进一步提升了运动估计的准确度，得到更好的效果；Li 等[5]则使用条件编码替代传统的残差编码，将从前后 2 帧中获取的上下文信息作为条件来直接压缩当前帧，利用丰富的上下文信息得到更好的压缩质量. 受其启发，本文将条件编码运用于动态点云的压缩框架中，然而由于点云独特的属性，如点云分布非均匀等，直接应用存在巨大的挑战.

2　本文方法

2.1　总体框架

本文将生成的上下文信息作为条件，用于当前帧点云压缩，以显著降低压缩时的比特率消耗. 将动态点云序列 $\boldsymbol{p}=\{\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_{t-1},\boldsymbol{p}_t,\cdots\}$ 定义为整个框架的输入，其中，$\boldsymbol{p}_t$ 表示当前帧的点云. 为了提高点云张量处理的效率，采用稀疏卷积[17]表示每个点云 $\boldsymbol{p}_t=\left\{\boldsymbol{C}_{\boldsymbol{p}_t},\boldsymbol{F}_{\boldsymbol{p}_t}\right\}$. 其中，$\boldsymbol{C}_{\boldsymbol{p}_t}$ 为点云的几何坐标矩阵，用来表示顶点的空间位置；$\boldsymbol{F}_{\boldsymbol{p}_t}$ 为特征矩阵，用来表示体素被占用情况. 已有的深度动态点云压缩方法中，帧间压缩常采用残差编码[6]的方式，公式为

$$\hat{\boldsymbol{p}}_t=f_{\mathrm{dec}}\left(\left\lfloor f_{\mathrm{enc}}\left(\boldsymbol{p}_t-\tilde{\boldsymbol{p}}_t\right)\right\rfloor\right)+\tilde{\boldsymbol{p}}_t \text{ with}$$
$$\tilde{\boldsymbol{p}}_t=f_{\mathrm{predict}}\left(\hat{\boldsymbol{p}}_{t-1},\boldsymbol{p}_t\right).$$

其中，$f_{\mathrm{enc}}(\cdot)$ 和 $f_{\mathrm{dec}}(\cdot)$ 表示编码器和解码器；$\hat{\boldsymbol{p}}_t$ 和 $\hat{\boldsymbol{p}}_{t-1}$ 表示当前帧和前一帧的重构点云，$\tilde{\boldsymbol{p}}_t$ 表示当前帧的预测点云；$\lfloor\cdot\rceil$ 表示量化操作；$f_{\mathrm{predict}}(\cdot)$ 表示生成预测帧的函数，通常使用 MEMC 来生成预测.

本文方法没有使用传统的残差编码，而是采用条件编码将特征空间中计算得到的上下文信息[14]作为条件，直接对当前帧进行压缩，公式为

$$\hat{\boldsymbol{p}}_t=f_{\mathrm{dec}}\left(\left\lfloor f_{\mathrm{enc}}\left(\boldsymbol{p}_t\mid\bar{\boldsymbol{p}}_t\right)\right\rfloor\mid\bar{\boldsymbol{p}}_t\right) \text{ with}$$
$$\bar{\boldsymbol{p}}_t=f_{\mathrm{context}}\left(\hat{\boldsymbol{p}}_{t-1},\boldsymbol{p}_t\right).$$

其中，$\bar{\boldsymbol{p}}_t$ 表示上下文信息，$f_{\mathrm{context}}(\cdot)$ 表示生成上下文的函数.

本文方法的结构如图 1 所示，包含特征提取模块、运动估计运动压缩模块、上下文生成模块和上下文编解码模块共 4 个部分. (1) 当前帧 $\boldsymbol{p}_t$ 和先前重构帧 $\hat{\boldsymbol{p}}_{t-1}$ 先经过特征提取模块得到特征空间表示 $\boldsymbol{y}_t$ 和 $\hat{\boldsymbol{y}}_{t-1}$；(2) $\boldsymbol{y}_t$ 和 $\hat{\boldsymbol{y}}_{t-1}$ 进入运动估计和运动压缩模块，建立 2 帧间的联系，输出运动向量 $\boldsymbol{m}_t$ 并对其压缩得到 $\hat{\boldsymbol{m}}_t$；(3) 将 $\hat{\boldsymbol{m}}_t$ 和 $\hat{\boldsymbol{y}}_{t-1}$ 输入上下文生成模块，计算得到上下文信息 $\bar{\boldsymbol{p}}_t$；(4) 在上下文编解码模块中，上下文信息 $\bar{\boldsymbol{p}}_t$ 作为条件帮助当前帧的编码和解码，当前帧先通过上下文编码器与上下文 $\bar{\boldsymbol{p}}_t$ 结合，编码得到 $\boldsymbol{z}_t$，然后通过上下文解码器最终输出当前帧的重构帧 $\hat{\boldsymbol{p}}_t$. 此外，还将上下文信息应用于熵建模.

2.2　特征提取

为了减少点云的空间冗余，将当前帧 $\boldsymbol{p}_t$ 和前一重构帧 $\hat{\boldsymbol{p}}_{t-1}$ 输入特征提取模块，获得在特征空间的表示 $\boldsymbol{y}_t$ 和 $\hat{\boldsymbol{y}}_{t-1}$. 特征提取模块的结构如图 2 所示，由 4 层稀疏卷积组成，其中，第 2，4 层稀疏卷积步长为 2，用于对点云进行下采样，每 2 层稀疏卷积后加上 3 层初始残差网络(inception-residual network, IRN)，目的是聚合下采样后的特征. 该模块可以得到当前帧和前一重构帧在特征空间中的表示，之后将它们输入场景流网络计算运动向量.

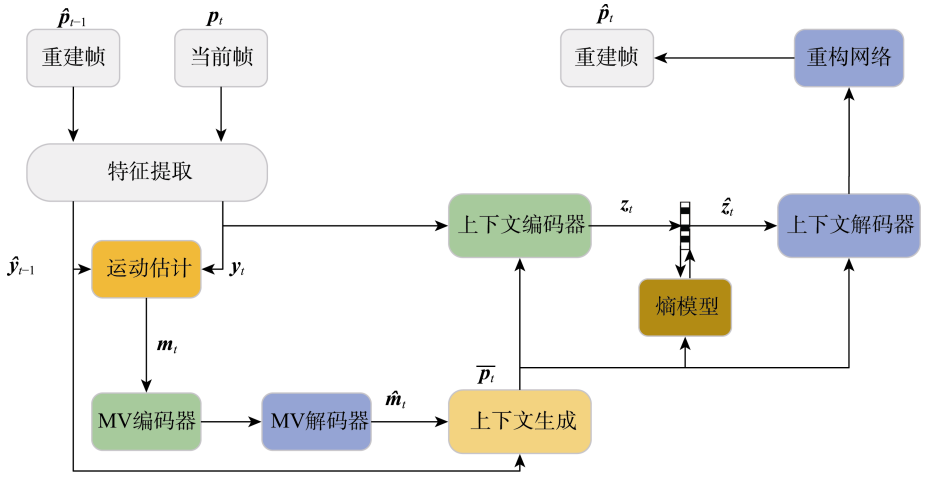
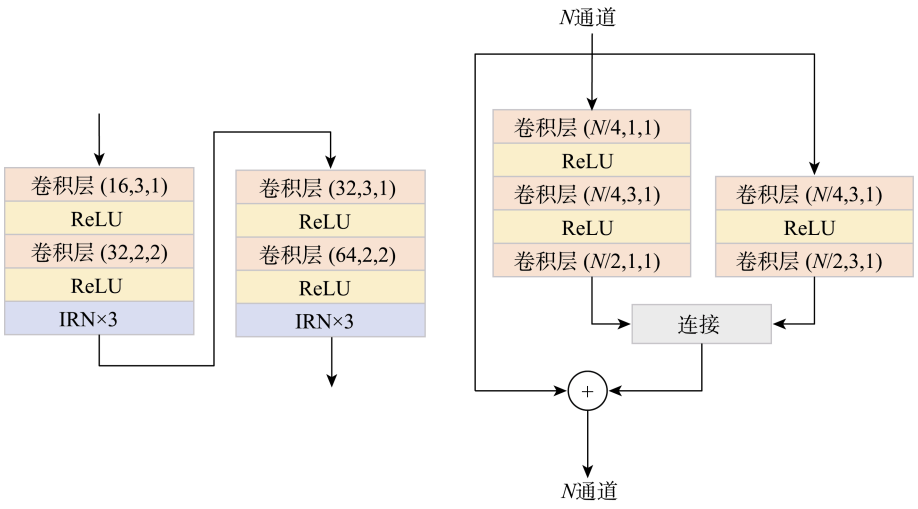


图 1 本文方法结构



a. 具体网络结构 b. IRN

图 2 特征提取模块结构

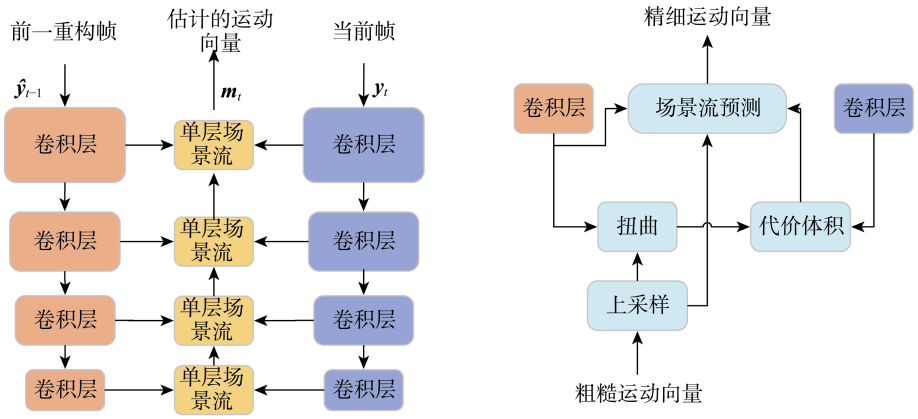
2.3 运动估计和运动压缩

利用根据特征提取模块得到的在特征空间中的相邻 2 帧点云，计算运动向量并将其压缩. 运动估计的目的是计算前一帧点云到当前帧点云的运动偏移信息，偏移信息可以执行上下文生成操作，获取当前帧上下文信息. 现有的场景流估计方法可以分为多尺度场景流[31]和单尺度场景流[2,32-33]. 其中，多尺度场景流可以有效地估计前后 2 帧运动较大的点云场景；单尺度方法对移动较快物体的运动估计具有更好的效果，但单尺度场景流通常要计算复杂的相似性矩阵，因此不适合大规模点云场景的运动估计. 考虑稀疏卷积在处理 3D 任务中发挥的独特优势，如稀疏卷积可以在缩放的特征空间关联点云坐标，同时可以匹配流嵌入层的特征，本文从多尺度场景流方法[31]中获取灵感，将稀疏卷积与该框架结合，设计一种在特征空间中的多尺度场景流估计网络，其结构如图 3 所示.

本文的场景流网络由多个单层场景流网络构成，每层将粗糙尺度的运动向量进行上采样，利用相同尺度的 2 帧点云特征进行场景流预测，得到细粒度的运动向量. 最终从场景流网络中输出 2 帧的最细粒度运动向量 $\boldsymbol{m}_t$，然后使用自动编码器的有损压缩网络压缩 $\boldsymbol{m}_t$. 在图 1 中表示为 MV 编码器和 MV 解码器.

2.4 上下文信息生成与使用

得到当前帧的上下文信息[14]的常规方法是直接将前一重构帧输入到由几个卷积组成的普通网络输出 $\overline{\boldsymbol{p}}_t$，但这样直接的方式很难提取到有利于压缩的上下文信息. 由于动态点云序列中通常存在复杂的动作变化，2 帧间相同位置可能具有较小的相关性，因此如果使用上述方式得到的 $\overline{\boldsymbol{p}}_t$ 作为条件去压缩当前帧可能会产生较大误差，无法充



a. 场景流估计网络　　b. 单层场景流

图 3　多尺度场景流估计网络结构

分发挥上下文信息在压缩中的有效性.

本文采用 MEMC 生成上下文信息，但不同的是选择在特征空间进行. 上下文信息的生成公式为

$$f_{\text{context}}\left(\hat{\boldsymbol{p}}_{t-1}\right)=f_{\text{warp}}\left(f_{\text{fe}}\left(\hat{\boldsymbol{p}}_{t-1}\right),\hat{\boldsymbol{m}}_t\right).$$

其中，$f_{\text{fe}}(\cdot)$ 表示特征提取函数，用于获取前一重构帧 $\hat{\boldsymbol{p}}_{t-1}$ 的特征空间表示；$\hat{\boldsymbol{m}}_t$ 是由运动估计模块和运动压缩模块获得的，携带当前帧和前一重构帧之间的运动信息，指引网络在什么位置提取有用的上下文信息；$f_{\text{warp}}(\cdot)$ 表示生成上下文信息的函数，这是受到自适应 3D 插值算法[4]启发而设计的. 输入是前一重构帧 $\hat{\boldsymbol{y}}_{t-1}$ 和运动向量 $\hat{\boldsymbol{m}}_t$；最终输出上下文信息 $\overline{\boldsymbol{p}}_t$，帮助当前帧的压缩. 上下文编解码网络结构如图 4 所示，其中，编码当前帧、解码当前帧和熵建模部分都用到上下文信息 $\overline{\boldsymbol{p}}_t$. 连接上下文信息 $\overline{\boldsymbol{p}}_t$ 和当前帧点云 $\boldsymbol{y}_t$，通过上下文编码器得到当前帧的潜在编码 $\boldsymbol{z}_t$，$\boldsymbol{z}_t$ 经过熵模型被编码成 $\hat{\boldsymbol{z}}_t$，这一步将显著地降低压缩的比特大小，提升压缩效率；然后，$\hat{\boldsymbol{z}}_t$ 输入到上下文解码器再次连接上下文信息进行解码，最终生成当前帧的重构帧 $\hat{\boldsymbol{p}}_t$.

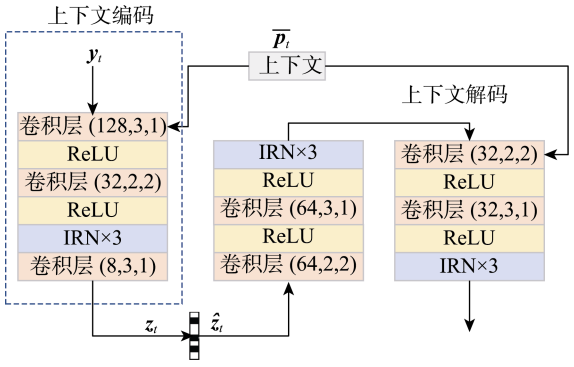


图 4　上下文编解码网络结构

2.5　熵模型

熵模型的作用是更好地对潜在代码 $\boldsymbol{z}_t$ 进行概率估计，以获得更准确的压缩效果. 本文设计了一个具有超先验编解码器和时间先验的深度熵模型，其结构如图 5 所示. 其中，$\boldsymbol{z}_t$ 是经过上下文编码器后的潜在代码，携带了当前帧信息和上下文信息.

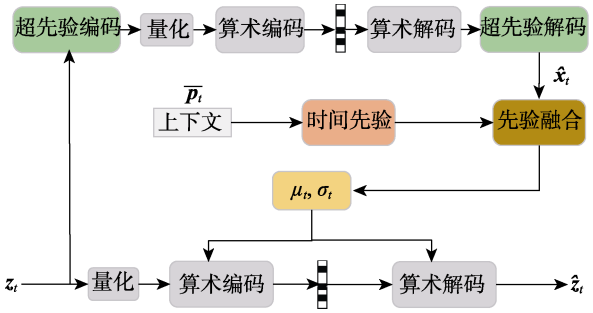


图 5　熵模型网络结构

熵模型经过超先验编解码(如图 5 中绿色部分所示)得到超先验参数. 由于动态点云序列存在很强的时间相关性，因此需要一个时间先验来更有效地利用相邻帧之间的时间信息，从而更准确地估计 $\boldsymbol{z}_t$ 的概率分布. 本文使用先前生成的上下文信息 $\overline{\boldsymbol{p}}_t$ 计算时间先验参数，将 $\overline{\boldsymbol{p}}_t$ 输入到时间先验编码器得到时间先验参数；将量化后的 $\boldsymbol{z}_t$ 使用高斯分布建模它的概率分布，高斯分布的均值 μ 和标准差 σ 将从先验融合模块(如图 5 中红色部分所示)获得；最后得到熵编码后的 $\hat{\boldsymbol{z}}_t$，其后续通过上下文解码生成当前帧的重构帧. 熵模型中各个模块的具体细节如图 6 所示.

2.6　损失函数

使用率失真联合损失函数优化本文方法，公式为 $L=R+\lambda D$. 其中，R 表示平均每点所占比特

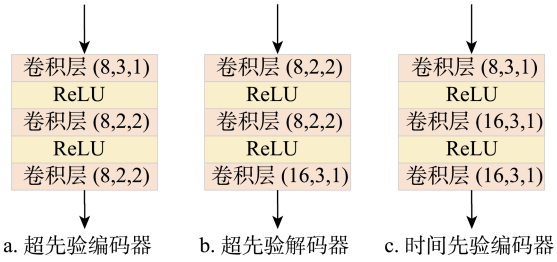


图 6 熵模型框架中各个模块的具体细节

数(bits per point, BPP); D 表示当前帧 $\boldsymbol{p}_t$ 与其重构帧 $\hat{\boldsymbol{p}}_t$ 之间的失真度[16]; λ 是拉格朗日乘子, 用来权衡失真与速率. R 由 $L_{\hat{F}_{m_t}}$, $L_{\hat{F}_{x_t}}$ 和 $L_{\hat{F}_{z_t}}$ 这 3 部分组成, 单个部分的计算公式为

$$L_k = \frac{1}{N_k}\sum_{i=1}^{N_k} -\mathrm{lb}\left(p_k\right),\ \left\{k \in \hat{F}_{m_t}, \hat{F}_{x_t}, \hat{F}_{z_t}\right\}.$$

其中, L_k 表示 $L_{\hat{F}_{m_t}}$, $L_{\hat{F}_{x_t}}$, $L_{\hat{F}_{z_t}}$ 中任意一个, N_k 表示相应的属性数量. D 的计算公式为

$$D = \frac{1}{M}\sum_{j=1}^{M} L_{\mathrm{BCE}}^{j}.$$

其中, L_{BCE}^{j} 可以公式化为

$$L_{\mathrm{BCE}}^{j} = \frac{1}{N_{p_t}}\left\{-\sum_i \left[o_i \mathrm{lb}\left(p_i\right) + \left(1-o_i\right)\mathrm{lb}\left(1-p_i\right)\right]\right\}.$$

其中, o_i 表示该点是否占用体素, 0 表示不占用, 1 表示占用; p_i 表示该点占用体素的概率.

3 实验与结果分析

3.1 实验设置

3.1.1 训练集

使用 owlii 动态人类点云数据集[34]训练本文模型, 它包含 4 个序列, 一共 2 400 帧, 分别是 Basketball-player, Dancer, Exercise 和 Model. 其中, Basketball-player 和 Dancer 序列包含较为复杂的运动且运动幅度较大; 每个序列的帧速率为 30 帧/s, 共 20 s; 为了减少训练时长和空间消耗, 本文将数据集从 11 bit 精度量化为 9 bit 精度.

3.1.2 测试集

使用 8iVFB 体素化点云数据集(简称 8iVFB 数据集)[8]和 MVUB(Microsoft voxelized upper bodies)体素化点云数据集(简称 MVUB 数据集)[8]评估本文方法. 在 8iVFB 数据集上选择 4 个点云序列, 分别为 Soldier, Loot, Redandblack 和 Longdress; 每个序列为 10 s 的 30 帧/s 动态点云, 共 1 200 帧. 在 MVUB 数据集上也选择 4 个点云序列, 分别是 Andrew, Phil, Ricardo 和 Sarah; 每个序列选取 100 帧进行测试.

3.1.3 训练细节

训练 4 个点云序列时, 使用 Adam 优化器, β 为(0.900, 0.999), 初始学习率设置为 0.000 8、每 15 个训练时期的衰减率为 0.7 的学习调度器; 分别设置 λ 为 3, 4, 5, 7, 9 来训练本文模型; 训练期间 batch 大小为 4, 共训练 200 个时期.

3.1.4 评估指标

使用 BPP 测量每个点的比特消耗, D1-PSNR[10]和点到面峰值信噪比(D2-PSNR)这 2 个失真指标来衡量点云的重建质量. 压缩的目的是在相同的 BPP 下获得尽可能高的 PSNR.

3.1.5 基 准

将本文方法与其他较先进的方法进行实验, 包括 D-DPCC[4], V-PCC[3]测试模型 v13 和静态点云几何压缩方法[16]. 使用单帧压缩的方式对静态压缩方法进行实验.

3.2 对比实验

图 7 所示为 8iVFB 数据集上, 4 种压缩方法在 Soldier, Loot, Redandblack 和 Longdress 这 4 个点云序列上的率失真曲线对比, 其中, Wang 等[16]方法表示静态压缩方法. 可以看出, 本文方法在 4 个点云序列上均优于对比方法. 表 1 所示为图 7 中 4 个点云序列相应的 BD-Rate[11]增益结果, 其中, 基准方法为 D-DPCC, 负值表示比特率节省, 即压缩算法更佳. 可以看出: (1) 与 D-DPCC 相比, 本文方法获得的平均 BD-Rate 增益为 13.62%(D1)和 17.11%(D2). (2) 本文方法在大部分点云序列上的表现优于基准方法, 例如, 在 Soldier 序列上获得了大于 25%(D1)和大于 17%(D2)的 BD-Rate 增益; 在 Loot 和 Longdress 点云序列上也有很大提升, Loot 获得了大于 9%(D1)和大于 14%(D2)的 BD-Rate 增益, Longdress 获得了大于 18%(D1)和大于 24%(D2)的 BD-Rate 增益; 在 Redandblack 序列上只取得了与 D-DPCC 相近的结果. 在 MVUB 数据集上, Andrew, Phil, Ricardo 和 Sarah 这 4 个点云序列 BD-Rate[11]增益结果对比结果如表 2 所示, 与表 1 中相同, 也选择 D-DPCC 为基准方法. 可以看出, 本文方法在 4 个点云序列上的压缩效果均优于 D-DPCC 方法, 分别取得了 15.15%(D1)和 11.02%(D2)的 BD-Rate 增益. 实验结果表明, 由于采用了特征空间的多尺度场景流估计、上下文信息

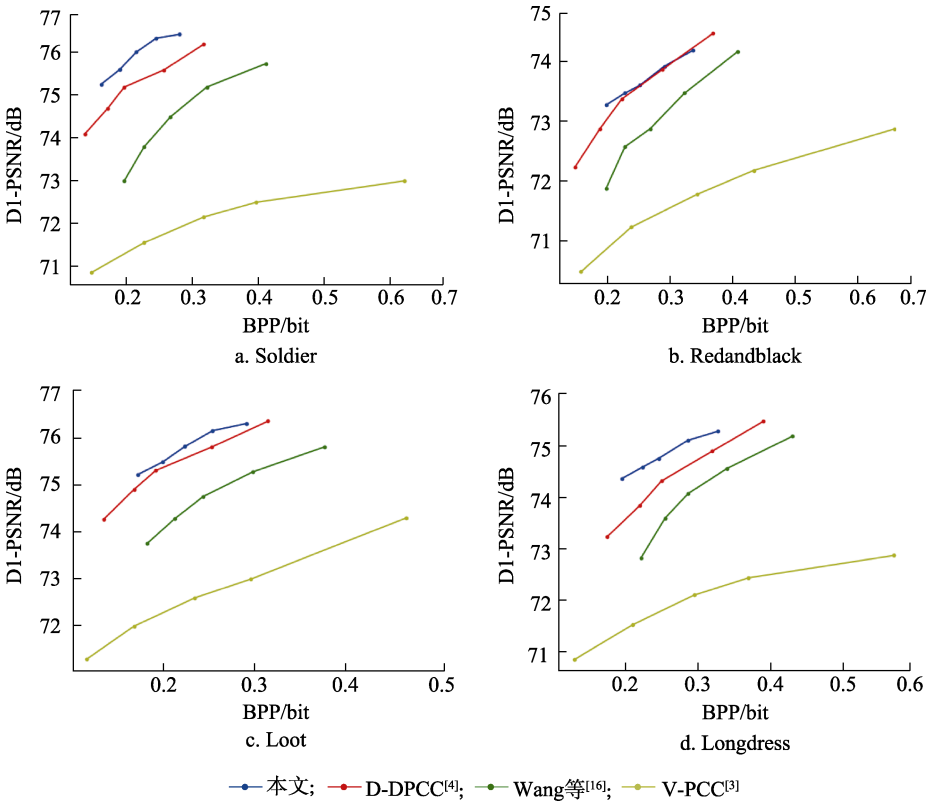


图 7　8iVFB 数据集上 4 种压缩方法在 4 个点云序列上的率失真曲线对比

表 1　8iVFB 数据集上与 D-DPCC[4]的 BD-Rate 结果　%

点云序列	D1	D2
Soldier	–25.21	–17.42
Redandblack	–1.29	–12.07
Loot	–9.81	–14.31
Longdress	–18.16	–24.64
平均	–13.62	–17.11

表 2　MVUB 数据集上与 D-DPCC[4]的 BD-Rate 结果　%

点云序列	D1	D2
Andrew	–15.35	–6.24
Phil	–15.59	–9.19
Ricardo	–13.27	–14.09
Sarah	–16.37	–14.54
平均	–15.15	–11.02

条件编码以及融合熵模型，本文方法在 2 个数据集上获得了比基准方法更优的结果.

3.3　可视分析与消融实验

3.3.1　可视化分析

将本文方法得到的重构点云、原始点云和 D-DPCC 重构点云进行可视化对比，结果如图 8 所示. 其中，选择 Loot 点云序列的第 8 帧进行对比，每个分图下方是手部特写. 将每种方法的 BPP 控制在 0.26，得到的 D1-PSNR 分别为 75.55 dB (D-DPCC)和 76.12 dB(本文方法). 可以看出，本文方法的重构帧在手部的重建质量好于基准方法. 选择 Soldier 点云序列的第 8 帧，将本文方法在不同 BPP 下的重建结果进行可视化对比，结果如表 3 所示. 可以看出，随着 BPP 的增大，点云中一些重建缺失越来越少，人物的面部细节越来越丰富.



a. 原始帧　　b. 本文　　c. D-DPCC[4]

图 8　重建帧可视化对比

表 3　本文方法在不同 BPP 下的重建结果可视化对比

BPP	可视化结果	D1-PSNR/dB
0.07	缺失较多	70.1
0.19	缺失一般	75.6
0.28	缺失较少	76.4

3.3.2　条件编码的有效性

为了验证上下文信息条件编码的有效性，将本文方法中的条件编码替换成传统的残差编码，然后与本文方法进行比较．选择在 Soldier 和 Redandblack 点云序列上进行实验，结果如图 9 所示．可以看出，与残差编码版本相比，本文方法获得了 2.41%(D1)和 3.32%(D2)的 BD-Rate 增益，在 Redandblack 上获得了 10.72%(D1)和 5.61%(D2)的 BD-Rate 增益．还在 Longdress 点云序列的第 8 帧将这 2 种方法可视化，结果如图 10 所示．可以看出，在图下方红圈位置中，图 10c 的重建帧有更多空缺，这是因为上下文信息作为条件进行压缩更能捕捉到 2 帧间的运动细节，有利于点云局部重建．

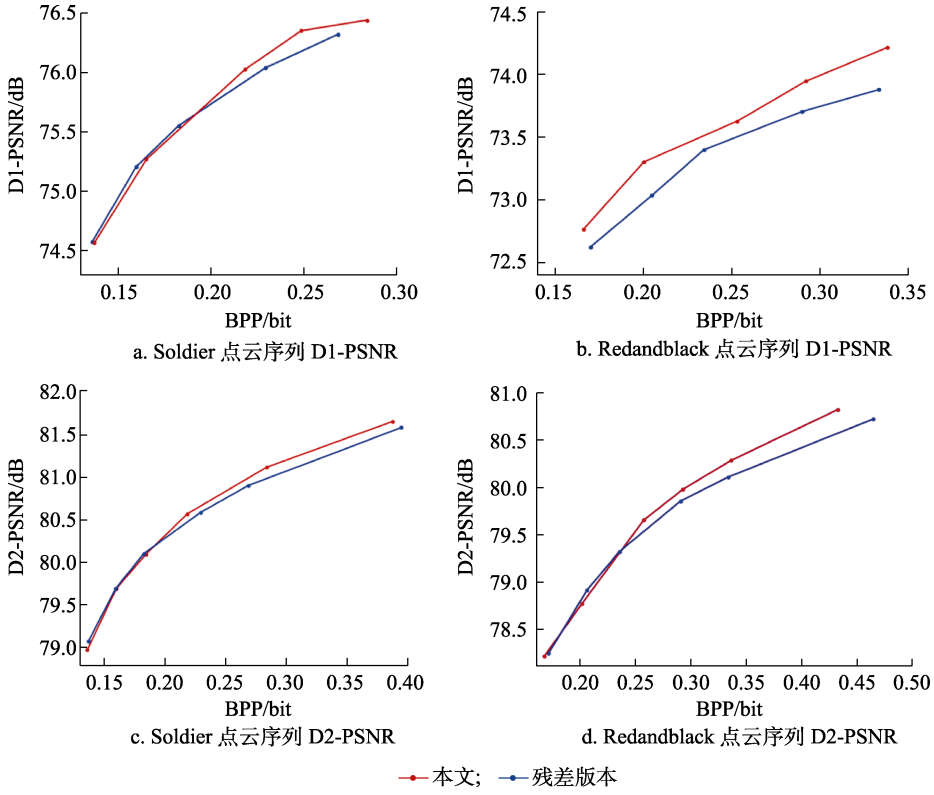
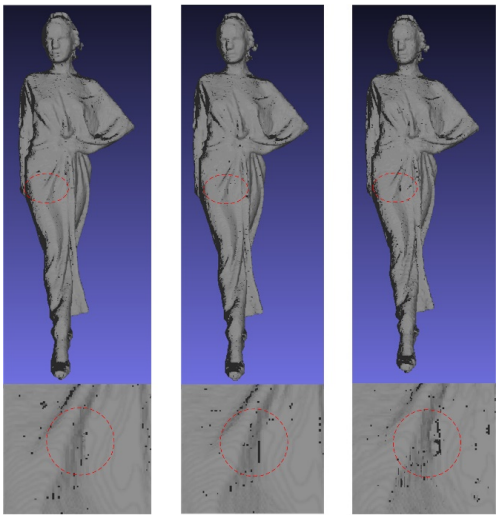


图 9　2 个点云序列上不同编码方式率失真曲线对比



a. 原始点云　b. 本文方法　c. 残差版本

图 10　条件编码与残差编码可视化对比

3.3.3　不同上下文信息生成方式比较

为了探究上下文信息的最佳生成方式，将本文方法与直接使用前一重构帧输入简单卷积层得到上下文信息的方法进行了对比实验，结果如图 11 所示．可以看出，与直接使用前一帧的方法相比，本文生成上下文的方法在编码效果上获得了显著提升，表明本文生成上下文信息的可靠性．

4　结　语

本文提出一种基于上下文信息的深度动态点云压缩方法．受到 Li 等[5]视频压缩方法的启发，将传统的残差编码方式替换成条件编码，在运动估计方面提出在特征空间执行多尺度场景流，以更准确地估计 2 帧运动；通过上下文生成模块计算当

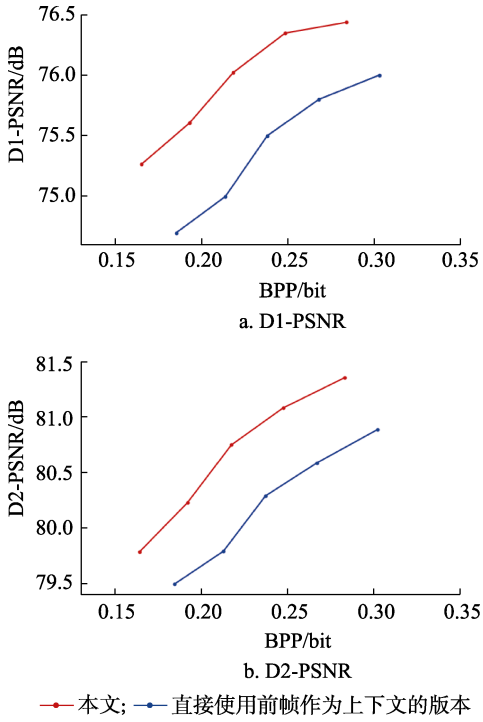


图 11　Soldier 点云序列上不同上下文生成方式率失真曲线对比

前帧的上下文信息，并作为条件直接对当前帧进行压缩，该信息将应用于当前帧的编码、解码和熵模型；还引入融合了超先验和时间先验的熵模型，以更好地对当前帧进行熵编码，显著降低比特消耗. 与较先进的 D-DPCC 进行实验的结果表明，在 8iVFB 数据集上，本文方法获得了平均 13.62%(D1) 的 BD-Rate 增益；在 MVUB 数据集上，该方法获得了平均 15.15%(D1)的 BD-Rate 增益. 未来还将探索更多上下文信息的生成方式以及更为有效的点云编码方式.

参考文献(References):

[1] 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

[2] Puy G, Boulch A, Marlet R. FLOT: scene flow on point clouds guided by optimal transport[C] //Proceedings of the 16th European Conference on Computer Vision. Heidelberg: Springer, 2020: 527-544

[3] Graziosi D, Nakagami O, Kuma S, *et al*. An overview of ongoing point cloud compression standardization activities: video-based (V-PCC) and geometry-based (G-PCC)[J]. APSIPA Transactions on Signal and Information Processing, 2020, 9(1): e13

[4] Fan T Y, Gao L Y, Xu Y L, *et al*. D-DPCC: deep dynamic point cloud compression via 3D motion prediction[C] //Proceedings of the 31st International Joint Conference on Artificial Intelligence. Freiburg: IJCAI, 2022: 898-904

[5] Li J H, Li B, Lu Y. Deep contextual video compression[C] //Proceedings of the 35th International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc, 2021: Article No.1386

[6] Kammerl J, Blodow N, Rusu R B, *et al*. Real-time compression of point cloud streams[C] //Proceedings of the IEEE International Conference on Robotics and Automation. Los Alamitos: IEEE Computer Society Press, 2012: 778-785

[7] Ranjan A, Black M J. Optical flow estimation using a spatial pyramid network[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 2720-2729

[8] d'Eon E, Harrison B, Myers T, *et al*. 8i voxelized full bodies-a voxelized point cloud dataset[R]. Geneva: MPEG, 2017

[9] Loop C, Cai Q, Escolano S O, *et al*. Microsoft voxelized upper bodies-a voxelized point cloud dataset[R]. Geneva: MPEG, 2016

[10] Tian D, Ochimizu H, Feng C, *et al*. Geometric distortion metrics for point cloud compression[C] //Proceedings of the IEEE International Conference on Image Processing. Los Alamitos: IEEE Computer Society Press, 2017: 3460-3464

[11] Bjøntegaard G. Calculation of average PSNR differences between RD-curves[OL]. [2023-06-18]. https://api.semanticscholar.org/CorpusID:61598325

[12] Schnabel R, Klein R. Octree-based point-cloud compression[C] //Proceedings of the 3rd Eurographics/IEEE VGTC Conference on Point-Based Graphics. Aire-la-Ville: Eurographics Association Press, 2006: 111-121

[13] Wen X Z, Wang X, Hou J H, *et al*. Lossy geometry compression of 3D point cloud data via an adaptive octree-guided network[C] //Proceedings of the IEEE International Conference on Multimedia and Expo. Los Alamitos: IEEE Computer Society Press, 2020: 1-6

[14] Garcia D C, de Queiroz R L. Intra-frame context-based octree coding for point-cloud geometry[C] //Proceedings of the 25th IEEE International Conference on Image Processing. Los Alamitos: IEEE Computer Society Press, 2018: 1807-1811

[15] Huang L L, Wang S L, Wong K, *et al*. OctSqueeze: octree-structured entropy model for LiDAR compression[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2020: 1310-1320

[16] Wang J Q, Ding D D, Li Z, *et al*. Multiscale point cloud geometry compression[C] //Proceedings of the Data Compression Conference. Los Alamitos: IEEE Computer Society Press, 2021: 73-82

[17] Choy C, Gwak J Y, Savarese S. 4D spatio-temporal ConvNets: minkowski convolutional neural networks[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2019: 3070-3079

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

[19] Thanou D, Chou P A, Frossard P. Graph-based compression of

dynamic 3D point cloud sequences[J]. IEEE Transactions on Image Processing, 2016, 25(4): 1765-1778

[20] Wang J Q, Ding D D, Chen H, *et al*. Dynamic point cloud geometry compression using multiscale inter conditional coding[OL]. [2023-06-18]. https://arxiv.org/abs/2301.12165

[21] Sheng X H, Li L, Liu D, *et al*. Deep-PCAC: an end-to-end deep lossy compression framework for point cloud attributes[J]. IEEE Transactions on Multimedia, 2022, 24: 2617-2632

[22] Ballé J, Minnen D, Singh S, *et al*. Variational image compression with a scale hyperprior[C] //Proceedings of the 6th International Conference on Learning Representations. Washington DC: ICLR, 2018: 4961-5007

[23] Sun Z H, Tan Z Y, Sun X Y, *et al*. Spatiotemporal entropy model is all you need for learned video compression[OL]. [2023-06-18]. https://arxiv.org/abs/2104.06083

[24] Biswas S, Liu J, Wong K, *et al*. MuSCLE: multi sweep compression of LiDAR using deep entropy models[C] //Proceedings of the 34th International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc, 2020: Article No.1859

[25] Bégaint J, Racapé F, Feltman S, *et al*. CompressAI: a PyTorch library and evaluation platform for end-to-end compression research[OL]. [2023-06-18]. https://arxiv.org/abs/2011.03029

[26] Lu G, Zhang X Y, Ouyang W L, *et al*. An end-to-end learning framework for video compression[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(10): 3292-3308

[27] Lu G, Cai C L, Zhang X Y, *et al*. Content adaptive and error propagation aware deep video compression[C] //Proceedings of the 16th European Conference on Computer Vision. Heidelberg: Springer, 2020: 456-472

[28] Djelouah A, Campos J, Schaub-Meyer S, *et al*. Neural inter-frame compression for video coding[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2019: 6420-6428

[29] Lu G, Ouyang W L, Xu D, *et al*. DVC: an end-to-end deep video compression framework[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2019: 10998-11007

[30] Hu Z H, Lu G, Xu D. FVC: a new framework towards deep video compression in feature space[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2021: 1502-1511

[31] Wu W X, Wang Z Y, Li Z W, *et al*. PointPWC-Net: cost volume on point clouds for (self-) supervised scene flow estimation[C] //Proceedings of the 16th European Conference on Computer Vision. Heidelberg: Springer, 2020: 88-107

[32] Wei Y, Wang Z Y, Rao Y M, *et al*. PV-RAFT: point-voxel correlation fields for scene flow estimation of point clouds[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2021: 6950-6959

[33] Li B, Zheng C, Giancola S, *et al*. SCTN: sparse convolution--transformer network for scene flow estimation[C] //Proceedings of the 36th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2022: 1254-1262

[34] Xu Y, Lu Y, Wen Z. Owlii dynamic human mesh sequence dataset[OL]. [2023-06-18]. https://mpeg-pcc.org/index.php/pcc-content-database/owlii-dynamic-human-textured-mesh-sequence-dataset