Contents lists available at ScienceDirect

Displays

journal homepage: www.elsevier.com/locate/displa

An End-to-end Dynamic Point Cloud Geometry Compression in Latent Space

Zhaoyi Jiang[a,*], Guoliang Wang[a], Gary K.L. Tam[b], Chao Song[a], Bailin Yang[a], Frederick W.B. Li[c]

[a] College of Computer Science & Information Engineering, Zhejiang Gongshang University, China
[b] Department of Computer Science, Swansea University, UK
[c] Department of Computer Science, University of Durham, UK

ARTICLE INFO

Communicated by S. Sarkar

Keywords:
Dynamic point clouds compression
Geometry encoding
Latent scene flow
Deep entropy model

ABSTRACT

Dynamic point clouds are widely used for 3D data representation in various applications such as immersive and mixed reality, robotics and autonomous driving. However, their irregularity and large scale make efficient compression and transmission a challenge. Existing methods require high bitrates to encode point clouds since temporal correlation is not well considered. This paper proposes an end-to-end dynamic point cloud compression network that operates in latent space, resulting in more accurate motion estimation and more effective motion compensation. Specifically, a multi-scale motion estimation network is introduced to obtain accurate motion vectors. Motion information computed at a coarser level is upsampled and warped to the finer level based on cost volume analysis for motion compensation. Additionally, a residual compression network is designed to mitigate the effects of noise and inaccurate predictions by encoding latent residuals, resulting in smaller conditional entropy and better results. The proposed method achieves an average 12.09% and 14.76% (D2) BD-Rate gain over state-of-the-art Deep Dynamic Point Cloud Compression (D-DPCC) in experimental results. Compared to V-PCC, our framework showed an average improvement of 81.29% (D1) and 77.57% (D2).

1. Introduction

The rapid advancement of sensing and acquisition technology has led to the widespread use of dynamic point cloud (DPC) in various fields, including autonomous driving, virtual reality, intelligent cities, and robotics [1]. Nowadays, different sensors and algorithms enable the capture of complex moving objects and scenes, resulting in large DPCs with intricate details. However, effectively storing and transmitting such large DPCs present challenges, requiring dynamic point cloud compression (DPCC) for efficient utilization in these applications. Unlike regular pixelized videos, DPCs exhibit irregularity, non-uniformity and sparsity, with varying numbers of points in each frame. Consequently, establishing explicit temporal correspondence between points in contiguous DPC frames becomes difficult. This hinders the exploration of temporal correlation. In this paper, our focus is to reduce temporal redundancies in the neural inter-frame geometry compression of DPC, utilizing motion estimation, motion compensation, and a deep entropy model in the latent space.

DPCC methods can be classified into two categories: traditional algorithms and neural methods. Traditional algorithms are rule-based methods that often rely on hand-crafted features and perform inter-prediction on consecutive frames. Due to the limitations of manual feature engineering, the restricted expressiveness and challenges in generalization and adaptability to new dataset, these often lead to suboptimal compression performance [2, 3]. The Moving Picture Experts Group (MPEG) has also approved two prominent point cloud compression standards: video-based point cloud compression (V-PCC) and geometry-based point cloud compression (G-PCC) [1, 4].

*Corresponding author: Tel.: +0-000-000-0000; fax: +0-000-000-0000;

http://dx.doi.org/10.1016/j.displa.xxxx.xx.xxx
Received 1 May 2013; Received in final form 10 May 2013; Accepted 13 May 2013; Available online 15 May 2013

V-PCC projects dynamic point clouds onto 2D images and texture videos, and uses video codecs for compression. However, this approach can encounter challenges with occlusion and shape distortion when mapping 3D patches onto 2D patches. In contrast, G-PCC employs an octree-based model and offers advantages in multiple resolutions and contextual range. However, it is limited to encoding static point clouds (SPC) only. To address temporal redundancy, Kammerl et al. proposed a double-tree structure to calculate the disparity between the octrees of consecutive frames and enable inter-prediction [5]. Other traditional algorithms, such as the work by Thanou et al. [6], leverage graph-based representations and feature matching for motion estimation. Similarly, Mekuria et al. [7] rely on hand-crafted features and rigid transformations. However, these traditional approaches heavily rely on point space operations, which can lead to inaccurate motion estimation and less effective motion compensation owing to the irregular and sparse nature of DPC [8].

Inspired by the success of deep learning techniques in image and video compression, recent research has focused on neural point cloud compression, surpassing traditional rule-based algorithms. For compressing static point clouds (SPC), learning-based methods can be classified into point-based, voxel-based, and octree-based approaches. Wang et al. [9] proposed a voxel-based method that utilizes the sparsity of point clouds to losslessly compress geometric occupancy using autoencoders based on sparse convolution. In an effort to extend learning-based SPC compression techniques to DPCC, Akhtar et al. [10] designed a predictor that performs convolution on target coordinates. They incorporate multi-scale features from the previous frame for motion compensation. However, their network lacks an explicit motion estimation and motion compensation (MEMC) network to guide inter-prediction. Fan et al. [11] proposed D-DPCC, which includes an end-to-end MEMC module. This module computes latent features of two successive frames for subsequent motion estimation. Moreover, they introduced 3D adaptively weighted interpolation (3DAWI) for motion compensation. While D-DPCC outperforms other DPCC frameworks, it employs a relatively simple concatenate operation for inter-prediction which limits the accuracy for fine motion prediction between consecutive frames. The relatively simple factorized entropy model for residual compression also leaves room for further encoding of redundancies. All in all, we observe a crucial limitation in current DPCC techniques, namely their inadequate modeling of temporal dynamics.

To address this limitation, we propose a novel end-to-end DPCC network that operates in the latent space and introduces a more effective MEMC network and deep entropy model. Firstly, a feature extraction module is designed to generate the latent representations of point clouds. Then, Our MEMC network leverages a multi-scale latent scene flow (LSF) estimation module to estimate finer motion information. To do so, we specifically incorporate a warping operation and cost volume layer in the motion estimation module, allowing it to focus on the smaller motion between the warped previous frame and current frame whilst the multi-scale network adapts to non-rigid or large motion. After calculating the motion vector, the motion compensation is carried out by passing the warped frame, previous frame, and motion vector together into the compensation module for prediction. To encode the residual efficiently, we propose a deep entropy model, combining spatial hyper-prior and temporal prior, to estimate the parameters for Gaussian entropy model. The model enhances compression efficiency, eliminates temporal redundancy, and maintains accuracy of reconstructed frames. All these successfully improve the modelling of temporal redundancies and result in an innovative DPCC with improved performance. Our main contributions are:

- We propose a novel end-to-end deep learning framework for 3D DPCC, which operates entirely in the latent space.
- We propose a Latent Scene Flow (LSF) module for our MEMC network. LSF enables the learning of motion vectors in latent space. By incorporating a multi-scale scene flow network, we can effectively and efficiently obtain accurate motion information for large-scale point clouds.
- We introduce a deep entropy model which joins the spatial hyper-prior and temporal prior for residual compression. It can enhance compression efficiency and improve reconstructed frame accuracy.
- To demonstrate the effectiveness of our proposed framework, we conduct experiments on the 8iVFB dataset [12] recommended by MPEG and show that our framework outperforms D-DPCC [11] by achieving state-of-the-art compression performance.

2. Related Work

2.1. Static Point Cloud Geometry Compression

Traditional method for compressing static point clouds involves creating a tree-based data structure [13] to reduce geometric redundancy. However, this approach has limitations in terms of reducing the size of point clouds. To address this issue, recent works [1] [14] have introduced handcrafted contexts for arithmetically encoded octrees to compress bitstreams. Although these methods have shown promising results, they are not efficient in capturing complex dependencies and correlations between nodes in the tree. OctSqueeze [2] addresses this problem by modeling dependencies between nodes and their multiple ancestral nodes as context information using a deep learning-based entropy model. However, this method does not consider the dependence between sibling nodes. To address this limitation, OctAttention [15] introduced the attention mechanism and extended the context acceptance domain to capture strong dependencies between sibling nodes. Additionally, voxel-based methods have been proposed, where the point cloud is quantified and the voxel occupancy rate is classified by neural network VoxelDNN[16]. MSVoxelDNN [17] combines octree and voxelization methods to adaptively divide point clouds and reduce their sparsity. Sparse convolution [18] has also been used to further reduce memory requirements and speed up computation, resulting in better compression performance [9]. However, these methods only consider spatial redundancy and cannot be easily extended to DPCC.

2.2. Dynamic Point Cloud Geometry Compression

Temporal redundancy is a challenging problem in DPCC, as the lack of time correspondence in the point cloud makes the redundancy of time difficult to remove. Although motion estimates can effectively remove the temporal redundancy, the points of two adjacent point clouds do not correspond one-to-one, making motion estimation challenging. Some existing methods, such as [6], have used graph-based representations and feature matching to estimate motion, while others, like [7], rely on handcrafted features and rigid transformations. Lossless coding methods, such as the 4D context used in [19], avoid the estimation of motion vectors but require a higher bit rate. D-DPCC [11] proposed an end-to-end deep DPCC framework based on a lossy autoencoder. However, their method uses a simple convolution network to obtain the original flow embedding between two frames. Despite the proposed multi-scale fusion module to refine the motion vectors, the limited depth of the CNN would limit the accuracy of fine motion. In summary, these methods suffer from limitations, such as the reliance on handcrafted features, rigid transformations, or inaccurate motion estimation to effectively encode temporal redundancies. To address this limitation, we follow deep learning approach and propose a Latent Scene Flow module to better estimate motion.

2.3. Video-Based Dynamic Point Cloud Compression

Recent research on DPCC technology has mostly focused on Video-based Point Cloud Compression (V-PCC), which has shown significant improvement compared to previous geometric image-based methods. However, 2D projection technology used in V-PCC can disrupt the motion continuity of the original 3D object, leading to reduced efficiency of inter-frame coding. To address this, [20] proposed a method that calculates motion vectors (MVs) using 3D motion and 3D-to-2D correspondences, and uses auxiliary information to estimate 2D geometry and attributes. However, downsampling the image video during compression can generate more noise points for the projected 3D point cloud. Researchers have explored the use of convolutional neural networks to address this issue, improving the accuracy of the occupied map video by transforming it into a binary segmentation problem (where the pixel value is either 0 or 1) [21]. The V-PCC method requires projecting 3D onto 2D images or videos, which can lead to problems such as occlusion and shape distortion, resulting in inaccurate motion estimation and increased residual coding. To tackle these technical challenges, we propose a new approach to DPCC based on deep learning.

2.4. Scene Flow Estimation

Scene flow estimation is an extension of 2D optical flow estimation in video compression, and using it in motion estimation can yield more accurate motion vectors (MVs). Several approaches, such as FlowNet3D [22], HPLFlowNet [23], and PointPWC-Net [24], have been proposed for this purpose. FlowNet3D [22] extracts features from point clouds [25] and uses stream embedding to fuse information from successive point clouds. HPLFlowNet [23] incorporates Bilateral Convolutional Layers to recover structural information from unstructured point clouds and fuses information from two consecutive point clouds, while PointPWC-Net [24] adopts a coarse-to-fine strategy, using cost volumes, upsampling, and twist layers to process point clouds. However, these methods estimate scene flow in the original space point cloud and are not applicable to large-scale point clouds. To address this issue, we propose a new approach that utilizes sparse convolution in the latent space to compress DPCs. Also, a multi-scale framework is employed to estimate motion and obtain finer scene flow.

2.5. Deep Learning Based Video Compression

DVC [26] proposed the first end-to-end video compression model, combining the classical architecture of traditional video compression methods and the nonlinear representation power of neural networks. However, it suffered from inaccurate current frame prediction and high residuals. To address this issue, [27] generated more precise current frame predictions using multiple reference frames and associated multiple MV fields, leading to reduced residuals. This approach aids in predicting MVs, thus decreasing the coding cost of the MV field. Instead of using the motion prediction module, [28] introduced an entropy model framework to estimate spatio-temporal redundancy in the feature space. By conducting all main operations in the feature space, such as motion estimation, motion compression, motion compensation, and residual compression, [8] achieved superior performance compared to DVC.

3. Methodology

3.1. Overview

The input of our method is a dynamic point cloud sequence $P = \{P_1, P_2, \ldots, P_{t-1}, P_t, \ldots\}$, where P_t represents the point cloud frame at a time step t. We utilize sparse convolution to improve the efficiency of tensor processing, resulting in a more compact representation for each frame of the DPC sequence. This is achieved by converting each frame into a sparse tensor, represented by $P_t = \{C_{P_t}, F_{P_t}\}$, where C_{P_t} represents the coordinate matrices of the point cloud, and F_{P_t} represents the associated feature matrices with all-one vectors to indicate voxel occupancy. Our framework reduces the bit-rate consumption of the entropy code by analyzing the correlation between the current frame P_t and the previously reconstructed frame $\hat{P}_{t-1}$. Our proposed framework, illustrated in Figure 1, consists of five modules:

1) Feature Extraction. To generate the latent representations, the input frame P_t is encoded as Y_t, and the previously decoded frame $\hat{P}_{t-1}$ is encoded as $\hat{Y}_{t-1}$.

2) Motion Estimation. The LSF estimation module then estimates the motion between Y_t and $\hat{Y}_{t-1}$ in the latent space to obtain the corresponding MVs M_t.

3) Motion Compression and Compensation. The decoded MVs ($\hat{M}_t$), along with Y_t and $\hat{Y}_{t-1}$, are then processed by the motion compensation module to generate the predicted frame ($\bar{Y}_t$) for the current frame.

4) Residual Compression. The residual R_t between the current frame Y_t and the predicted frame $\bar{Y}_t$ will be compressed by a deep entropy model to enhance the reconstruction quality.

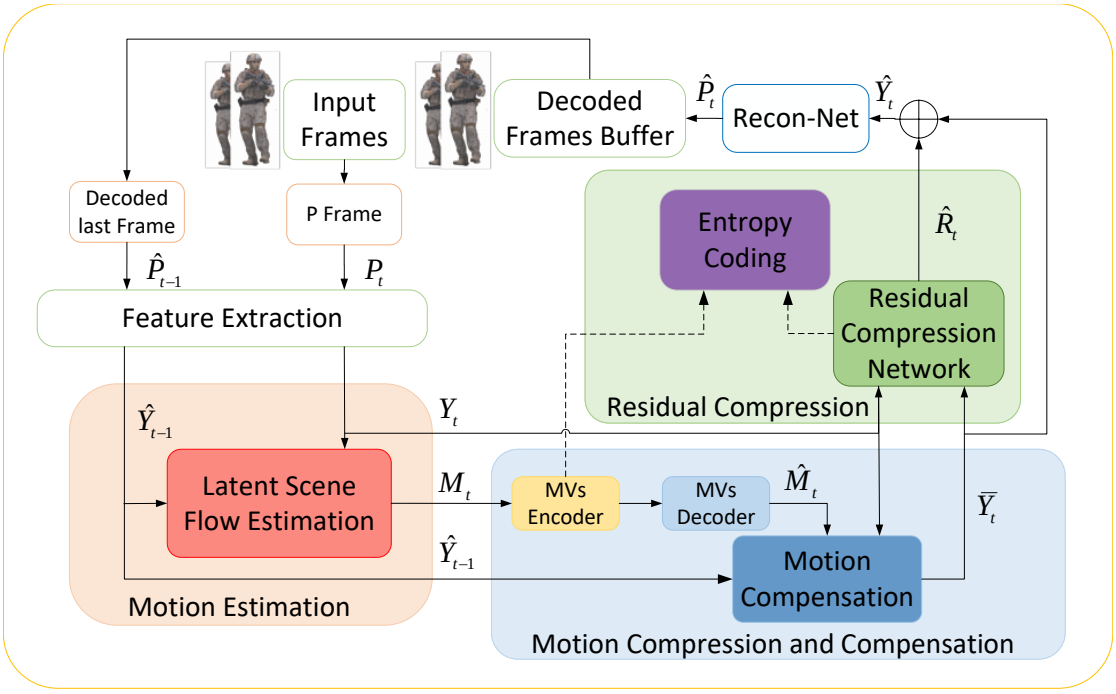


Fig. 1. Our dynamic point cloud geometric compression framework involves several stages, starting with the selection of the previous frame $\hat{P}_{t-1}$ as a reference for encoding the current frame P_t [29]. We then perform feature extraction, motion estimation, motion compression, motion compensation, and residual compression in a specific order. In the latent space, we extract features Y_t and $\hat{Y}_{t-1}$, and obtain latent motion vectors M_t and reconstructed motion vectors $\hat{M}_t$. After motion compensation, we obtain the prediction $\bar{Y}_t$ of Y_t. The reconstruction residual is denoted by $\hat{R}_t$, and the reconstruction value of Y_t in the latent space is $\hat{Y}_t$. Ultimately, we obtain the reconstructed current frame $\hat{P}_t$.

5) Point Cloud Reconstruction. After incorporating the reconstructed residual $\hat{R}_t$ into the predicted frame $\bar{Y}_t$, we obtain the latent reconstructed frame $\hat{Y}_t$. Subsequently, the decoded frame $\hat{P}_t$ is reconstructed using a hierarchical reconstruction approach based on binary classification. This method effectively trims erroneous voxels, ensuring the preservation of details in the reconstructed point cloud [9].

Our framework's advantage lies in its ability to accurately preserve the details of the reconstructed point cloud while minimizing the bit-rate consumption of the entropy code.

3.2. Feature Extraction

We propose a novel framework that utilizes the powerful representation capability of deep features to reduce spatial or temporal redundancy in DPC through motion compensation and residual compression in the latent space. To generate the latent representations, we encode the input frame P_t and the previously reconstructed frame $\hat{P}_{t-1}$ as Y_t and $\hat{Y}_{t-1}$, respectively. To enable efficient feature extraction, we adopt sparse convolution-based CNN down-sampling similar to [11] and [9]. The feature extraction module includes two sparse convolution layers with a step of 2 for down-sampling, followed by three Inception-Residual Network units (as shown in Figure 2(b)) for effective local feature aggregation. The use of sparse convolution offers the advantage of reducing bit-rate consumption for entropy coding by analyzing the correlation between P_t and $\hat{P}_{t-1}$.

3.3. Motion Estimation

D-DPCC utilizes the Inter Prediction method for DPCC. However, its approach of connecting two frames in series

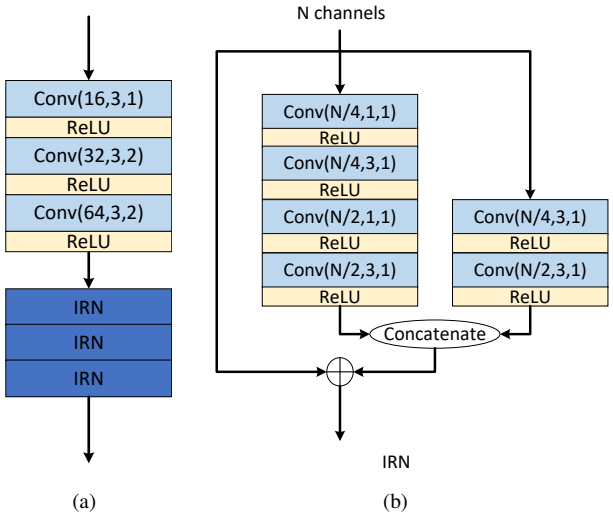
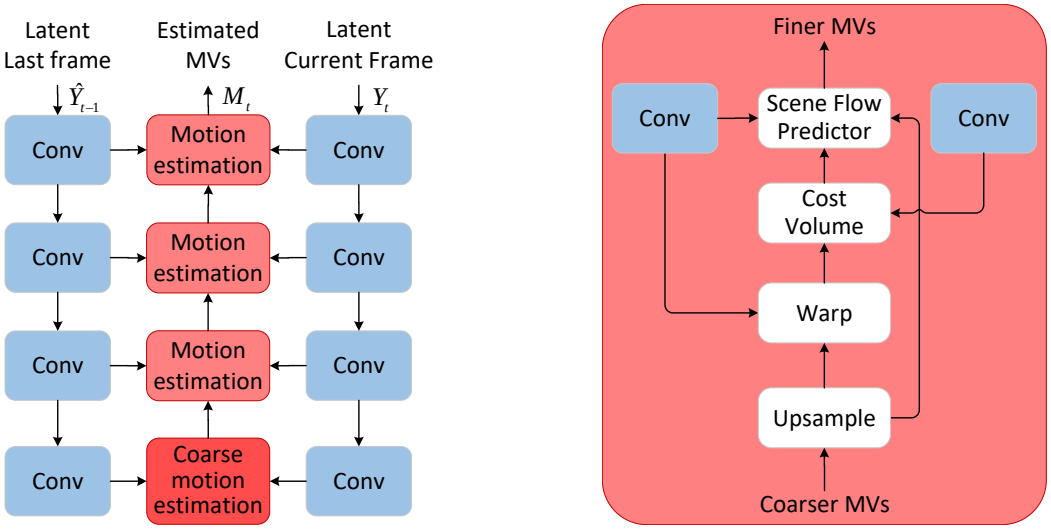


Fig. 2. Feature Extraction. (a) The architecture of sparse convolution is represented by Conv(c, n, s), where c is the output channel, n is the kernel size, and s is the step size. ReLU denotes the Rectified Linear Unit. We employ Inception-Residual Network (IRN) units for local feature aggregation. (b) Diagram of the IRN unit. The Concatenate module is utilized to concatenate feature channels.

through the convolutional network to obtain the original stream embedding lacks accuracy, resulting in lower residual compression efficiency. In order to overcome this limitation, we introduce the scene flow module for MV estimation.

To this end, we observe that most existing MV estimation methods have their own drawbacks. Coarse-to-fine methods



(a) Latent Multi-scale Scene Flow Estimation (b) Motion estimation in one level

Fig. 3. (a) Our scene flow estimation adopts a multi-scale, bottom-up approach. (b) The motion estimation framework in one level includes an upsampling module to warp the previous frame using coarser MVs, a cost volume layer for patch-to-patch cost aggregation, and a scene flow predictor layer for finer MV prediction. The coarse motion estimation module takes two sparse tensors as input and directly outputs finer MVs using the scene flow predictor network.

can estimate large-scale scene flows but are time-consuming and struggle with fast-moving small objects. On the other hand, single-scale methods can accurately estimate the scene flow, but they require computing a similarity matrix of N * N, where N is the number of points in the current frame. Such methods are not suitable for DPCs with millions of points per frame. To address this challenge, our proposed framework conducts scene flow estimation in the latent space, which serves as a higher-dimensional abstraction of the original data. This allows for efficient computation of the MVs without the need to calculate the coordinates of points in the original point cloud. To our knowledge, all existing scene flow estimation methods operate in the original space of the point cloud and cannot estimate the MVs based on the latent representation.

To efficiently handle large-scale DPCs, our proposed framework estimates the motion using latent tensors of adjacent frames and sparse convolution. The advantage of using sparse convolution lies in its ability to correlate coordinates in the downscaled latent space while matching the features of the flow embedding layer, making it an ideal solution to compress DPCs. Our multi-scale framework also allows for finer scene flow estimation. As depicted in Figure 3, the framework adopts a multi-scale, bottom-up approach. In each level of motion estimation module, the previous frame of point cloud is warped by the upsampled coarser MVs, then the warped previous frame and the current frame are concatenated to aggregate the cost in a patch-to-patch manner using the cost volume module and then obtains finer MVs using the Scene Flow Predictor module. The coarse motion estimation module takes two sparse tensors as input and directly outputs finer MVs using the scene flow predictor network. In our design, we specifically incorporate a warping operation and cost volume module in the motion estimation module, allowing it to focus on the smaller motion between the warped previous frame and current frame whilst the multi-scale

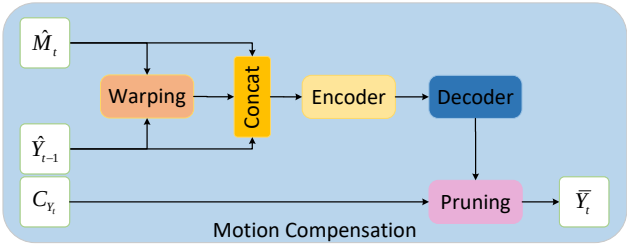


Fig. 4. Motion Compensation Module. Y_t and $\hat{Y}_{t-1}$ are the corresponding latent representations in the latent space. $\hat{M}_t$ represents the reconstructed MVs, and $\bar{Y}_t$ is the predicted value of Y_t. C_{Y_t} is the coordinate of the current frame. The Pruning step involves aligning the current frame by pruning the sparse tensor of the rough predicted frame with the coordinates of the latent current frame.

network adapts to non-rigid or large motion. By doing so, our scene flow modules allow finer MVs to be predicted. Unlike [24], our approach uses the characteristic of sparse tensor aggregation to avoid the expensive KNN search.

3.4. Motion Compression and Motion Compensation

We propose a lossy compression network based on auto-encoder to compress the MVs, which are then used to calculate the predicted frame in our motion compensation framework (shown in Figure 4). As point clouds are unstructured and sparse, matching between adjacent frames can be challenging, leading to larger errors and reducing compression efficiency. To address this, we use the interpolation method [25] to warp the previous frame and obtain the latent warped frame of the current frame. We also incorporate temporal prior [26] by splicing the warped frame, the latent previous frame, and the reconstructed MVs to improve prediction accuracy and reduce residual compression. The rough prediction frame is obtained through a sparse convolution network, and 'pruning' is performed on the coordinates of the latent current frame to align them and ob-

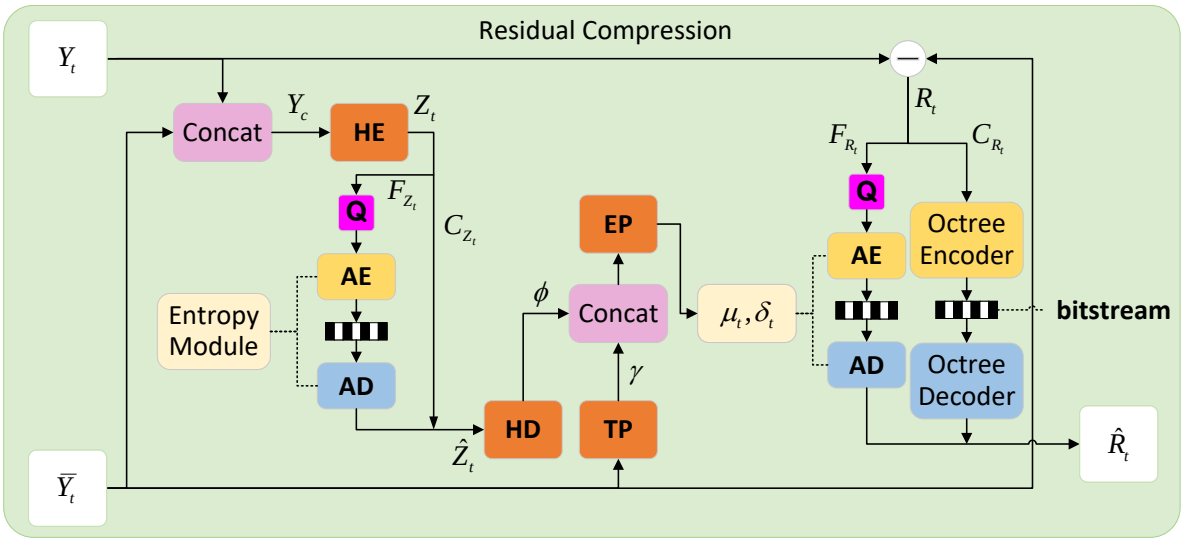


Fig. 5. Overall framework for residual compression, which includes several components such as quantization (Q), arithmetic encoding (AE), arithmetic decoding (AD), hyper encoder (HE), hyper decoder (HD), temporal prior (TP), and entropy parameter (EP). The variables used in this framework include sparse tensors $Y_t, \bar{Y}_t, R_t, \hat{R}_t, Y_c, Z_t$, and $\hat{Z}_t$. Y_t and $\bar{Y}_t$ are latent current frame and latent predicted frame respectively. The residual feature between Y_t and $\bar{Y}_t$ is represented by R_t, and $\hat{R}_t$ is the decoded residual. Y_c is the concatenation of Y_t and $\bar{Y}_t$, Z_t is the hyper latent obtained from Y_c, and $\hat{Z}_t$ is the decoded hyper latent. Additionally, C_{R_t}, F_{R_t}, and F_{Z_t} are needed for encoding. The parameters μ_i and δ_i are obtained using $g_{EP}(\phi, \gamma; \theta_{EP})$, where ϕ is obtained using $g_{HD}(\hat{z}_t; \theta_{HD})$, and γ is obtained using $g_{TP}(\bar{Y}_t; \theta_{TP})$ in the Gaussian Entropy model [30]. More network details are shown in Figure 6.

tain the fine prediction frame. Importantly, we use the coordinate information of the current frame in both the interpolation and the final 'pruning' module, which is losslessly compressed and transmitted to the decoder at the beginning of the network framework. Our proposed method provides more accurate time information, reduces matching errors, and improves overall prediction accuracy.

3.5. Residual Compression

Despite various compression methods, distortions still exist in the prediction frame of point clouds. Traditional point cloud compression methods such as XOR [5] and pixel-based compression methods like [26] have limitations in capturing the subtle movement of points in the octree structure, leading to large changes in the underlying octree. In recent studies, encoding residuals in the latent space has shown better results for both video compression methods [8] and point cloud compression methods [9]. This is because downsampled point clouds have similar geometric information between two frames, and the high-dimensional feature information in the latent space can reduce the impact of noise in the residual network. Motivated by the spatio-temporal entropy model [28], we introduce a deep entropy model for residual compression, which can further enhance the compression efficiency and maintain the accuracy of the reconstructed frames.

The proposed residual compression, shown in Figure 5, is inspired by the autoencoder-based image compression works [30] which employ a hyperprior entropy model to more accurately model the probability distribution of latent variables. In our framework, the Hyper Encoder, Hyper Decoder, Temporal Prior, and Entropy Parameter are multi-layer perceptrons with different network parameters based on sparse convolution. The advantage of our approach is that it enables the hyper-prior spatial and temporal information to be fused to estimate the parameters (mean μ and scale δ) required by the Gaussian entropy model [30]. Specifically, the current and predicted frames are concatenated to obtain Y_c as input for the hyper-prior spatial entropy model, which is then used to obtain the hyper latent Z_t through Hyper Encoder coding. The resulting quantized attributes F_{Z_t} are then factorized entropy encoded to obtain the bitstream. Similarly, the hyper-prior temporal entropy is obtained by passing the predicted frame $\bar{Y}_t$ through the Temporal Prior module. The Entropy Parameter then fuses the hyper-prior spatial and temporal information to estimate the parameters (mean μ and scale δ) required by the Gaussian entropy model. The residual R_t, obtained from Y_t and $\bar{Y}_t$, is losslessly compressed using a traditional octree encoder [1] in MPEG for the geometry coordinates C_{R_t}, and using arithmetic encoding [31] for the quantized attributes $\hat{F}_{R_t}$. The probability distribution is modeled as a Gaussian distribution using the obtained

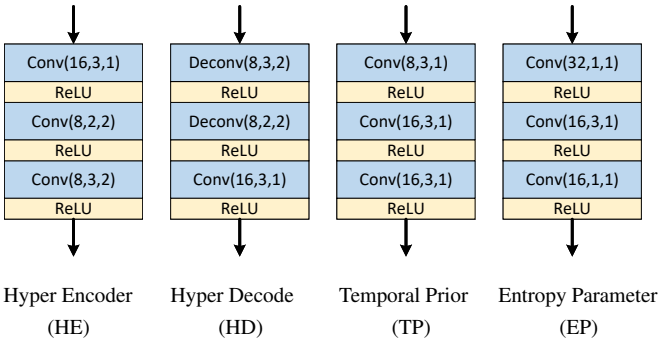


Fig. 6. Details of each module in the residual compression network. 'Conv' denotes sparse convolution followed by the number of output channels, the kernel size, and the stride. The 'Deconv' corresponds to upsampled convolutions.

parameters, and $\hat{R}_t$ can be reconstructed.

3.6. Point cloud Reconstruction

The reconstructed residuals $\hat{R}_t$ and $\bar{Y}_t$ are added up by the decoder to reconstruct the current frame $\hat{P}_t$. The decoder here is a hierarchical reconstruction based on binary classification, upsampled by three sparse transposed convolutional layers with a step size of 2. The three classification networks use a binary classification method to trim the wrong voxels to guarantee the reconstructed point cloud details [9].

3.7. Loss Function

Our proposed framework takes advantage of a rate-distortion joint loss function to optimize the loss L:

$$L = R + \lambda D = L_{\hat{F}_{M_t}} + L_{\hat{F}_{Z_t}} + L_{\hat{F}_{R_t}} + \lambda \left(\frac{1}{M} \sum_{j=1}^{M} L_{BCE}^{j} \right) \quad (1)$$

with R representing the bits per point (bpp) by encoding the current frame P_t. We utilize a multiscale loss (D) [9] to measure the distortion between the decoded current frame $\hat{P}_t$ and P_t, with λ serving as a Lagrange multiplier to trade off distortion against rate. $L_{\hat{F}_{M_t}}, L_{\hat{F}_{Z_t}}$ and $L_{\hat{F}_{R_t}}$ represent the number of bits used to encode $\hat{F}_{M_t}, \hat{F}_{Z_t}$ and $\hat{F}_{R_t}$ respectively, with M representing the number of scales. The binary cross entropy (BCE) loss is applied, with L_{BCE}^{j} denoting the j-th scale BCE. Binary classification is used to determine whether generated voxels are occupied or not. Our framework offers significant advantages in terms of rate-distortion optimization and effective binary classification.

$$L_{\vartheta} = \frac{1}{N_{\vartheta}} \sum_{i=1}^{N_{\vartheta}} -\log_2 \left(p_{\vartheta} \right), \vartheta \in \left\{ \hat{F}_{M_{t'}}, \widehat{F}_{Z_{t'}}, \hat{F}_{R_t} \right\} \quad (2)$$

where L_{ϑ} can be calculated using the corresponding probability density function, and N_{ϑ} represents the number of corresponding attributes.

$$L_{BCE}^{j} = \frac{-1}{N_{P_t}} \left\{ \sum_{i} \left[o_i \log_2 p_i + (1 - o_i) \log_2 (1 - p_i) \right] \right\} \quad (3)$$

where o_i represents the occupancy label of the corresponding voxel, with 0 indicating empty and 1 indicating occupied. p_i denotes the probability of a voxel being occupied.

4. Experiments

4.1. Experimental Settings

Training strategy. To ensure a fair comparison, we adopt a similar training strategy as the most advanced method D-DPCC [11], using the same training and testing datasets. Our model is optimized using the Adam optimizer [32] with = (0.9, 0.999), along with a learning rate scheduler that reduces the learning rate by a factor of 0.7 every 10 epochs. To obtain the optimal model parameters for different rate points, we trained our model for 100 epochs with $\lambda = 3, 4, 5, 7, 9$ as the training parameters. During training, the batch size was set to 1 and the learning rate was 0.0008.

Training Dataset. We train our model using the Owlii dynamic human point cloud dataset [33], consisting of four point cloud sequences: basketball_player, dancer, exercise, and model. Each sequence spans 20 seconds with a frame rate of 30 frames per second, resulting in 600 frames in total, each with a resolution of 11 bits. To speed up training, reduce storage consumption, and enhance model robustness, we quantized the dataset from 11-bit precision to 9-bit precision.

Testing Datasets. In order to assess the effectiveness of our proposed method, we employ the 8i Voxelized Full Bodies dataset [12] which is potential test material for MPEG and/or JPEG standardization efforts. This dataset comprises four point cloud sequences, namely Soldier, Loot, Redandblack, and Longdress, each featuring a full-body human subject captured at 30 fps over a period of 10s. The spatial resolution of each frame is 1024*1024*1024, and the number of points in each frame ranges from 700,000 to 1,100,000.

Evaluation Metrics. We measure the effectiveness of our proposed framework using bpp as the metric for the average number of bits used per point in each frame. To evaluate the quality of our compressed point cloud sequences, we use point-to-point error (D1 ↑) and point-to-surface error (D2 ↑) [34]. These metrics help us derive the peak signal-to-noise ratio (PSNR) and measure the distortion between the decoded and ground-truth frames. Our goal is to achieve the highest PSNR possible under the same bpp constraint.

Baseline. We compare our proposed method with several state-of-the-art compression methods, including D-DPCC, V-PCC Test Model v13, and a static point cloud geometric compression method proposed in [9].

4.2. Experimental Results

Figure 7 displays the rate distortion (RD) curves of various compression techniques. Our proposed method outperforms other approaches significantly in the Soldier and Longdress point cloud sequences, with some improvement observed in the Loot dataset as well. Furthermore, the results for the Redandblack dataset are comparable to those of the D-DPCC method. Our framework achieves superior performance due to the LSF and motion compensation modules compared to the state-of-the-art methods.

The table 1 summarizes the BD-rate gains of our proposed framework. Compared to D-DPCC, we achieved an average improvement of 12.09% (D1) and 14.76% (D2). Furthermore, compared to V-PCC, our framework showed an average improvement of 81.29% (D1) and 77.57% (D2). Our method shows significant coding efficiency improvement over D-DPCC for point cloud sequences with a compact model, particularly for the datasets Soldier and Longdress. Our LSF approach demonstrated superior performance compared to the Inter Prediction method of D-DPCC and produced more accurate motion vectors. However, for the dataset Redandblack, which consists of a point cloud sequence of a woman wearing a skirt, there is a significant gap between the skirt and the human body. To speed up encoding and decoding time and save space cost, we traded off the accuracy of the scene flow network, which resulted in comparable results to D-DPCC. In future work, we aim to op-

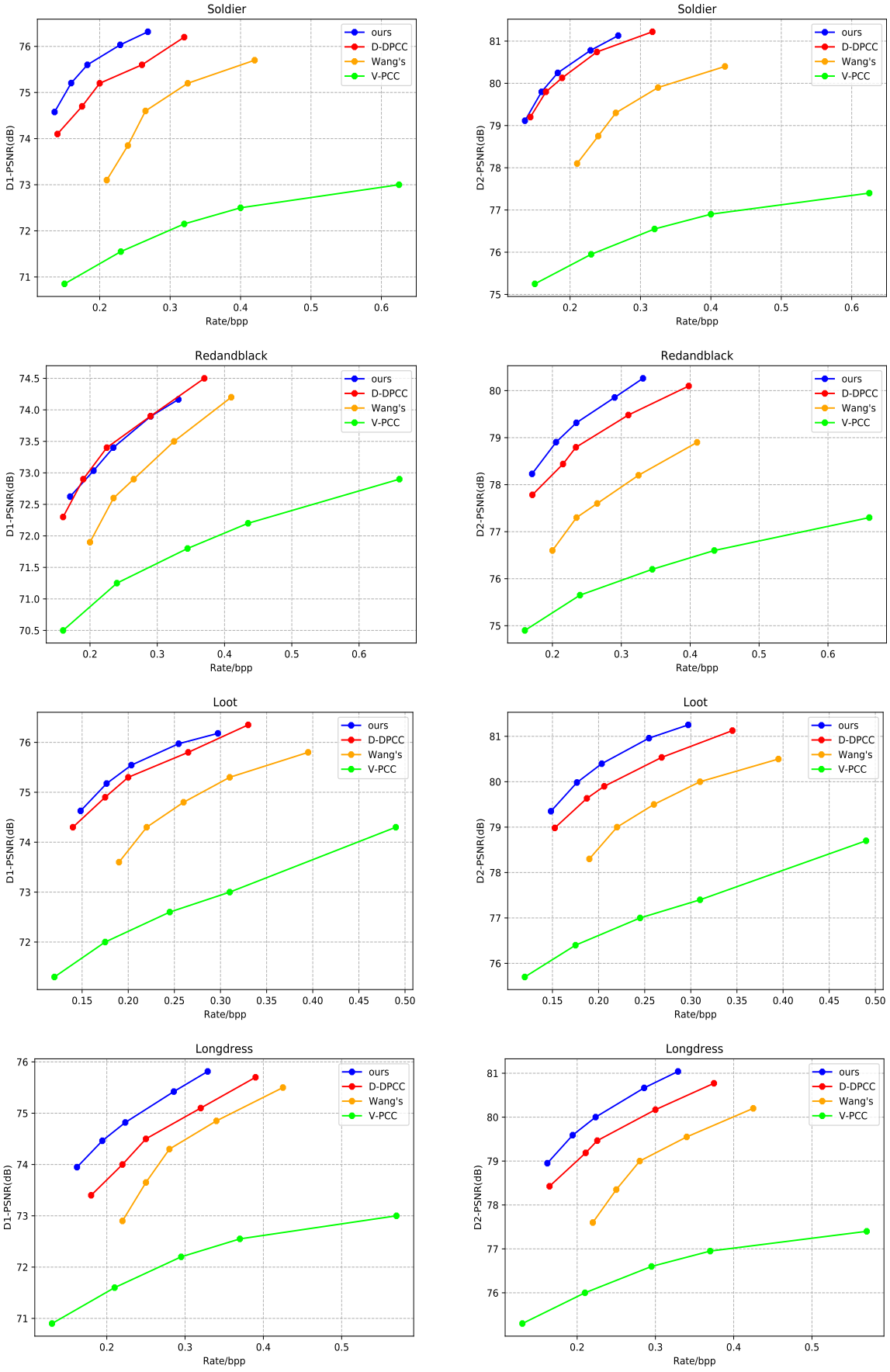


Fig. 7. Rate-distortion (RD) curves comparing our proposed codec with other methods, using the Soldier, Redandblack, Loot, and Longdress sequences as test data. The graphs depicts results of PSNR measurements based on point-to-point error (D1) as in the first column and point-to-surface error (D2) as in the second column.

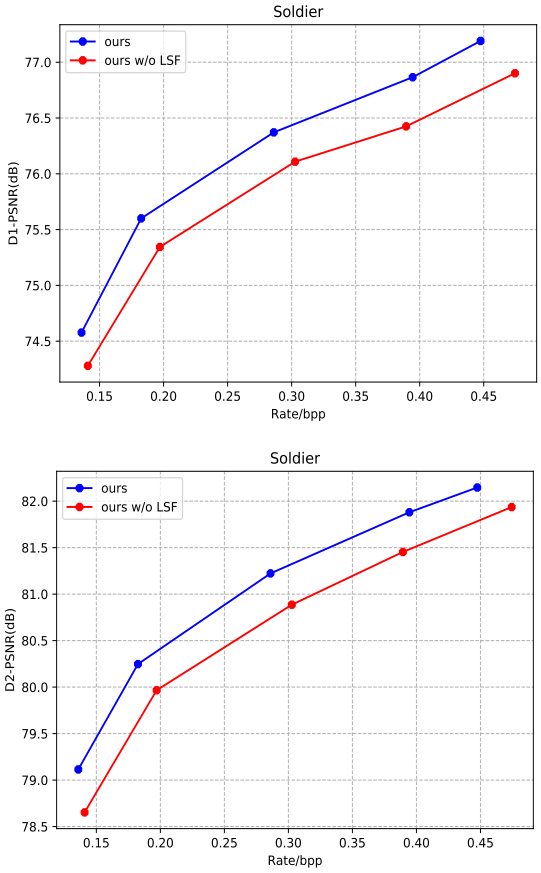


Fig. 8. An ablation study was conducted on the Soldier point cloud sequence to evaluate the impact of the Local Structure-aware Fusion (LSF) module. It is noteworthy that the 'ours w/o LSF' method also utilizes the scene flow network but estimates motion vectors (MVs) in the original space.

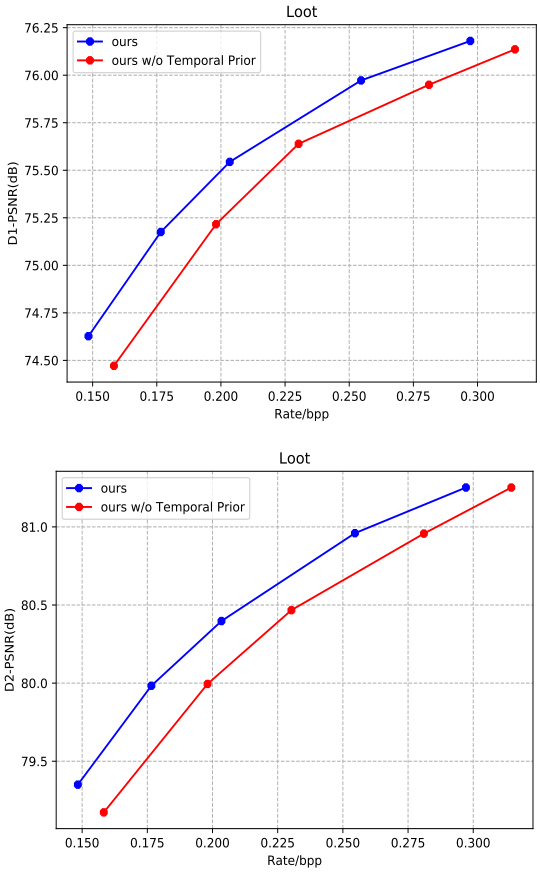


Fig. 9. An ablation study was conducted on the Loot point cloud sequence to evaluate the effectiveness of the motion compensation module with and without temporal prior. The method without temporal prior refers to the direct use of warped frames as the prediction frames for the current frame in the motion compensation module.

timize the scene flow network to improve the prediction accuracy. Despite this implementation issue, the overall results show significant double-digit improvements due to our emphasis to better model temporal redundancies with scene flow.

Sequence	D-DPCC		Wang's		V-PCC	
	D1 ↓	D2 ↓	D1 ↓	D2 ↓	D1 ↓	D2 ↓
Soldier	-21.44	-6.22	-48.92	-47.29	-94.37	-80.47
Redandblack	2.02	-16.72	-26.06	-47.89	-68.41	-74.63
Loot	-8.52	-17.52	-40.13	-41.76	-80.58	-78.41
Longdress	-20.41	-18.60	-34.74	-42.43	-81.81	-76.78
Average	-12.09	-14.76	-37.46	-44.84	-81.29	-77.57

Table 1. Improvements in BD-Rate (%) compared to D-DPCC, Wang's and V-PCC (inter) framework. Negative values indicate bit-rate savings, while positive values indicate higher bit-rate costs. Smaller values (↓) indicate greater improvement over the compared method.

Component \ Method	Ours(ms)	Ours w/o LSF(ms)
Feature Extraction	83	0
Sample	0	204
Scene Flow	369	922
Upsample	0	137
Total	452	1263

Table 2. Runtime in milliseconds. The runtime of the LSF module and the scene flow method in the original space is compared (average on the whole dataset). It is noteworthy that the 'ours w/o LSF' method also utilizes the scene flow network but estimates motion vectors (MVs) in the original space.

4.3. Analysis and Ablation Study

Effectiveness of Latent Scene Flow. The scene flow network is proficient in estimating motion vectors (MVs) accurately between two consecutive frames of point clouds. Since deep features possess robust representation in various applications, we have chosen to implement scene flow modules of point clouds in the latent space. We visualize the estimated MVs in the figure 11. The part of the red virtual coil in the MVs in the figure represents the movement of the point cloud. It can be seen from the figure that D-DPCC can only estimate some large movements, and some fine movements cannot be well estimated, which will increase the burden of residual compression. Figure 8 presents the D1 and D2 rate-distortion curves' comparison results with/without the Latent Scene Flow (LSF) module. The findings indicate that the LSF module significantly enhances the original spatial scene flow estimation, and obtains excellent performance for each bit per pixel (BPP).

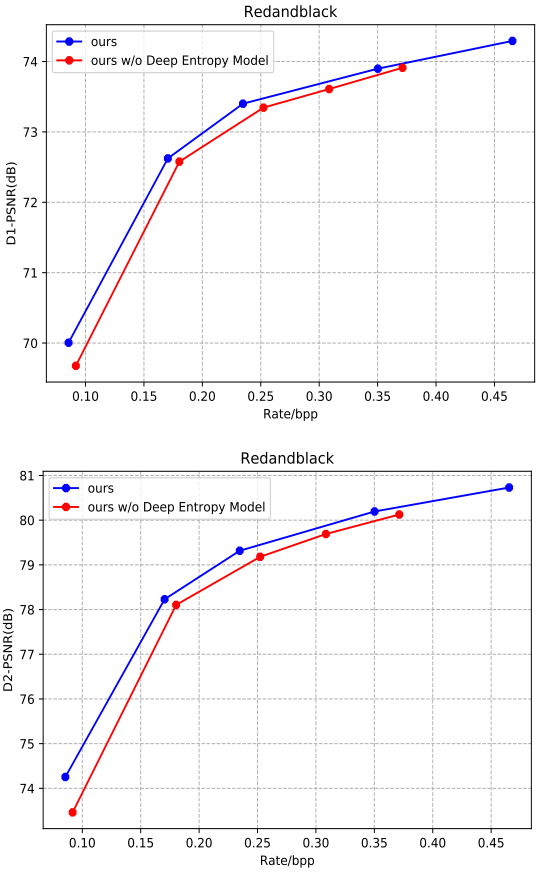


Fig. 10. An ablation study was conducted on the Redandblack point cloud sequence to evaluate the effectiveness of the entropy model with and without deep entropy model. The method without deep entropy model refers to the factorized entropy model[31] used in D-DPCC.

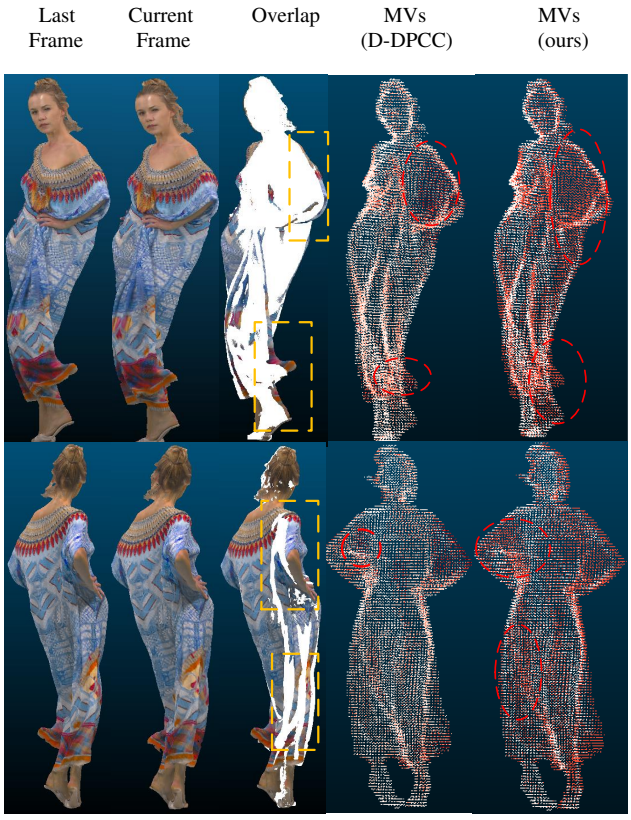


Fig. 11. Visualization of estimated motion vectors (MVs). The left two columns are the 34th and 35th frames of the Longdress point cloud sequence. The middle column shows the result of overlapping the previous frame and the current frame, where the yellow dashed box indicates the area with obvious movement. In order to facilitate observation, the color attributes of the current frame are hidden. The two columns on the right are the MVs estimated by D-DPCC and ours respectively, where the red dotted box represents the position of the motion. In the comparison in the second row, due to the overlap of point cloud visualization colors, the displayed MVs are oriented towards the front of the character.

Moreover, the LSF module's computation time is reduced by more than half, as the original spatial scene flow estimation involves sampling and interpolation steps. The comparison of runtime is shown in the table 2. The improved performance validates the efficacy of the LSF module.

Effectiveness of motion compensation. In the motion compensation module, we employ motion vectors (MVs) to deform the previous frame and obtain the warped frame. Then, we introduce time information into the warped frame within the motion compensation module. Figure 9 presents the comparison results of the D1 and D2 rate-distortion curves with/without temporal prior. Notably, the motion compensation method of refining the warped frame by splicing MVs and the previous reconstruction frame enhances the efficiency compared to the approach lacking temporal prior. In particular, when using the same hyperparameters λ, we observed that the reconstruction accuracy of the point cloud sequence remains similar, while the cost of bits per pixel (BPP) significantly increases. Our motion compensation module provides more precise time information to refine the warped frame further, thereby reducing the matching error between two consecutive frames of voxelized point cloud and improving the predicted frame's accuracy.

Effectiveness of Entropy Model. In the residual compression module, we use the deep entropy model which joins the spatial hyper-prior and temporal prior. The results are shown in the Figure 10. Using the spatial and temporal prior information of the point cloud in the latent space, the deep entropy model can further improve the compression performance, especially at low bpp.

4.4. Visualization Analysis

We compared the reconstruction results with D-DPCC, and the results are shown in the figure 12. It can be found that when the compression ratios of ours and D-DPCC are both set to 0.25 bit rates per pixel (bpp), the reconstruction results have some distortion, as shown in the enlarged yellow circle in the figure. Obviously, our reconstruction effect is better than D-DPCC. As shown in the figure 13, we compared the reconstruction results of our method under different bpps. Comparing the previous two images, we can clearly see that the facial details of the characters become clear. As the bpp increases, the protrusions and missing on the surface of the reconstructed point cloud gradually decrease.

Fig. 12. Visual comparison of frame reconstruction between D-DPCC and our method, based on the 30th frame of the Loot point cloud sequence, which contains 779986 points. The compression rate is set to 0.25 bpp. D1-PSNR of D-DPCC and our method are 75.6 and 76, respectively.

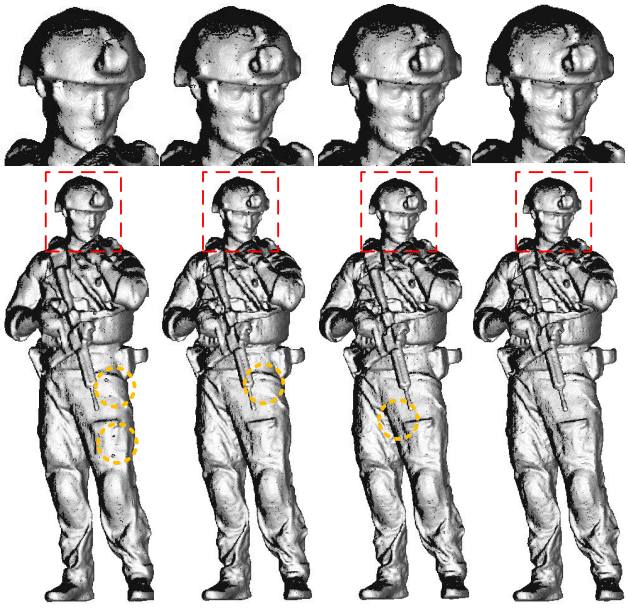


Fig. 13. Visual comparison under different bit rates per pixel (bpp) based on the 10th frame of the soldier point cloud sequence. The first row shows the enlarged views of the image contents highlighted by the corresponding red dotted boxes. Obvious protrusions or missing details are highlighted by the yellow boxes.

5. Conclusion

We present a novel deep end-to-end compression framework for DPCs that optimizes motion estimation, motion compensation, motion compression, and residual compression modules, all in latent space. We utilize sparse convolution for DPCC and a multi-scale framework to estimate motion in the latent space, which enhances the accuracy of the scene flow. Inspired by video compression techniques, we introduce time information to the motion compensation module of the current frame to improve the prediction accuracy and reduce residual compression. Additionally, for residual compression, we adapt the deep entropy model from video-based compression and introduce it to DPCC for the first time, which provides a more precise modeling of the probability distribution of latent variables. We conduct an end-to-end joint training of the entire framework to optimize the rate-distortion ratio. Compared to state-of-the-art D-DPCC, our proposed compression network achieves a BD-Rate gain of 12.09 %.

Acknowledgments

This research was partially supported by Zhejiang Province Natural Science Foundation No. LY21F020013, LY22F020013, the National Natural Science Foundation of China No. 62172366. Gary Tam is supported by the Royal Society grant IEC/NSFC/211159. For the purpose of Open Access the author has applied a CC BY copyright licence to any Author Accepted Manuscript version arising from this submission.

References

[1] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P.A. Chou, R.A. Cohen, M. Krivokuća, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan, A. Tabatabai, A.M. Tourapis, V. Zakharchenko, Emerging mpeg standards for point cloud compression, IEEE Journal on Emerging and Selected Topics in Circuits and Systems 9 (2019) 133–148.

[2] L. Huang, S. Wang, K. Wong, J. Liu, R. Urtasun, Octsqueeze: Octree-structured entropy model for lidar compression, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 1310–1320.

[3] S. Biswas, J. Liu, K. Wong, S. Wang, R. Urtasun, Muscle: Multi sweep compression of lidar using deep entropy models, Image and Video Processing(eess.IV) (2020).

[4] O.N. D Graziosi, S. Kuma, A. Zaghetto, T. Suzuki, A. Tabatabai., 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 9 (2020).

[5] J. Kammerl, N. Blodow, R.B. Rusu, S. Gedikli, M. Beetz, E. Steinbach, Real-time compression of point cloud streams, in: 2012 IEEE International Conference on Robotics and Automation, 2012, pp. 778–785. doi:10.1109/ICRA.2012.6224647.

[6] D. Thanou, P.A. Chou, P. Frossard, Graph-based compression of dynamic 3d point cloud sequences, IEEE Transactions on Image Processing 25 (2016) 1765–1778.

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

[8] Z. Hu, G. Lu, D. Xu, Fvc: A new framework towards deep video compression in feature space, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 1502–1511. doi:10.1109/CVPR46437.2021.00155.

[9] J. Wang, D. Ding, Z. Li, Z. Ma, Multiscale point cloud geometry compression, in: 2021 Data Compression Conference (DCC), 2021, pp. 73–82. doi:10.1109/DCC50243.2021.00015.

[10] A. Akhtar, Z. Li, G.V. der Auwera, Inter-frame compression for dynamic point cloud geometry coding, 2022. arXiv:2207.12554.

[11] T. Fan, L. Gao, Y. Xu, Z. Li, D. Wang, D-DPCC: Deep dynamic point cloud compression via 3d motion prediction, in: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI, 2022, pp. 898–904. doi:10.24963/ijcai.2022/126.

[12] E. d'Eon, B. Harrison, T. Myers, P.A. Chou., 8i voxelized full bodies - a voxelized point cloud dataset (2017).

[13] R. Schnabel, R. Klein, Octree-based point-cloud compression, in: Proceedings of the 3rd Eurographics / IEEE VGTC Conference on Point-Based Graphics, SPBG'06, Eurographics Association, 2006, p. 111–121.

[14] F. Song, Y. Shao, W. Gao, H. Wang, T. Li, Layer-wise geometry aggregation framework for lossless lidar point cloud compression, IEEE Transactions on Circuits and Systems for Video Technology 31 (2021) 4603–4616.

[15] C. Fu, G. Li, R. Song, W. Gao, S. Liu, Octattention: Octree-based large-scale contexts model for point cloud compression, Proceedings of the (AAAI) Conference on Artificial Intelligence 36 (2022) 625–633.

[16] D.T. Nguyen, M. Quach, G. Valenzise, P. Duhamel, Learning-based lossless compression of 3d point cloud geometry, in: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 4220–4224. doi:10.1109/ICASSP39728.2021.9414763.

[17] D.T. Nguyen, M. Quach, G. Valenzise, P. Duhamel, Multiscale deep context modeling for lossless point cloud geometry compression, in: 2021 IEEE International Conference on Multimedia Expo Workshops (ICMEW), 2021, pp. 1–6. doi:10.1109/ICMEW53276.2021.9455990.

[18] C. Choy, J. Gwak, S. Savarese, 4d spatio-temporal convnets: Minkowski convolutional neural networks, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3070–3079. doi:10.1109/CVPR.2019.00319.

[19] E. Peixoto, E. Medeiros, E. Ramalho, Silhouette 4d: An inter-frame lossless geometry coder of dynamic voxelized point clouds, in: 2020 IEEE International Conference on Image Processing (ICIP), 2020, pp. 2691–2695. doi:10.1109/ICIP40778.2020.9190648.

[20] L. Li, Z. Li, V. Zakharchenko, J. Chen, H. Li, Advanced 3d motion prediction for video-based dynamic point cloud compression, IEEE Transactions on Image Processing 29 (2020) 289–302.

[21] W. Jia, L. Li, A. Akhtar, Z. Li, S. Liu, Convolutional neural network-based occupancy map accuracy improvement for video-based point cloud compression, IEEE Transactions on Multimedia 24 (2022) 2352–2365.

[22] X. Liu, C.R. Qi, L.J. Guibas, Flownet3d: Learning scene flow in 3d point clouds, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 529–537. doi:10.1109/CVPR.2019.00062.

[23] X. Gu, Y. Wang, C. Wu, Y.J. Lee, P. Wang, Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3249–3258. doi:10.1109/CVPR.2019.00337.

[24] W. Wu, Z.Y. Wang, Z. Li, W. Liu, L. Fuxin, Pointpwc-net: Cost volume on point clouds for (self-)supervised scene flow estimation, in: A. Vedaldi, H. Bischof, T. Brox, J.M. Frahm (Eds.), Computer Vision – ECCV 2020, Springer International Publishing, 2020, pp. 88–107.

[25] C.R. Qi, L. Yi, H. Su, L.J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, CoRR abs/1706.02413 (2017).

[26] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, Z. Gao, Dvc: An end-to-end deep video compression framework, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[27] J. Lin, D. Liu, H. Li, F. Wu, M-LVC: Multiple frames prediction for learned video compression, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020. doi:10.1109/cvpr42600.2020.00360.

[28] Z. Sun, Z. Tan, X. Sun, F. Zhang, D. Li, Y. Qian, H. Li, Spatiotemporal entropy model is all you need for learned video compression, 2021. arXiv:2104.06083.

[29] M. Bosi, K. Brandenburg, S.R. Quackenbush, L.D. Fielder, K. Akagiri, H. Fuchs, M. Dietz, Iso/iec mpeg-2 advanced audio coding, Journal of The Audio Engineering Society 45 (1997) 789–814.

[30] J. Ballé, D. Minnen, S. Singh, S.J. Hwang, N. Johnston, Variational image compression with a scale hyperprior, in: arXiv:1802.01436 [eess.IV], 2018. doi:10.48550/arXiv.1802.01436.

[31] J. Ballé, V. Laparra, E.P. Simoncelli, End-to-end optimized image compression, 2017. arXiv:1611.01704.

[32] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017. arXiv:1412.6980.

[33] Y. Xu, Y. Lu, Z. Wen, Owlii dynamic human mesh sequence dataset (2017).

[34] D. Tian, H. Ochimizu, C. Feng, R. Cohen, A. Vetro, Geometric distortion metrics for point cloud compression, in: 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 3460–3464. doi:10.1109/ICIP.2017.8296925.