Scalable Point Cloud Coding for Reconstruction and Classification

Quoc Anh Le, Vu Ha Le, Mohamed-Chaker Larabi, Giuseppe Valenzise

Scalable Point Cloud Coding for Reconstruction and Classification

Le Quoc Anh
lqanh@vnu.edu.vn
AVITECH, VNU University
of Engineering and Technology
Hanoi, Vietnam

Le Vu Ha
halv@vnu.edu.vn
FET, VNU University
of Engineering and Technology
Hanoi, Vietnam

Mohamed-Chaker Larabi
chaker.larabi@univ-poitiers.fr
XLIM, UMR 7252, CNRS,
Université de Poitiers
Poitiers, France

Giuseppe Valenzise
giuseppe.valenzise@l2s.centralesupelec.fr
L2S, CNRS, CentraleSupélec,
Université Paris-Saclay
Gif-sur-Yvette, France

ABSTRACT

Point clouds are widely used in various applications for human visualization and machine vision tasks. However, most point cloud coding methods are optimized for human visualization, resulting in degraded performance or suboptimal transmitted information for machine vision tasks. In this paper, we propose a point cloud geometry coding framework that supports reconstruction and classification tasks. Our framework is based on the PCGCv2 architecture to generate a latent space representation of the point cloud. Subsequently, a residual method is applied to generate representations for two tasks from the latent space. We evaluate our framework on the ModelNet10 dataset and show that it achieves a 59.2% reduction in BD bitrate for classification tasks compared to a non-specialized coding framework while maintaining comparable performance with the state-of-the-art point cloud codec for reconstruction. This study is the preliminary study to design a point cloud coding framework that is geared toward point cloud coding for humans and machines.

KEYWORDS

point cloud geometry, scalable coding, sparse tensor, point cloud reconstruction, point cloud classification

1 INTRODUCTION

Point cloud is a popular visual content format used in various applications such as AR/VR, autonomous driving, and construction [1]. Their ability to represent spatial information makes them invaluable for tasks ranging from immersive experiences to complex infrastructure development. However, the large volume of data in point clouds poses challenges for storage, transmission, and processing. Therefore, effective point cloud coding/compression methods are essential to use point clouds in practical applications [2].

Visual content compression methods have primarily focused on the reconstruction quality for human visualization. This results in decreased machine vision performance or suboptimal information transmission [3]. Therefore, research directions on coding for humans and machines have attracted recent attention in the research community. Acknowledged the potential of deep learning-based coding, several pioneering studies proposed for image coding have shown high effectiveness for machine vision performance while still having comparable reconstruction performance with the existing coding framework [4, 5]. The main idea of these methods is to explore the coding framework with additional loss functions for machine vision tasks, making the framework generate the representation for machine vision tasks. The scalable approach is widely used in several image coding studies to generate representations for multiple tasks [5]. The residual method is a simple scheme but an effective way for scalable coding [6]. To our knowledge, there is a lack of studies based on this approach for point cloud coding.

We proposed a point cloud coding framework for reconstruction and classification tasks. The proposed method uses a state-of-the-art point cloud geometry compression framework as the baseline model, named PCGCv2 [7]. We recognize the potential to leverage information from the latent space of PCGCv2 for point cloud classification. The latent space has information to reconstruct the point clouds, a data-intensive task, making it contain sufficient information for classification tasks. By applying the residual method to the latent space of PCGCv2, our coding framework consists of two branches that are able to generate two representations for point cloud classification and reconstruction.

2 RELATED WORKS

Recently, several studies have been proposed for point cloud coding focused on optimizing machine vision performance. Liu et al. (2023) introduced a point cloud coding framework for human and machine vision [3]. The main idea of the method is to leverage a well-known point cloud codec called VoxelContext-Net to compress point cloud data with different densities for each task. In the application for point cloud classification, the method utilizes a point cloud selection module to choose key points based on the farthest point sample algorithm. The generated sparse point cloud data is fed to a specialized classifier for sparse point clouds. On the other hand, for human vision, the framework uses all point cloud information as the input of VoxelContext-Net to achieve the highest reconstruction quality. However, with this approach, there will be a significant amount of redundant information when using both tasks simultaneously. Ma et al. (2023) proposed a balanced human-machine scheme in point cloud geometry coding [8]. The method mainly uses a learned semantic mining module to aggregate multitask features. This allows the method to retain the geometry

and semantic properties in point cloud coding. Although there is an improvement in compression efficiency, there is a significant amount of redundant bits for machine vision tasks, as only a single bitstream is used for both human and machine vision tasks. Following the concept of coding for machines, Ulhaq and Bajic (2023) proposed a point cloud compression codec designed for a single machine vision task [9]. The main idea is based on the Information Bottleneck concept to derive the loss function and network architecture. However, this work has a limitation because it explores compression for classification only.

3 PROPOSED METHOD

The diagram in Figure 1 illustrates the scheme of the proposed method. In this scheme, leveraging the efficiency of PCGCv2 for the point cloud geometry compression task, we use the encoder of PCGCv2 to generate the latent space representation. Subsequently, we applied the residual method to generate representations for point cloud classification (base branch) and point cloud geometry compression (both base branch and enhancement branch). We will provide a detailed description of the proposed method in the next paragraphs.

Base brach: The classification backbone is designed based on the PointNet architecture using sparse convolution, and the last layer of the classification backbone is a global max pooling block to generate the feature vector. Subsequently, the feature vector is element-wise multiplication with a trainable gain vector $v \in \mathbb{R}^{1024\times 1}$, similar to the approach employed in the work by Ulhaq and Bajic [9]. The feature vector is multiplied by a constant scalar value of 10 for enhanced stability and convergence. On the decoder side, a set of pointwise convolutional layers with kernel size 1 is used to create a block equivalent to "share MLP," which is used in the PoitnNet [10].

Synthesis transform ($h_r(.)$): Because the information transmitted by the base branch is a feature vector $\hat{Z_b}$ with a size of 1024×1. On the other hand, we utilize the residual method for the scalable scheme. Therefore, we need to adjust the size of the feature vector $\hat{Z_b}$ to match the size of the features from the latent space Y for the subtraction/addition operation. We will employ the repeated technique to restore the size of the feature vector as the size of the feature after the max pooling layer. Then, we use a set of sparse convolution layers similar to the classification backbone to generate the latent space representation of the base branch (Y_p). Figure 2 displays the process of adjusting the size of feature vector $\hat{Z_b}$ for the residual method.

Enhancement branch: For a latent space Y, a compression method is applied to create the base representation $Z_b = f_b(Y)$ to minimize distortion $D_b = \mathbb{E}_Y[d_b(g_b(Z_b), T)]$ for a specific machine vision task (T), where $d_b(\cdot,\cdot)$ is a function to measure the distortion of the machine vision task, and $g_b(.)$ is the learnable decoding function for the machine vision task. In the residual scheme, a compression method is applied to create the enhancement representation $Z_r = f_e(Y_r); Y_r = Y - Y_b; Y_b = h_r(Z_b)$ to minimize distortion $D_r = \mathbb{E}_Y[d_e(g_e(Z_r) + Y_b, Y)]$, where $d_e(\cdot,\cdot)$ is a function to measure the distortion of the human visualization and $g_e(.)$ is learnable decoder for latent space reconstruction. Based on the work of Andrade et al. (2023) [6], we use the residual bottleneck block (RBB) as the basic unit of the enhancement branch. Figure 3 illustrates the detailed architecture of RBB.

4 EXPERIMENTS

This research was conducted on a workstation running the Ubuntu 20.04 operating system, equipped with an Intel Core i9-10900K CPU, 64GB of RAM, and an RTX 3090 GPU with 24GB of VRAM. The point cloud compression frameworks were implemented in a Python 3.8 environment, utilizing the PyTorch 1.8.1 deep learning framework with CUDA 11.1.

We use the ModelNet10 dataset for training and evaluating, which comprises 4,899 object models, divided into a training set with 3,991 objects and a testing set with 908 objects [11]. The raw data format of the ModelNet10 dataset is in the form of mesh objects. We applied the preprocessing step for the raw mesh data to generate the point cloud data with a number of sampled points of 5×10^5 and used a resolution of 128 for the voxelization process. For the human visualization benchmark, we randomly select 50 mesh objects (5 object models for each class) for evaluation.

In the training phase, we train PCGCv2 with the ModelNet10 dataset and keep the training parameters the same as those provided in the PCGCv2 paper [7]. We choose the highest rate model of PCGCv2 for the next phase. Then, we freeze the encoder, entropy bottleneck, and decoder of PCGCv2. Next, we train the base branch for the point cloud classification task for the accuracy-rate under different values of λ_b, where λ_b is the hyperparameter used to control the rate in the base branch. We vary the λ_b value between 160 and 20000. We add a minor reconstruction penalty on a transformed Z_b in the final loss function for training the base branch with the hyperparameter $\beta = 0.1$. The loss function for training the base branch is defined as follows:

$$\mathcal{L}_b = D_b + \lambda_b H\left(\hat{Z}_b\right) + \beta\mathbb{E}\left[d_e\left(h_r\left(\hat{Z}_b\right), Y\right)\right], \tag{1}$$

where D_b is the cross-entropy loss for the classification task, the function $H(\cdot)$ is used for entropy calculation, and $d_e(\cdot)$ is mean square error (MSE) loss.

Based on the results of the base branch, we will select a model with the best accuracy-rate performance to continue using for training the enhancement branch. To train the enhancement branch, we freeze the layers used in the base branch. We use λ_r as the hyperparameter to control the rate in the enhancement branch. We vary the λ_r value between 0 and 0.25. The loss function for training the enhancement branch is defined as follows:

$$\mathcal{L}_e = D_r + \lambda_r H\left(\hat{Y_r}\right), \tag{2}$$

where D_b is the mean square error (MSE) loss.

During the training process, we use Adam optimizer with an initial learning rate set to 0.005 and halve it every epoch until it reaches $1e-5$. We use 50 epochs for training, and if the validation loss does not decrease for 5 consecutive epochs, the training process is stopped.

For comparison of classification performance, we implement several methods for point cloud classification, including PointNet, PointNet++, and MinkPointNet. For PointNet and PointNet++. For point cloud compression for classification, we implemented the study by Ulhaq and Bajic (2023) using the framework they provided

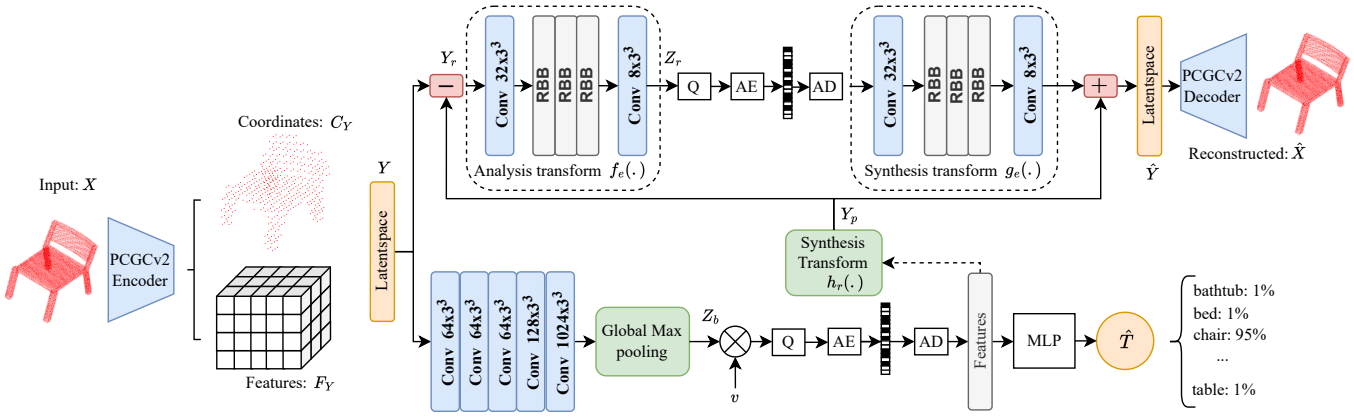

Figure 1: The overall architecture of the proposed method in scalable point cloud coding for reconstruction and classification. "Conv $c \times n^3$" represents the sparse convolution layer, where c is the output channels and n^3 is the kernel size. "Q", "AE", and "AD" are quantization, arithmetic encoder, and arithmetic decoder, respectively. In the enhancement branch, the analysis transform ($f_e(.)$) and synthesis transform ($g_e(.)$) have the same architecture and use the Residual Bottleneck Block (RBB) as the basic unit. The dashed line arrow indicates that the enhancement branch does not influence the base branch.

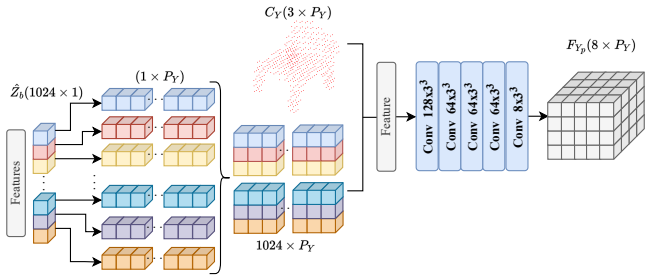

Figure 2: The process of adjusting the size of feature vector $\hat{Z}_b$. P_Y is the number of points of C_Y.

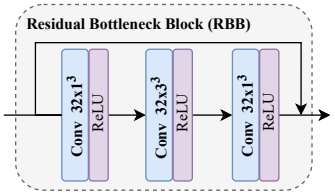

Figure 3: The architecture of the Residual Bottleneck Block.

in their GitHub repository [9]. Additionally, we implement a non-specialized codec for point cloud classification using G-PCC as the codec and PointNet++ as the classifier. To evaluate the performance of human visualization, we implement two methods, G-PCC [2] and GEO-CNNv2 [12].

5 RESULTS

Figure 4 shows the rate-accuracy curves of the proposed codec compared to other input compression codecs and results of baseline models, including PointNet, PointNet++, and MinkPointNet. The results of the method proposed by Ulhaq and Bajic (2023) [9] show

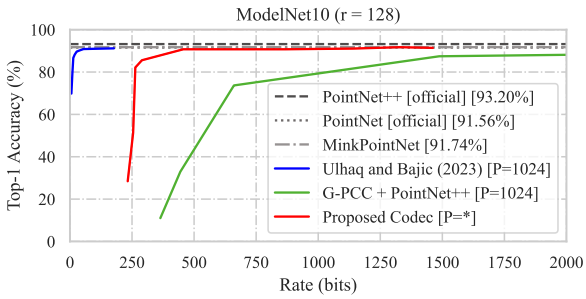

Figure 4: Rate-accuracy curves evaluated on the ModelNet10 dataset. P is the number of points in the input X. "*" denotes a dense point cloud.

the best optimization in terms of accuracy-rate for point cloud classification. We use Bjøntegaard-Delta (BD) to calculate the improvement in rate and accuracy compared with the nonspecialized codec for point cloud classification (G-PCC + PointNet++). The results show that our proposed codec achieves a 59.2% reduction, while the method proposed by Ulhaq and Bajic (2023) achieves a 98.9% reduction in BD bitrate. Our proposed codec has the optimal model at 450 bits with 90.7% accuracy, which will be the selected model for the next phase, the human visualization benchmark.

Table 1: BD-rate gains on ModelNet10 of the proposed codec against other compression methods using D1 and D2 based BD-rate measurement.

Metrics	G-PCC	GEO-CNNv2	PCGCv2
D1/D2	-6.4/38.2	-15.2/-21.9	40.5/42.1

Table 1 and Figure 5 show that the reconstruction performance of the proposed method is degraded compared with PCGCv2. PCGCv2

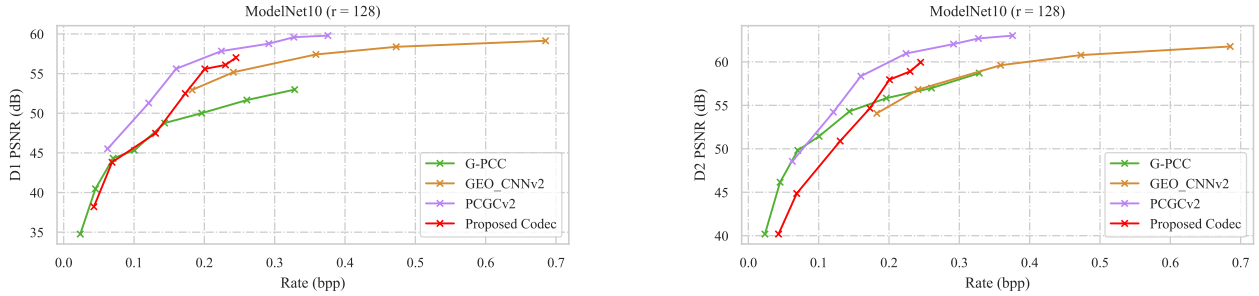

Figure 5: Rate-distortion curves of ModelNet10 dataset: (left) D1-based PSNR, (right) D2-based PSNR

Table 2: Average coding time of different methods on the ModelNet10 dataset.

Time	G-PCC	GEO-CNNv2	PCGCv2	Proposed codec
Enc (s)	0.27	0.24	0.09	0.14
Dec (s)	0.36	14.92	0.14	0.16

has averages larger than 40% BD-rate gains against our proposed method. Compared to other compression methods, our proposed codec achieved comparable performance with G-PCC and GEO-CNNv2. Table 2 indicates that integrating an additional classification task does not significantly increase the coding time of the proposed codec.

6 CONCLUSION

In this study, we present a framework that utilizes the residual method in scalable point cloud coding for reconstruction and classification. We provide detailed information about the method and conduct objective performance evaluations on the ModelNet10 dataset. The proposed method demonstrates effectiveness in the classification task by achieving high gains in rate-accuracy performance compared to a nonspecialized codec. For the reconstruction task, the proposed codec achieved performance comparable to that of the existing methods. We hope our work will help future research on designing a point cloud coding frame supporting multiple tasks.

REFERENCES

[1] Jianqiang Wang et al. "Sparse tensor-based multiscale representation for point cloud geometry compression". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[2] Sebastian Schwarz et al. "Emerging MPEG standards for point cloud compression". In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9.1 (2018), pp. 133–148.

[3] Lei Liu, Zhihao Hu, and Jing Zhang. "PCHM-Net: A New Point Cloud Compression Framework for Both Human Vision and Machine Vision". In: *2023 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2023, pp. 1997–2002.

[4] Shuai Yang et al. "Towards coding for human and machine vision: Scalable face image coding". In: *IEEE Transactions on Multimedia* 23 (2021), pp. 2957–2971.

[5] Hyomin Choi and Ivan V Bajić. "Scalable image coding for humans and machines". In: *IEEE Transactions on Image Processing* 31 (2022), pp. 2739–2754.

[6] Anderson de Andrade et al. "Conditional and Residual Methods in Scalable Coding for Humans and Machines". In: *arXiv preprint arXiv:2305.02562* (2023).

[7] Jianqiang Wang et al. "Multiscale point cloud geometry compression". In: *2021 Data Compression Conference (DCC)*. IEEE. 2021, pp. 73–82.

[8] Xiaoqi Ma et al. "HM-PCGC: A Human-Machine Balanced Point Cloud Geometry Compression Scheme". In: *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2023, pp. 2265–2269.

[9] Mateen Ulhaq and Ivan V Bajić. "Learned Point Cloud Compression for Classification". In: *2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE. 2023, pp. 1–6.

[10] Charles R Qi et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 652–660.

[11] Zhirong Wu et al. "3d shapenets: A deep representation for volumetric shapes". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1912–1920.

[12] Maurice Quach, Giuseppe Valenzise, and Frederic Dufaux. "Improved deep point cloud geometry compression". In: *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE. 2020, pp. 1–6.