



Special Paper

방송공학회논문지 제30권 제7호, 2025년 12월 (JBE Vol. 30, No. 7, December 2025)

<https://doi.org/10.5909/JBE.2025.30.7.1143>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

# Enhancing Lossless Compression in AI-PCC via Distribution-Aware Feature Extraction

Dowan Kwon<sup>a)</sup>, Joon Kwon<sup>a)</sup>, Junsik Kim<sup>b)</sup>, and Kyuheon Kim<sup>a)‡</sup>

## Abstract

The Moving Picture Experts Group (MPEG) is currently standardizing AI-based Point Cloud Compression (AI-PCC). Since AI-PCC reconstructs point clouds using geometric features extracted from the input, the quality of reconstruction improves as these extracted features become more precise. However, although point clouds exhibit variations in their geometric distributions, the current AI-PCC framework extracts features using identical cubic receptive fields. To address this limitation, this paper proposes a distribution-aware feature extraction method that uses an orthotropic rect-kernel extractor/aggregator for dense point clouds to capture linear and planar characteristics, and a sparse dilated convolution extractor for sparse point clouds to reflect point dispersion. The proposed method enhances lossless compression performance in AI-PCC by performing adaptive feature extraction to the structural heterogeneity of point clouds.

Keyword : 3D Point Cloud, Geometry Coding, AI-PCC, Feature Extraction, Lossless Coding

a) Department of Electronics and Information Convergence Engineering,  
Kyung Hee University

b) Hyper-Reality Metaverse Research Laboratory, Electronics and  
Telecommunications Research Institute

‡ Corresponding Author : Kyuheon Kim

E-mail: [kyuheonkim@khu.ac.kr](mailto:kyuheonkim@khu.ac.kr)

Tel: +82-31-201-2442

ORCID: <https://orcid.org/0000-0003-1553-936X>

※ This work was supported by an internal grant from the Electronics and Telecommunications Research Institute (ETRI) [24BC2100, Development of High-Quality, Ultra-Fast Rendering and Compression Technologies Based on Neural Radiance Field Representation] and the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center (ITRC) support program (IITP-2025-RS-2021-II212046) supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP).

· Manuscript October 16, 2025; Revised November 12, 2025; Accepted November 17, 2025.

## 1. Introduction

Point clouds represent real-world objects or scenes as a set of points in a three-dimensional coordinate system, making them an emerging next-generation 3D content format capable of capturing highly precise spatial information<sup>[1][2][3][4]</sup>. Each point contains geometric information such as its position  $(x, y, z)$  or  $(r, \theta, \phi)$  along with attribute information such as color or reflectance. However, compared to the conventional 2D image content, point clouds require tens of times more data, making compression technologies essential for efficient storage and

Copyright © 2025 Korean Institute of Broadcast and Media Engineers. All rights reserved.

“This is an Open-Access article distributed under the terms of the Creative Commons BY-NC-ND (<http://creativecommons.org/licenses/by-nc-nd/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited and not altered.”

transmission.

Accordingly, the Moving Picture Experts Group (MPEG) Working Group 7 (3D Graphics and Haptics, 3DGH) has been standardizing point cloud compression technologies since 2017. Representative standards include Video-based Point Cloud Compression (V-PCC)<sup>[5][6]</sup> and Geometry-based Point Cloud Compression (G-PCC)<sup>[7]</sup>. These standards are primarily signal processing-based approaches and achieve a balance between compression efficiency and reconstruction quality by exploiting the structural and spatial correlations inherent in point clouds.

Recently, with the advancement of artificial intelligence-based compression techniques, extensive research has been made to overcome the limitations of traditional signal processing-based compression method<sup>[8][9][10][11][12]</sup>. Accordingly, at the 148th MPEG meeting held in November 2024, the standardization of AI-based Point Cloud Compression (AI-PCC) was officially launched under the AI-based 3D Graphics Coding (AI-GC) group<sup>[13][14][15]</sup>.

AI-PCC performs decoding process based on geometric features extracted from the input point cloud, and thus, the precision and expressiveness of the extracted features play a crucial role in determining the overall coding performance. However, the current AI-PCC feature extraction framework still applies convolution operations using uniform cubic receptive fields<sup>[16]</sup>, without considering the variations in geometric distributions not only across different point cloud contents but also within a single point cloud. This unified design limits the network's ability to effectively capture the directional and anisotropic characteristics of 3D structures. To address this limitation, this paper proposes a distribution-based feature extraction method that consists of an orthotropic rect-kernel extractor/aggregator for dense point clouds to capture linear and planar features, and a sparse dilated extractor for sparse point clouds to represent dispersed structures.

## II. Background

Before describing the proposed technique, this section introduces the overall geometry compression methods of the current standard AI-PCC<sup>[17]</sup>, TMAP (Test Model for AI-PCC)<sup>[18]</sup>, to provide a better understanding of the proposed approach.

### 1. TMAP

TMAP fundamentally utilizes a hierarchical coding structure, starting from the bit-depth of the input point cloud and progressively performing down-sampling, followed by encoding at each down-sampled layer. The lossless compression method applied to each layer is referred to as Octree-based Coding, whereas the lossy compression method is known as Feature-based Coding, which is beyond the scope of this paper and is described in detail in [18].

When Octree-based Coding is applied to all layers, the entire input point cloud is encoded in a lossless manner. In contrast, lossy compression is achieved by applying Feature-based Coding from the input bit-depth down to a specific bit-depth, while Octree-based Coding is used for the remaining layers.

### 2. Overview of Octree-based Coding

The Octree-based Coding method extracts features according to the bit-depth of the input point cloud, estimates the probability of occupancy information based on these extracted features, and then performs entropy coding within a hierarchical structure.

Figure 1 illustrates the Octree-based Coding structure of TMAP at an arbitrary bit-depth. The Down-sample, Up-sample, Feature Encoder, and Octree Encoder stages are performed in the encoder, while the Feature Decoder,

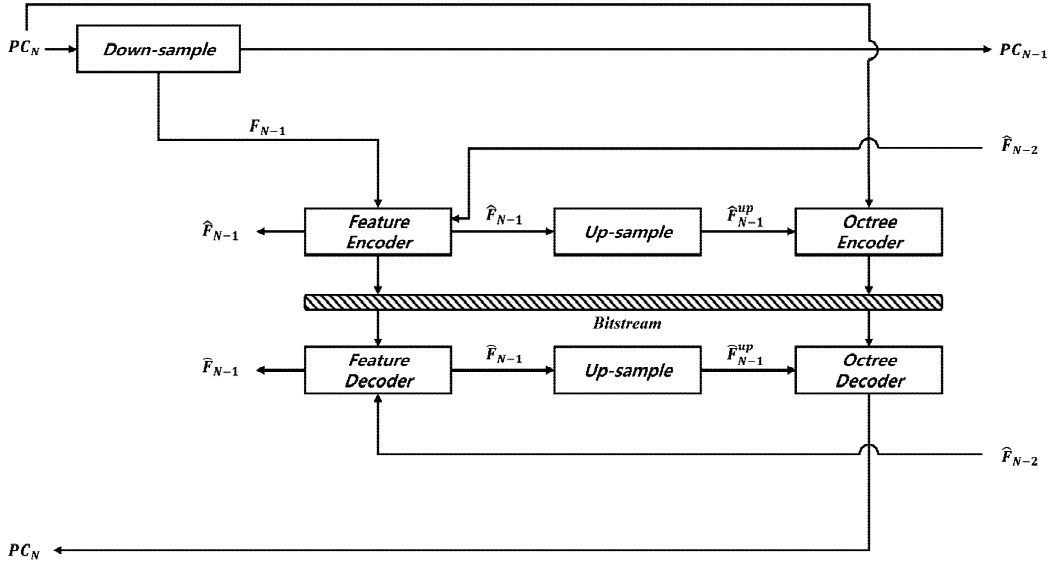


Fig. 1. Structure of TMAP Octree-based coding

Octree Decoder, and Up-Sample stages are conducted in the decoder.

Starting with the encoding process, the 3D input coordinates at bit-depth  $N$  ( $PC_N$ ) pass through the Down-sample module, which spatially down-samples the point cloud to  $PC_{N-1}$  while, in tandem, extracting representative features  $F_{N-1}$  via sparse convolutions<sup>[19][20]</sup> aligned to  $PC_{N-1}$ . Next, the Feature Encoder leverages the lower-bit-depth features  $\hat{F}_{N-2}$ , aligned with  $PC_{N-2}$ , to condition a Hyperprior Synthesis Transform that estimates the distribution of the current features  $F_{N-1}$ . The estimated probability distribution is then used for entropy encoding of quantized current features  $\hat{F}_{N-1}$ . Then, the reconstructed features  $\hat{F}_{N-1}$  are up-sampled to the resolution of  $PC_N$ , yielding  $\hat{F}_{N-1}^{up}$ . The Octree Encoder uses  $\hat{F}_{N-1}^{up}$  to predict occupancy probabilities for points in  $PC_N$  and performs arithmetic encoding conditioned on these probabilities.

In the decoding process, the Feature Decoder decodes the entropy-coded features  $\hat{F}_{N-1}$  using  $\hat{F}_{N-2}$  through the same Hyperprior Synthesis Transform used in the encoder.

The reconstructed features  $\hat{F}_{N-1}$  are then input to the Up-sample process to yield  $\hat{F}_{N-1}^{up}$ . Finally,  $\hat{F}_{N-1}^{up}$  is fed into the Octree Decoder, which estimates occupancy probabilities using the same network architecture as the encoder and reconstructs  $PC_N$  via arithmetic decoding conditioned on the predicted probabilities.

### III. Proposed Method

In this section, we present the proposed feature extraction/aggregation modules. These modules replace the Down-sample and Up-sample stage in Figure 1 by performing spatial down/up-sampling and generating representative feature tailored to the distributional characteristics of dense and sparse point clouds, respectively. To facilitate understanding of the proposed network diagrams, we first define the terminology and notation used in architecture. ‘‘Sconv.’’ denotes a network block that performs sparse convolution. The tuple ‘‘(a, b, c)’’ indicates the kernel size, and the following scalar ‘‘d’’ denotes the dilation parameter. An optional

arrow “ $\uparrow$ ” or “ $\downarrow$ ” defines up-sampling or down-sampling with stride 2, respectively. “Tanh” is shorthand for the hyperbolic tangent activation function. “Concat.” refers to concatenation; in this work, independently extracted features are appended rather than summed, allowing subsequent layers to learn how to fuse them.

For dense point clouds, where local point density is relatively uniform and neighboring points often form planar or linear structures, we propose an orthotropic rect-kernel extractor and aggregator. These extractor and aggregator apply direction-wise rectangular kernels aligned with planar and linear orientations, aggregate the corresponding features, and fuse them into a unified feature representation. The detailed architecture is illustrated in Figure 2.

In the first stage, each branch applies axis-aligned, elongated kernels along the  $x$ -,  $y$ -, or  $z$ - axis to capture linear structures, as in Equation (1).

$$\begin{aligned} L_x &= \text{sconv}_{K_x^{line}}^{s=1, d=1}(X_N), \\ L_y &= \text{sconv}_{K_y^{line}}^{s=1, d=1}(X_N), \\ L_z &= \text{sconv}_{K_z^{line}}^{s=1, d=1}(X_N) \dots \end{aligned} \tag{1}$$

Here,  $X_N$  is the input sparse tensor constructed from  $PC_N$ : its coordinates follow  $PC_N$ , and its features are initialized to ones. In  $\text{sconv}_{K_{\{x,y,z\}}^{line}}^{s,d}(\cdot)$ ,  $K_{\{x,y,z\}}^{line}$  denotes axis-aligned rectangular kernels elongated along the subscripted axis, with  $K_x^{line} \in \mathbb{R}^{2 \times 1 \times 1}$ ,  $K_y^{line} \in \mathbb{R}^{1 \times 2 \times 1}$ , and  $K_z^{line} \in \mathbb{R}^{1 \times 1 \times 2}$ ;  $d$  is the dilation parameter and  $s$  is the stride. The outputs  $L_x$ ,  $L_y$ , and  $L_z$  are the line-oriented features extracted along the  $x$ -,  $y$ -, or  $z$ - axis for each  $\text{sconv}_{K_{\{x,y,z\}}^{line}}^{s,d}(\cdot)$ , respectively.

In the second stage, plane-oriented kernels aligned with the  $yz$ ,  $xz$ , and  $xy$  planes transform these into planar responses, as in Equation (2).

$$\begin{aligned} P_{yz} &= \text{sconv}_{K_{yz}^{plane}}^{s=1, d=1}(L_x), \\ P_{xz} &= \text{sconv}_{K_{xz}^{plane}}^{s=1, d=1}(L_y), \\ P_{xy} &= \text{sconv}_{K_{xy}^{plane}}^{s=1, d=1}(L_z) \dots \end{aligned} \tag{2}$$

Here,  $P_{yz}$ ,  $P_{xz}$ , and  $P_{xy}$  are plane-oriented responses for the  $yz$ -,  $xz$ -, and  $xy$ - planes provided from each

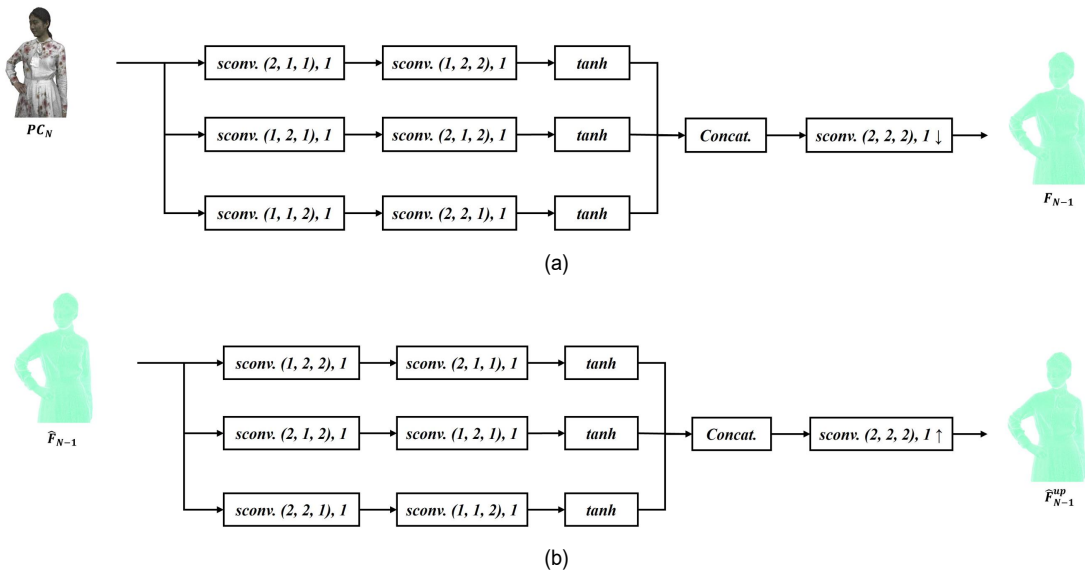


Fig. 2. Structure of orthotropic rect-kernel (a) extractor and (b) aggregator

$$Z = \text{Concat}(\tanh(P_{yz}), \tanh(P_{xz}), \tanh(P_{xy})), F_{N-1} = \text{sconv}_{K^{mix}}^{s=2, d=1}(Z) \dots \quad (3)$$

$\text{sconv}_{K_{\{yz, xz, xy\}}^{plane}}^{s, d}(\cdot)$ , and  $K_{\{yz, xz, xy\}}^{plane}$  are rectangular kernels spanning the indicated plane, with  $K_{yz}^{plane} \in \mathbb{R}^{1 \times 2 \times 2}$ ,  $K_{xz}^{plane} \in \mathbb{R}^{2 \times 1 \times 2}$ , and  $K_{xy}^{plane} \in \mathbb{R}^{2 \times 2 \times 1}$ .

Finally, the branch outputs are passed through activation function, concatenated in a channel-append manner, and fused by a  $s = 2$  down-sampling convolution to produce  $F_{N-1}$ , as in Equation (3).

Here,  $\text{Concat}(\cdot)$  denotes coordinate-wise concatenation, and  $K^{mix} \in \mathbb{R}^{2 \times 2 \times 2}$  is a mixing kernel that blends the concatenated features  $Z$  within a local  $2 \times 2 \times 2$  neighborhood, producing a down-sampled, fused representation.

The next module is the orthotropic rect-kernel aggregator as shown in Figure 2 (b), which replaces the Up-sample block in Figure 1. It takes  $\hat{F}_{N-1}$  as input and outputs  $\hat{F}_{N-1}^{up}$ . Similar to the orthotropic rect-kernel extractor, the flow is split into three branches, each with two convolutional stages. Unlike the extractor, the order is reversed: the first stage uses plane-oriented rectangular kernels aligned with the  $yz$ ,  $xz$ , and  $xy$  planes to aggregate planar features; the second stage applies axis-aligned elongated kernels along the  $x$ -,  $y$ -, and  $z$ - axes to refine linear structures. Branch outputs pass through the activation function, are appended rather than summed, and are finally up-sampled via a stride 2 convolution to produce  $\hat{F}_{N-1}^{up}$ .

Next, we introduce the proposed sparse dilated extractor for improving feature extraction on sparse point clouds, as illustrated in Figure 3. This module targets point cloud with highly non-uniform, irregular spatial distributions by employing dilated sparse convolutions whose kernels are spatially dispersed. By enlarging the receptive field without increasing the kernel mass, the extractor captures long-range, low-density structures and aggregates features over disconnected neighborhoods typical of sparse geometry. Note that we do not propose a new aggregator for sparse point cloud since Up-sample in Figure 1 is not used in the current TMAP pipeline for sparse-point inference. Instead, a dedicated MLP-based feature aggregation stage is applied after feature extraction<sup>[21]</sup>.

As shown in Figure 3, this module shares the same input-output interface as Figure 2 (a), but the extractor design differs. The sparse dilated extractor captures irregular sparsity by probing multi-radius neighborhoods via dilation. Throughout this extractor, the convolutional layers use a fixed kernel  $K^{dil} \in \mathbb{R}^{2 \times 2 \times 2}$  with dilation rates  $d_1 < d_2 < d_3$  for each branch, as in Equation (4).

$$\begin{aligned} S_1 &= \text{sconv}_{K^{dil}}^{s=1, d=d_1}(X_N), \\ S_2 &= \text{sconv}_{K^{dil}}^{s=1, d=d_2}(X_N), \\ S_3 &= \text{sconv}_{K^{dil}}^{s=1, d=d_3}(X_N) \dots \end{aligned} \quad (4)$$

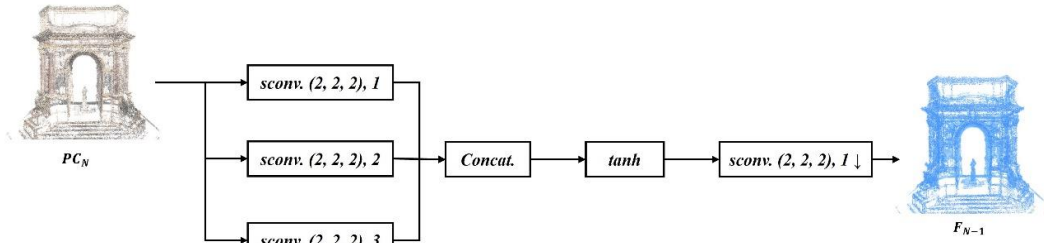


Fig. 3. Structure of sparse dilated extractor

$$Z_{sparse} = \text{Concat}(\tanh(S_1), \tanh(S_2), \tanh(S_3)), F_{N-1} = \text{sconv}_{K^{mix}}^{s=2, d=1}(Z_{sparse}) \dots \quad (5)$$

The features from each branch are concatenated (channel-append at identical coordinates) to form  $Z_{sparse}$ , passed through hyper tangent activation function, and fused by a stride 2 convolution to produce  $F_{N-1}$ , as in Equation (5).

## IV. Experimental Results

This section presents the performance evaluation and analysis of the proposed method. Before detailing the results, we briefly summarize the information used for model training and inference.

### 1. Experimental Setup

#### 1.1. Datasets

The experiments follow the Common Test Conditions (CTC)<sup>[22]</sup> defined by the MPEG AI-GC group, which specify four dataset categories: RWTT, Static, Dynamic, and LiDAR. Since we focus on static scenes, we exclude Dynamic and LiDAR categories, and use RWTT and Static categories for both training and testing. RWTT<sup>[23]</sup> consists of dense point clouds obtained by sampling textured 3D mesh models represented by vertices, edges, and textures, while Static<sup>[24]</sup> consists of sparse, voxelized point-cloud frames that represent surfaces. Finally, the training and testing datasets within each category follow the same categorization defined in the CTC.

#### 1.2. Training Environment

For RWTT content, we performed 50 epochs of base training using Adam ( $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , weight decay=0) with an initial learning rate of  $5 \times 10^{-4}$ , followed by 20 epochs of fine-tuning; a step-decay schedule halved the learning rate every 10 epochs. For Static content, we used

the same optimizer and initial learning rate for 50 epochs of training with the learning rate likewise halved every 10 epochs. These procedures follow the CTC training methodology, which is recommended but not mandated, and, for fairness, we kept the training method consistent with the CTC models to ensure comparable model construction and evaluation. We trained on a single NVIDIA RTX A6000 (48 GB GDDR6, PCIe) paired with an AMD EPYC<sup>TM</sup> 7702P (64 cores @ 2.00 GHz) using PyTorch 2.4.1 with CUDA 11.8, with sparse-convolution operations implemented via MinkowskiEngine 0.5.4.

#### 1.3. Inference Environment

For inference, we used a single NVIDIA RTX 3090 Ti (24 GB GDDR6X, PCIe) with a 13th-Gen Intel Core<sup>TM</sup> i9-13900K (24 cores @ 3.00 GHz), running PyTorch 2.0.0 with CUDA 11.8 and MinkowskiEngine 0.5.4, and otherwise matching the training setup. All codec parameters followed the CTC test conditions for lossless geometry coding, and TMAP v3.1, the latest official AI-PCC test model, served as the anchor.

#### 1.4. Performance Evaluation

Lossless compression performance is assessed using the bits-per-input-point (bpip) ratio, defined as the bitstream size per input point of the proposed method divided by that of the anchor and reported as a percentage.

### 2. Compression performance of the proposed methods

After integrating the proposed method into TMAP v3.1, the performance of the orthotropic rect-kernel extractor and aggregator on RWTT content is summarized in Table 1.

As shown in Table 1, the proposed method reduced the bitstream size for all three test sequences; RWTT 156

Table 1. Lossless compression performance of the orthotropic rect-kernel extractor and aggregator compared to TMAP v3.1

Content	Bitstream size [Bits]		Bpip ratio [%]
	Anchor	Proposed	
RWTT 059 Tomb vox10	1958312	1958112	100.0
RWTT 156 Vishnu vox10	1040696	1010840	97.1
RWTT 211 Foxstatue vox10	1415968	1382736	97.7
Average	-	-	98.3

Table 2. Time complexity of the orthotropic rect-kernel extractor and aggregator compared to TMAP v3.1

Content	Relative Time Complexity (Encoder) [%]	Relative Time Complexity (Decoder) [%]
RWTT 059 Tomb vox10	101.5	107.6
RWTT 156 Vishnu vox10	101.4	110.9
RWTT 211 Foxstatue vox10	102.0	109.0
Average	101.6	109.2

Vishnu vox10 achieved the largest reduction of 2.9% bpip ratio, while RWTT 059 Tomb vox10 showed the smallest reduction of 200 bits. Although this value is too small to be visibly reflected in the table, it corresponds to approximately 0.02% bpip ratio reduction when converted.

In terms of runtime overhead, which is presented in Table 2, we observed average increases of 1.6% on the encoder and 9.2% on the decoder. The increase in complexity is likely due to the additional GPU computations introduced by the proposed design. Unlike the previous approach, which used a single cubic kernel, the proposed method employs three separate branches, each with spe-

cially designed kernels to extract features along linear and planar directions, resulting in extra computational overhead on the GPU.

Similarly, Table 3 presents the performance of the proposed sparse dilated extractor, with TMAP v3.1 serving as the anchor, and the dilation rates in Equation (4) were set to  $d_1 = 1$ ,  $d_2 = 2$ , and  $d_3 = 3$ .

As shown in Table 3, all Static sequences exhibited bitstream reductions, with an average bpip ratio decrease of 1.9%. The largest reduction was observed for Facade 00009 vox12 at about 3.2%, while Shiva 00035 vox12 showed the smallest reduction at about 1.3%.

Table 3. Lossless compression performance of the sparse dilated extractor compared to TMAP v3.1

Content	Bitstream size [Bits]		Bpip ratio [%]
	Anchor	Proposed	
Arco Valentino Dense vox12	17193760	16948192	98.6
Facade 00009 vox12	12147912	11763864	96.8
House Without Roof 00057 vox12	26924024	26537944	98.6
Shiva 00035 vox12	11023640	10875488	98.7
Staua Klimt vox12	5576400	5460720	97.9
Average	-	-	98.1

Table 4. Time complexity of the sparse dilated extractor compared to TMAP v3.1

Content	Relative Time Complexity (Encoder) [%]	Relative Time Complexity (Decoder) [%]
Arco Valentino Dense vox12	102.4	107.8
Facade 00009 vox12	102.3	108.5
House Without Roof 00057 vox12	103.5	103.6
Shiva 00035 vox12	102.6	109.2
Staua Klimt vox12	102.0	118.5
Average	102.6	109.5

Similar to the RWTT results, the time complexity shown in Table 4 increased by approximately 2.6% for the encoder and 9.5% for the decoder on average. This increase is attributed to the same reason described in Table 2. In the proposed method, three separate branches are used, and each branch applies convolution operations with different dilation rates, which introduces additional computations.

In summary, by adapting kernel design to the distribution characteristics of dense and sparse point clouds, the proposed methods enable distribution-aware feature extraction, produce more informative features, improve the precision of occupancy probability estimation, and consequently reduce the bits per input point.

### 3. Analysis on compression performance of proposed methods

As described in Section II-1, TMAP performs hierarchical coding according to the bit-depth of input point cloud. Therefore, to conduct a more detailed analysis of the proposed method's performance, the efficiency of the Octree-based Coding was examined at each hierarchical level. The performance analysis of orthotropic rect-kernel extractor and aggregator for the RWTT content is shown in Figure 4, while that of sparse dilated extractor for Static content is presented in Figure 5. In Figures 4 and 5, the y-axis represents the Geometry Bitstream Size Ratio (%), which indicates the percentage change in bitstream size at each level compared with the anchor. A positive value denotes an improvement achieved by the proposed method, whereas a negative value indicates a performance loss. For clearer visualization of the results, a red baseline was placed at the point where the ratio equals zero. In addition, as indicated in the legends of each figure, the curves are labeled in the form of slice with number, which represents

the slice\_id used to distinguish multiple subdivided slices within the tested point cloud sequence.

As shown in Figure 4, the performance trends for the RWTT test sequences exhibit highly consistent patterns across different bit-depths. While noticeable gains are observed at very low bit-depths, the mid-level layers show irregular variations in performance depending on the content. However, as the bit-depth increases, the results tend to converge toward regions of improvement. Although several mid-level layers experience local performance losses, the overall compression efficiency is primarily influenced by the highest bit-depth layers, which contain the largest data volume. Consequently, an average compression gain can be observed in the overall performance evaluation.

Figure 5 presents the performance of the proposed method for Static contents across different bit-depths. As shown in Figure 5, irregular gains and losses are observed at the lower and middle bit-depth layers for each content, and no consistent trend can be identified. In some cases, even different slices within the same content exhibit opposite characteristics such as Slice 0 and Slice 1 in Figure 5 (a), or Slice 1 and Slice 3 in Figure 5 (c). However, across all contents and slices, the results tend to either maintain or achieve performance gains at higher bit-depths, ultimately leading to an overall compression improvement.

From the analyses of Figures 4 and 5, it can be concluded that both proposed structures achieve compression gains primarily because their inference capability is more effective at higher bit-depths, where the spatial resolution is higher. Conversely, the performance variation across lower and intermediate layers appears to be less consistent. This indicates that the proposed methods are particularly effective in estimating occupancy probabilities for geometrically complex regions at high resolutions.

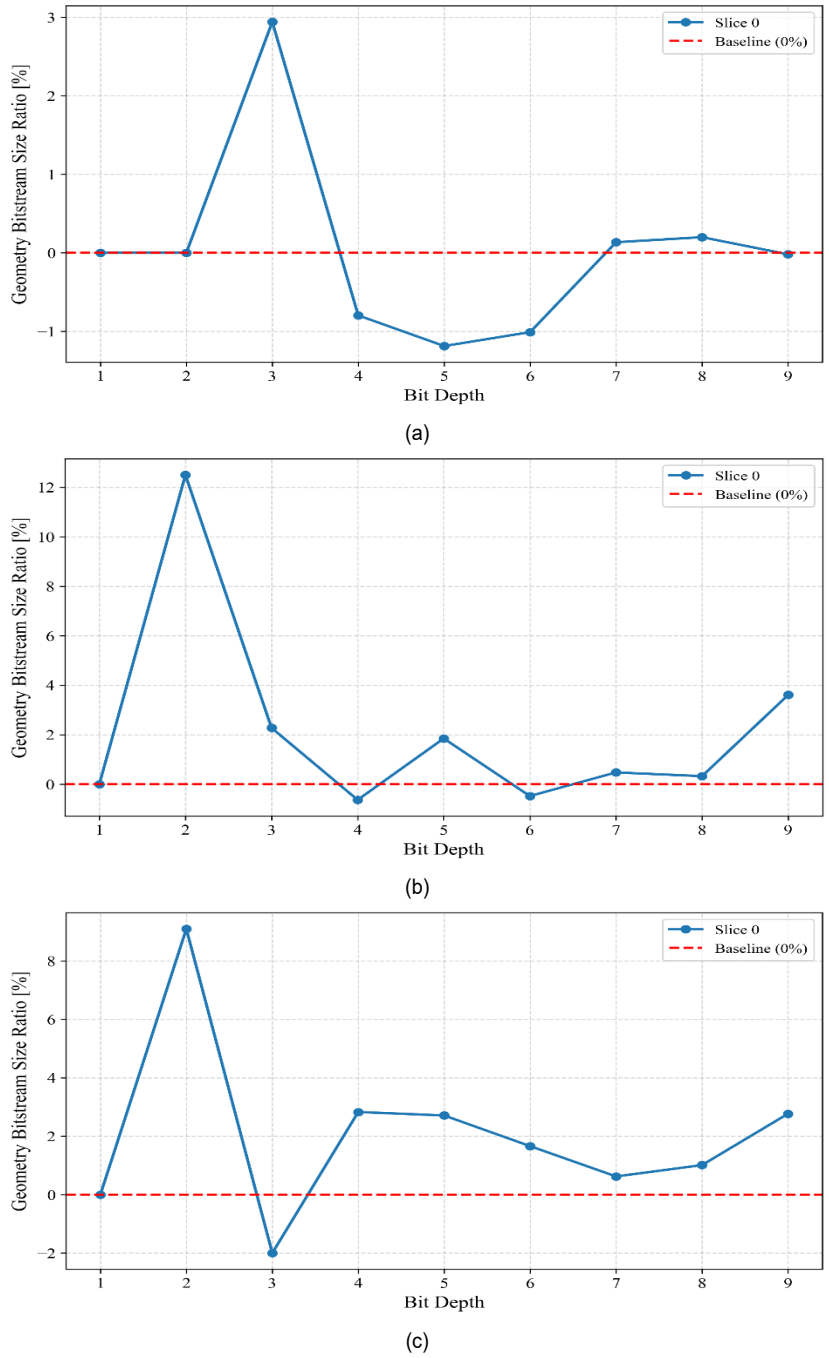


Fig. 4. Performance of the orthotropic rect-kernel extractor and aggregator across bit-depths for RWTT contents: (a) RWTT 059 Tomb vox10, (b) RWTT 156 Vishnu vox10, and (c) RWTT 211 Foxstatue vox10

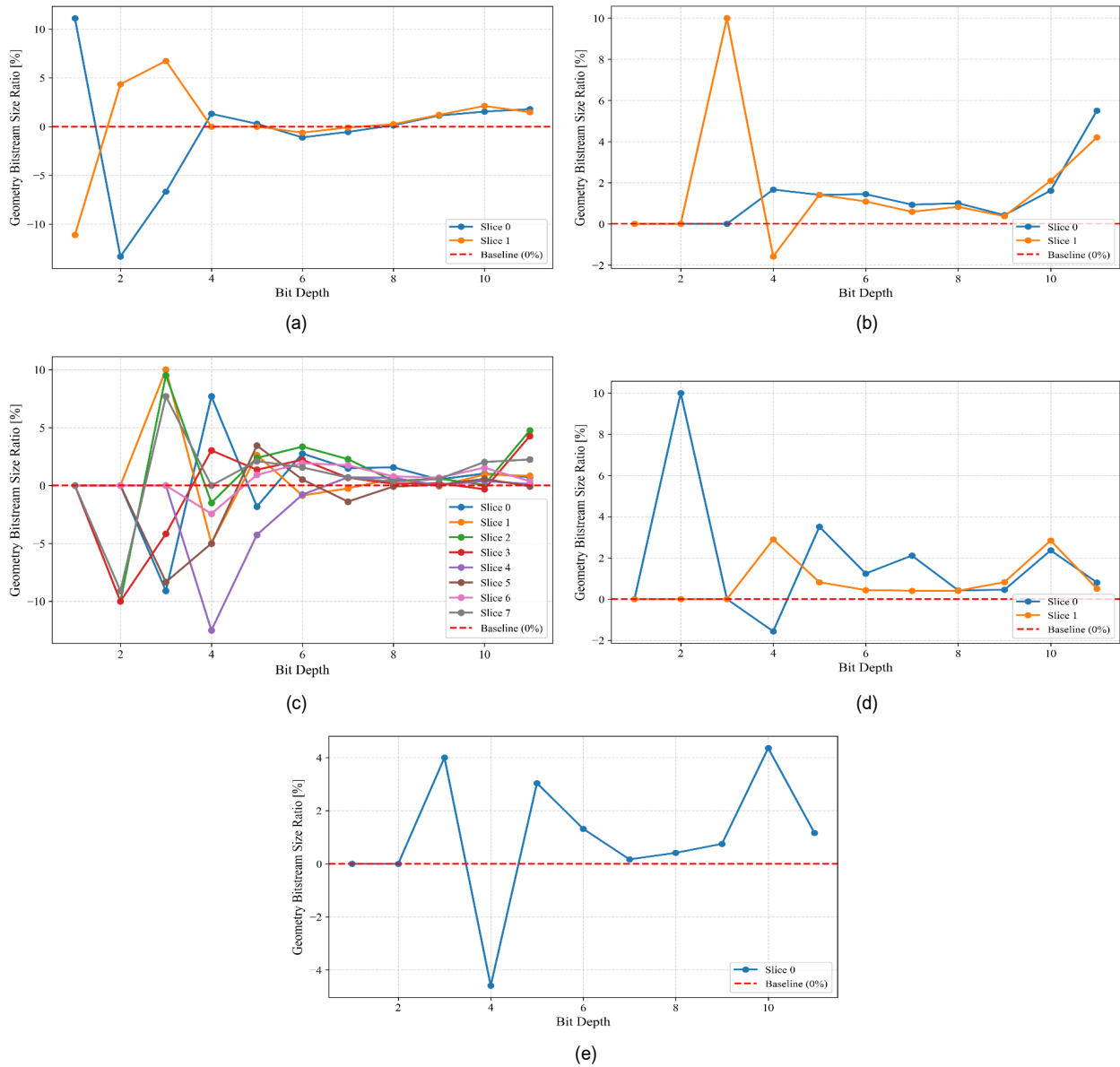


Fig. 5. Performance of the sparse dilated extractor across bit-depths for Static contents: (a) Arco Valentino Dense vox12, (b) Facade 00009 vox12, (c) House Without Roof 00057 vox12, (d) Shiva 00035 vox12, and (e) Staua Klimt vox12

## V. Conclusion

In this paper, we proposed two types of feature processing modules to enhance the feature representation and generation capability of the current AI-PCC standard codec, TMAP. For dense point cloud compression, we designed

the orthotropic rect-kernel extractor and aggregator that learn directional features along linear and planar orientations. We also introduced a sparse dilated extractor that captures irregular spatial sparsity for the sparse point cloud. The proposed methods were developed to improve occupancy probability prediction, thereby achieving gains

in lossless compression efficiency. As confirmed in Section IV, the proposed techniques do not exhibit uniform performance across all bit-depths; rather, the most significant gains are consistently observed at higher bit-depth regions, where the geometric structure becomes more complex. This implies that the proposed architectures are particularly effective at extracting and utilizing features in geometrically intricate regions with high-resolution. As a future direction, we plan to refine the framework to robustly exploit feature information even in lower bit-depth layers, thereby ensuring consistent performance across all hierarchical levels and further improving the overall compression efficiency.

## References

- [1] C.-Y. Tsai and S.-H. Tsai, "Simultaneous 3D object recognition and pose estimation based on RGB-D images," *IEEE Access*, Vol. 6, pp. 28859 - 28869, Feb. 2018.  
doi: <https://doi.org/10.1109/ACCESS.2018.2808225>
- [2] J. Zhou, H. Xu, Z. Ma, Y. Meng, and D. Hui, "Sparse point cloud generation based on turntable 2D lidar and point cloud assembly in augmented reality environment," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, Glasgow, United Kingdom, pp. 1 - 6, 2021.  
doi: <https://doi.org/10.1109/I2MTC50364.2021.9459981>
- [3] K. Lee, J. Yi, Y. Lee, S. Choi, and Y. M. Kim, "GROOT: A real-time streaming system of high-fidelity volumetric videos," in *Proc. Int. Conf. Mobile Comput. Netw.*, New York, USA, pp. 1 - 14, 2020.  
doi: <https://doi.org/10.1145/3372224.3419214>
- [4] S. Jia, X. Gong, F. Liu, and L. Ma, "AI-powered LiDAR point cloud understanding and processing: An updated survey," *IEEE Trans. Intell. Transport. Syst.*, Vol. 26, No. 8, pp. 11249-11275, Aug. 2025.  
doi: <https://doi.org/10.1109/TITS.2025.3568500>
- [5] S. Rhyu, J. Kim, J. Im, and K. Kim, "Contextual homogeneity-based patch decomposition method for higher point cloud compression," *IEEE Access*, Vol. 8, pp. 207805 - 207812, Nov. 2020.  
doi: <https://doi.org/10.1109/ACCESS.2020.3038800>
- [6] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, and P. A. Chou, et al., "Emerging MPEG standards for point cloud compression," *IEEE J. Emerg. Sel. Top. Circuits Syst.*, Vol. 9, No. 1, pp. 133 - 148, Mar. 2019.  
doi: <https://doi.org/10.1109/JETCAS.2018.2885981>
- [7] S. Rhyu, J. Kim, G. H. Park, and K. Kim, "Low complexity patch projection method for efficient and lightweight point cloud compression," *ETRI J.*, Vol. 46, No. 4, pp. 683 - 696, Aug. 2024.  
doi: <https://doi.org/10.4218/etrij.2023-0242>
- [8] J. Pang, M. A. Lodhi, and D. Tian, "GRASP-Net: Geometric residual analysis and synthesis for point cloud compression," *Proc. 1st Int. Workshop on Advances in Point Cloud Compression, Processing and Analysis (APCCPA '22)*, New York, USA, pp. 11 - 19, Oct. 2022.  
doi: <https://doi.org/10.1145/3552457.3555727>
- [9] J. Wang, D. Ding, Z. Li, X. Feng, C. Cao, and Z. Ma, "Sparse tensor-based multiscale representation for point cloud geometry compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 45, No. 7, pp. 9055 - 9071, Jul. 2023.  
doi: <https://doi.org/10.1109/TPAMI.2022.3225816>
- [10] M. Yim and J.-C. Chiang, "Mamba-PCGC: Mamba-based point cloud geometry compression," *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Abu Dhabi, United Arab Emirates, pp. 3554 - 3560, Oct. 2024.  
doi: <https://doi.org/10.1109/ICIP51287.2024.10647269>
- [11] L. Xie, W. Gao, H. Zheng, and G. Li, "SPCGC: Scalable point cloud geometry compression for machine vision," *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Yokohama, Japan, pp. 17272 - 17278, May 2024.  
doi: <https://doi.org/10.1109/ICRA57147.2024.10610894>
- [12] J. Prazeres, R. Rodrigues, M. Pereira, and A. M. G. Pinheiro, "Performance analysis of deep learning-based lossy point cloud geometry compression coding solutions," *IEEE Access*, vol. 13, pp. 76000 - 76015, Apr. 2025.  
doi: <https://doi.org/10.1109/ACCESS.2025.3561895>
- [13] M. Preda and G. Bregeon, "Consolidated report on AI-based point cloud compression results," in *ISO/IEC JTC1/SC29/WG7 MPEG input document m69550*, Kemer, Nov. 2024.
- [14] J. Wang, R. Xue, J. Li, D. Ding, Y. Lin, and Z. Ma, "A versatile point cloud compressor using universal multiscale conditional coding - Part I: Geometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 47, No. 1, pp. 269 - 287, Jan. 2025.  
doi: <https://doi.org/10.1109/TPAMI.2024.3462938>
- [15] J. Wang, R. Xue, J. Li, D. Ding, Y. Lin, and Z. Ma, "A versatile point cloud compressor using universal multiscale conditional coding - Part II: Attribute," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 47, No. 1, pp. 252 - 268, Jan. 2025.  
doi: <https://doi.org/10.1109/TPAMI.2024.3462945>
- [16] M. A. Lodhi, J. Pang, J. Ahn, and D. Tian, "[AI-GC][CfP-response] UniFHid Part 1: InterDigital's answer to MPEG AI-PCC for Track 1 Geometry-only," in *ISO/IEC JTC1/SC29/WG7 MPEG input document m70282*, Kemer, Turkey, Nov. 2024.
- [17] MPEG 3D Graphics Coding and Haptics Coding, "Working draft of AI-based point cloud coding," in *ISO/IEC JTC1/SC29/WG7 MPEG output document N01257*, July 2025.
- [18] MPEG 3D Graphics Coding and Haptics Coding, "TMAP v3 for AI-based point cloud coding," in *ISO/IEC JTC1/SC29/WG7 MPEG output document N01258*, July 2025.
- [19] B. Graham, "Sparse 3D convolutional neural networks," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Swansea, U.K., pp. 150.1 - 150.9, Sep. 2015.  
doi: <https://doi.org/10.5244/C.29.150>
- [20] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal convnets: Minkowski convolutional neural networks," in *Proc. IEEE/CVF Conf.*

Comput. Vis. Pattern Recognit. (CVPR), Long Beach, CA, USA, pp. 3075 - 3084, Jun. 2019.

doi: <https://doi.org/10.1109/CVPR.2019.00319>

- [21] B. Liu, Y. Ma, L. Li, and D. Liu, "Report on probability estimation," in ISO/IEC JTC1/SC29/WG7 MPEG input document m72144, Online, Apr. 2025.
- [22] MPEG 3D Graphics Coding and Haptics Coding, "CTC on AI-based point cloud coding," in ISO/IEC JTC1/SC29/WG7 MPEG output

document N01256, July 2025.

- [23] A. Maggiordomo, F. Ponchio, P. Cignoni, and M. Tarini, "Real-world textured things: A repository of textured models generated with modern photo-reconstruction tools," Comput. Aided Geometric Des., Vol. 83, No. 101943, Nov. 2020.  
doi: <https://doi.org/10.1016/j.cagd.2020.101943>
- [24] Culture 3D Clouds, <http://c3dc.fr> (accessed Oct. 14, 2025).

---

## Introduction Authors

---



Dowan Kwon

- 2024 : Bachelor of Science in Electronic Engineering, Kyung Hee University
- 2024 ~ Present : Master's Course of Electronics and Information Convergence Engineering, Kyung Hee University
- ORCID : <https://orcid.org/0009-0006-2150-4488>
- Research interests : Point Cloud Compression, Image Processing, Multimedia Systems



Joon Kwon

- 2024 : Bachelor of Electronic Engineering and Astronomy & Spase Science, Kyung Hee University
- 2024 ~ Present : Master's Course of Electronics and Information Convergence Engineering, Kyung Hee University
- ORCID : <https://orcid.org/0009-0003-8085-1170>
- Research interests : Point Cloud Compression, Image Processing, Multimedia Systems



Junsik Kim

- 2017 : Bachelor of Science in Electronic Engineering, Kyung Hee University
- 2019 : Master of Science in Electrical and Electronic Engineering, Kyung Hee University
- 2022 : Ph.D. in Electrical and Electronic Engineering, Kyung Hee University
- 2022 ~ 2023 : TTA Senior Engineer
- 2024 ~ Present : ETRI Research Engineer
- ORCID : <https://orcid.org/0000-0002-0287-9640>
- Research interests : Point Cloud Compression, Deep Learning, Image Processing, Multimedia Systems



Kyuheon Kim

- 1989 : Bachelor of Science in Electronic Engineering, Hanyang University
- 1996 : Ph.D. in Electrical and Electronic Engineering, University of Newcastle upon Tyne, U.K.
- 1996 ~ 1997 : Research Fellow, University of Sheffield, U.K.
- 1997 ~ 2006 : Team Leader, Interactive Media Research Team, Electronics and Telecommunications Research Institute (ETRI), Korea
- 2006 ~ Present : Professor, College of Electronics and Information, Kyung Hee University, Korea
- ORCID : <https://orcid.org/0000-0003-1553-936X>
- Research interests : Digital Broadcasting, Image Processing, Multimedia Communication, Interactive Media Systems