MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Point Cloud Geometry Compression using Parameterized Graph Fourier Transform

Kirihara, Hinata; Ibuki, Shoichi; Fujihashi, Takuya; Koike-Akino, Toshiaki; Watanabe, Takashi

TR2024-106 August 07, 2024

Abstract

Existing point cloud coding (PCC) methods based on graph signal processing (GSP) have been proposed for dealing with the irregular structures of 3D points. GSP-based PCC can achieve better rate-distortion performance against the typical tree-based PCC, whereas it requires computational resources for searching hyperparameter sets for graph shift operators and performing eigenvalue decomposition for each parameter set. This paper proposes a novel PCC to reduce the time complexity without the degradation of rate- distortion performance. Specifically, we leverage the properties of the parameterized graph shift operator to realize 1) a reduction in the number of hyperparameters, and 2) a decomposition-free hyperparameter search. Evaluations using ShapeNet point cloud dataset show that the proposed scheme achieves almost the same rate-distortion performance with significant reduction on the computational cost compared to the existing graph-based PCCs.

ACM SIGCOMM 2024

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Point Cloud Geometry Compression using Parameterized Graph Fourier Transform

Hinata Kirihara
Osaka University
Suita, Osaka, Japan
kirihara.hinata@ist.osaka-u.ac.jp

Shouichi Ibuki
Osaka University
Suita, Osaka, Japan
ibuki.shouichi@ist.osaka-u.ac.jp

Takuya Fujihashi
Osaka University
Suita, Osaka, Japan
tfuji@ist.osaka-u.ac.jp

Toshiaki Koike-Akino
Mitsubishi Electric Research Laboratories
Cambridge, MA, United States
koike@merl.com

Takashi Watanabe
Osaka University
Suita, Osaka, Japan
watanabe@ist.osaka-u.ac.jp

ABSTRACT

Existing point cloud coding (PCC) methods based on graph signal processing (GSP) have been proposed for dealing with the irregular structures of 3D points. GSP-based PCC can achieve better rate-distortion performance against the typical tree-based PCC, whereas it requires computational resources for searching hyperparameter sets for graph shift operators and performing eigenvalue decomposition for each parameter set. This paper proposes a novel PCC to reduce the time complexity without the degradation of rate-distortion performance. Specifically, we leverage the properties of the parameterized graph shift operator to realize 1) a reduction in the number of hyperparameters, and 2) a decomposition-free hyperparameter search. Evaluations using ShapeNet point cloud dataset show that the proposed scheme achieves almost the same rate-distortion performance with significant reduction on the computational cost compared to the existing graph-based PCCs.

CCS CONCEPTS

• Information systems → Multimedia streaming; **• Theory of computation → Graph algorithms analysis**; **• Networks → Cross-layer protocols**.

KEYWORDS

Point Clouds, Graph Signal Processing, Parameterized Graph Shift Operator, Decomposition-Free

ACM Reference Format:
Hinata Kirihara, Shouichi Ibuki, Takuya Fujihashi, Toshiaki Koike-Akino, and Takashi Watanabe. 2024. Point Cloud Geometry Compression using Parameterized Graph Fourier Transform. In *Emerging Multimedia Systems (EMS '24), August 4, 2024, Sydney, Australia.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3643832.3661840

1 INTRODUCTION

The miniaturization and cost reduction of cameras and 3D laser scanners, such as those used in Light Detection and Ranging (LiDAR), alongside the emergence of Augmented Reality (AR) and Mixed Reality (MR) devices, have significantly facilitated the acquisition and utilization of 3D data. To further enhance the utilization of 3D data across various applications, holographic-type communication [1], which allows high-quality 3D data communication with remote users, is being pursued in communication fields as a pivotal next-generation technology.

Point clouds [10] have been highlighted in recent years as a versatile 3D data format. A point cloud consists of numerous 3D points, with each point comprising 3D coordinates and color components. For the realization of holographic-type communication, point cloud transmission poses significant challenges due to the extensive traffic. Several coding methods have been proposed to reduce the traffic, including tree-based [6, 8], graph-based [11, 12], and 2D mapping-based [7, 9]. For example, geometry-based point cloud coding (G-PCC) and point cloud library (PCL), which are the typical tree-based methods, used a tree structure to compress the 3D geometry information. Graph-based coding methods, utilizing Graph Signal Processing (GSP), can efficiently minimize redundant information by applying Graph Fourier Transform (GFT) for 3D coordinates and color components. This involves constructing a graph signal where each 3D point is a vertex and the distances between 3D points are edge weights. The GFT can be defined by performing eigenvalue decomposition of the graph shift operator based on the edge weights.

The recent studies reported the graph-based PCC yields better rate-distortion performance compared to the tree-based coding methods. However, the graph shift operator requires setting seven parameters—derived from the adjacency matrix, the degree matrix, and hyperparameters to obtain better rate-distortion performance. The search space for these parameters is vast; with M potential values per parameter, the total combinations are approximately M^7. Moreover, eigenvalue decomposition for each parameter set takes $O(N^3)$ time complexity in a point cloud with N 3D points, leading to a total of $O(N^3M^7)$ computational cost required to find the optimal parameter set.

To address these challenges, this paper introduces a new graph-based PCC method that reduces the time complexity required for finding a better graph shift operator. Specifically, the proposed scheme realizes 1) parameter reduction that does not significantly impact rate-distortion performance, and 2) hyperparameter search without the need for additional eigenvalue decompositions by employing a similarity transformation for the graph shift operator.

Evaluations show that the proposed scheme achieves better 3D reconstruction quality under the same bandwidth constraints compared to existing graph-based coding methods and reduces computational time, offering substantial improvements over techniques that search all the potential hyperparameter sets.

2 GRAPH CONSTRUCTION FOR PCC

Graph structure-based PCC compresses point clouds by employing a GFT for the point cloud. For this purpose, each point cloud is modeled as a weighted, undirected graph signal $\mathcal{G} = (\boldsymbol{V}, \mathcal{E}, \boldsymbol{W})$, where $\boldsymbol{V}$ and $\mathcal{E}$ denote the vertex and edge sets of $\mathcal{G}$, respectively. The matrix $\boldsymbol{W}$ represents an adjacency matrix with positive weights, where the element $W_{i,j}$ denotes the weight of the edge between vertex i and vertex j. The degree matrix $\boldsymbol{D}$ is derived from the adjacency matrix $\boldsymbol{W}$ as follows:

$$\boldsymbol{D} = \mathrm{diag}(D_1, D_2, \cdots, D_N),\ D_i = \sum_{n=1}^{N} W_{i,n}, \tag{1}$$

where N is the number of points in the point cloud. The basis functions used in the GFT are defined by the graph laplacian operator $\boldsymbol{L}$, which is derived from the adjacency matrix $\boldsymbol{W}$ and the degree matrix $\boldsymbol{D}$. The graph shift operator $\boldsymbol{L}$ is formulated from seven hyperparameters as follows[4]:

$$\boldsymbol{L} = m_1 \boldsymbol{D}_a^{e_1} + m_2 \boldsymbol{D}_a^{e_2} \boldsymbol{W}_a \boldsymbol{D}_a^{e_3} + m_3 \boldsymbol{I}, \tag{2}$$

where $\boldsymbol{W}_a = \boldsymbol{W} + a\boldsymbol{I}$, and $\boldsymbol{D}_a$ is the degree matrix derived from the adjusted adjacency matrix $\boldsymbol{W}_a$. Here, $\boldsymbol{I}$ is the $N \times N$ identity matrix. The set $\boldsymbol{m} = (m_1, m_2, m_3, e_1, e_2, e_3, a) \in \mathbb{R}^7$ represents the hyperparameter set that defines the graph shift operator. The graph shift operator is generally similar to a real symmetric matrix, therefore possessing orthogonal eigenvectors and non-negative eigenvalues corresponding to each eigenvector. The eigenvalue decomposition of the graph shift operator $\boldsymbol{L}$ is defined as: $\boldsymbol{L} = \boldsymbol{\Phi}\boldsymbol{\Delta}\boldsymbol{\Phi}^{-1}$, where $\boldsymbol{\Phi}$ is a matrix of eigenvectors, and $\boldsymbol{\Delta}$ is a diagonal matrix containing the eigenvalues. The GFT coefficients $\boldsymbol{f}$, which represent the frequency components of each attribute, are calculated by projecting the attribute values $\boldsymbol{s}$ onto the basis functions $\boldsymbol{\Phi}$:$\boldsymbol{f} = \boldsymbol{s}\ \boldsymbol{\Phi}$. The corresponding bit sequence is then derived by quantization and entropy encoding of the GFT coefficients $\boldsymbol{f}$. On the receiving end, entropy decoding and dequantization are carried out to retrieve an estimate of the GFT coefficients $\hat{\boldsymbol{f}}$. The Inverse Graph Fourier Transform (IGFT) is finally utilized to reconstruct an estimate of each attribute $\hat{\boldsymbol{s}}$ from the estimated GFT coefficients and basis functions, defined as: $\hat{\boldsymbol{s}} = \hat{\boldsymbol{f}}\ \boldsymbol{\Phi}^{-1}$.

3 PROPOSED SCHEME

3.1 Overview

Fig. 1 presents an end-to-end architecture of the proposed point cloud encoder and decoder. It consists of a point cloud encoder and a decoder utilizing the GFT. The encoder models a point cloud as a weighted undirected graph signal $\mathcal{G} = (\boldsymbol{V}, \mathcal{E}, \boldsymbol{W})$, where $\boldsymbol{V}$ and $\mathcal{E}$ represent the sets of vertices and edges, respectively. $\boldsymbol{W}$ is an adjacency matrix with positive weights, with the (i, j) element $W_{i,j}$ indicating the weight of the edge between vertices i and j. For this model, the 3D coordinates $\boldsymbol{p} = [x, y, z]^T \in \mathbb{R}^{3\times N}$, consisting of N vertices, is treated as the vertex set of the graph $\mathcal{G}$. Each element $W_{i,j}$ of the adjacency matrix is defined by the following equation:

$$W_{i,j} = \exp\left(-\frac{||\boldsymbol{p}_i - \boldsymbol{p}_j||_2^2}{\kappa_p}\right), \tag{3}$$

where κ_p is a parameter representing the variance of the distance between two points. The bitstream after Huffman coding is then transmitted over wired and wireless networks, and the decoder reconstructs it back into a point cloud.

3.2 Encoder

The encoder generates the adjacency matrix $\boldsymbol{W}$ and the degree matrix $\boldsymbol{D}$ from the 3D coordinates of the input point cloud. The GFT coefficients $\boldsymbol{f}$ are then obtained by applying GFT for the 3D coordinates. These coefficients are subsequently quantized to $\boldsymbol{c}$ according to a predetermined quantization step size q. The step size q is defined by the quantization parameter r, ensuring that the range between the maximum and minimum quantized values is r. The quantization process involves:

$$\begin{aligned} \boldsymbol{c} &= \mathrm{round}\left(\boldsymbol{f}/q\right),\ q = \min q', \\ \text{s.t.}\ &\max\left(\mathrm{round}\left(\boldsymbol{f}/q'\right)\right) - \min\left(\mathrm{round}\left(\boldsymbol{f}/q'\right)\right) = r \end{aligned} \tag{4}$$

where round$(\cdot)$ denotes the rounding operation of each element. The quantized GFT coefficients $\boldsymbol{c}$ are then encoded into a bit string using Huffman coding, optimizing the data for efficient transmission.

3.2.1 *Parameter Reduction.*

To effectively minimize traffic while preserving the quality of point cloud reconstruction, it is essential to optimize the hyperparameter set $\boldsymbol{m} = (m_1, m_2, m_3, e_1, e_2, e_3, a)$ of graph shift operator and quantization parameter r under the given bandwidth constraint. When the number of searches for each parameter is M, this involves a search across M^8 parameter combinations. This exhaustive search is computationally expensive and inefficient. The proposed method simplifies the parameter space by focusing on parameters that significantly impact traffic reduction without compromising encoding effectiveness.

A critical observation from the definition of eigenvectors is that the variation of parameter m_3 does not affect the resulting eigenvectors. Based on this insight, we simplify the graph shift operator by setting $m_3 = 0$, reducing the complexity from seven parameters to six:

$$\boldsymbol{L}_{\text{6params}} = m_1 \boldsymbol{D}_a^{e_1} + m_2 \boldsymbol{D}_a^{e_2} \boldsymbol{W}_a \boldsymbol{D}_a^{e_3}. \tag{5}$$

Further simplification is achieved by normalizing m_1 with respect to m_2, defining $m_1' = \frac{m_1}{m_2}$, which allows us to consolidate these parameters into a single parameter m_1'. This transformation retains the essential characteristics of the operator while reducing the parameter space from six to five:

$$\boldsymbol{L}_{\text{5params}} = \frac{\boldsymbol{L}_{\text{6params}}}{m_2} = m_1' \boldsymbol{D}_a^{e_1} + \boldsymbol{D}_a^{e_2} \boldsymbol{W}_a \boldsymbol{D}_a^{e_3}. \tag{6}$$

The resultant graph shift operator $\boldsymbol{L}_{\text{5params}}$ still needs to be a real symmetric matrix for defining GFT, i.e., the eigenvectors is an orthogonal. This requirement ensures that $e_2 = e_3$, leading us to define a consolidated exponent $e_+ = \frac{e_2+e_3}{2}$, and thus we can

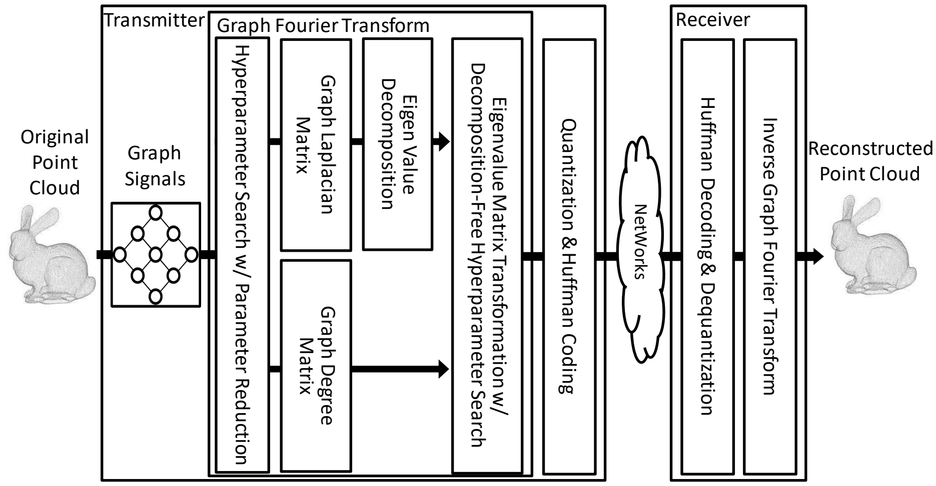


Figure 1: Overview of Proposed Scheme

represent the operator with four parameters:

$$L_{4\text{params}} = m'_1 D_a^{e_1} + D_a^{e_+} W_a D_a^{e_+}. \tag{7}$$

In summary, the proposed scheme can search the adequate basis function $\boldsymbol{\Phi}$, which achieves better rate-distortion performance, from a four hyperparameter set $\boldsymbol{m'} = (m'_1, e_1, e_+, a)$ and a quantization parameter q.

3.2.2 Decomposition-Free Hyperparameter Search. Utilizing the graph shift operator $L_{4\text{params}}$, as defined in the previous section, allows for a quick determination of the adequate basis function $\boldsymbol{\Phi}$. However, obtaining basis functions involves an eigenvalue decomposition for each hyperparameter set $\boldsymbol{m'} = (m'_1, e_1, e_+, a)$. It is typically requiring $O(N^3)$. To address this computational challenge, we propose a novel approach that leverages a similarity transformation, reducing the need for repeated eigenvalue decompositions during the hyperparameter search.

The similarity transformation is applied to the graph shift operator using the degree matrix D_a, defined as follows.

$$L_{5\text{params+}} = D_a^{e_-} L_{4\text{params}} D_a^{-e_-} = m'_1 D_a^{e_1} + D_a^{e_+ + e_-} W_a D_a^{e_+ - e_-} \tag{8}$$

where e_- represents a transformation parameter defined as $\frac{e_2 - e_3}{2}$. This transformation maintains the structural integrity of the graph while enabling parameter adjustments that influence the eigenvector configurations. The transformed eigenvector matrix $\boldsymbol{\Phi'}$ relates to the original eigenvector matrix $\boldsymbol{\Phi}$ from $L_{4\text{params}}$ via:

$$\boldsymbol{\Phi'} = D_a^{e_-} \boldsymbol{\Phi}, \ \boldsymbol{\Phi'}^{-1} = \boldsymbol{\Phi}^{-1} D_a^{-e_-} \tag{9}$$

where $D_a^{-e_-}$ adjusts the scaling of the eigenvectors without necessitating new eigenvalue decompositions.

From Eq. (9), by applying the newly introduced hyperparameter e_- to transform the eigenvector matrix derived from the hyperparameter set $\boldsymbol{m'} = (m'_1, e_1, e_+, a)$, the proposed scheme can generate different basis functions. This process does not require a new eigenvalue decomposition of the transformed eigenvector matrix. This modification enhances computational efficiency by utilizing existing eigenvector matrices. Additionally, the set of possible graph shift operators for $L_{5\text{params}}$ and $L_{5\text{params+}}$ are equivalent. This equivalence ensures that the optimized rate-distortion performance is maintained even without additional decompositions.

3.3 Decoder

The decoder reconstructs the received bitstream into $\hat{\boldsymbol{c}}$, which is an estimate of the quantized GFT coefficients. The GFT coefficients are decoded by applying the quantization step size q that was used in the encoder. The decoding process is expressed as: $\hat{\boldsymbol{f}} = q \ \hat{\boldsymbol{c}}$. Subsequently, an estimate $\hat{\boldsymbol{s}}$ of the 3D coordinates for each 3D point is derived using the IGFT. The IGFT is applied utilizing $\boldsymbol{\Phi'}$ as the basis function.

3.4 Hyperparameter Learning from Training Point Clouds

The optimized hyperparameter set for the given point cloud can potentially improve the rate-distortion performance across different point clouds. In this case, the time complexity is reduced to $O(1)$. For this purpose, the hyperparameter sets $\boldsymbol{m'}$ can be optimized using a training dataset comprised of point clouds. The optimization aims to minimize the Bjøntegaard delta (BD) rate, which is defined in relation to both the Chamfer Distance [3, 5] and the entropy of the quantized GFT coefficients. The Chamfer Distance measures the similarity between two point clouds based on the displacement of their 3D coordinates, and is calculated as follows:

$$d = \frac{1}{2|S|} \sum_{\boldsymbol{p} \in S} \min_{\widehat{\boldsymbol{p}} \in \widehat{S}} ||\boldsymbol{p} - \widehat{\boldsymbol{p}}||_2 + \frac{1}{2|\widehat{S}|} \sum_{\widehat{\boldsymbol{p}} \in \widehat{S}} \min_{\boldsymbol{p} \in S} ||\boldsymbol{p} - \widehat{\boldsymbol{p}}||_2 \tag{10}$$

where S is the set of points in the source point cloud, and $\widehat{S}$ is the set of points in the decoded point cloud. Entropy, which serves as a lower bound on the average bit length required for data communication, is defined by the distribution of quantized GFT coefficients:

$$E = -\sum_{q \in \boldsymbol{c}} P(q) \log_2 P(q) \tag{11}$$

where $\boldsymbol{c}$ represents the GFT coefficients after quantization, and $P(q)$ is the probability of quantized value q.

By minimizing both Chamfer Distance and entropy, the parameter set aims to ensure that each point in the received point cloud closely matches its counterpart in the original point cloud, while simultaneously reducing the required traffic. However, Chamfer Distance and entropy typically exhibit a trade-off relationship.

Bjøntegaard delta (BD) rate [2] is used as a relative measure to compare different hyperparameter sets by evaluating the trade-offs between point cloud fidelity (as measured by Chamfer Distance) and required traffic (as quantified by entropy). The BD rate is calculated using the following integral, which represents the average logarithmic difference in entropy between two hyperparameter configurations over the range of observed Chamfer Distances:

$$\text{BD rate} = 10^{\frac{1}{D_H - D_L}\int_{D_L}^{D_H}(\log E_2(d_2) - \log E_1(d_1))dr} \tag{12}$$

$$D_H = \min\left(\max_{d_1 \in C_1}(d_1), \max_{d_2 \in C_2}(d_2)\right)$$

$$D_L = \max\left(\min_{d_1 \in C_1}(d_1), \min_{d_2 \in C_2}(d_2)\right)$$

$$\boldsymbol{C} = \{(d, E) \mid \{(d', E') \mid d' < d \wedge E' < E\} = \emptyset\}$$

In this paper, one hyperparameter set is fixed during the calculation of the BD rate and serves as a benchmark. Here, the combination graph Laplacian matrix specified in Eq. (15) is used as the benchmark. Minimizing the BD rate improves the quality of the reconstructed point cloud and reduces traffic for effective transmission.

When we consider M candidate values for each parameter of $\boldsymbol{m'} = (m'_1, e_1, e_+, a)$, a total M^4 hyperparameter sets are utilized for learning. During the learning phase, for each point cloud in the training dataset, the top $K\%$ of hyperparameter sets with the lowest BD rates are selected. If the total number of point clouds in the training dataset is T, then the total number of candidate sets for each parameter is $M^4 \times K \times T$. Finally, the optimal value for each parameter is determined as the mode—the most frequently occurring value among the candidates. For this study, we set $K = 10\%$ and $T = 3$.

4 EVALUATION

4.1 Settings

Dataset: To evaluate the rate distortion performance of the proposed scheme, we used ShapeNet point cloud dataset. We focused on objects categorized as "Chair".

Metric: Quality of the point cloud was measured using the Chamfer Distance, as defined in Eq. (10), and traffic was measured using entropy, as outlined in Eq. (11).

Comparison Methods: We compared the proposed scheme against three widely recognized graph shift operators: "Regular", "Signless", and "Combination". Each operator is defined as follows:

$$\text{Regular}: \boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W} \tag{13}$$

$$\text{Signless}: \boldsymbol{L} = \boldsymbol{D} + \boldsymbol{W} \tag{14}$$

$$\text{Combination}: \boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-0.5}\boldsymbol{W}\boldsymbol{D}^{-0.5} \tag{15}$$

These graph shift operators are referred to as well-known graph Laplacian operators.

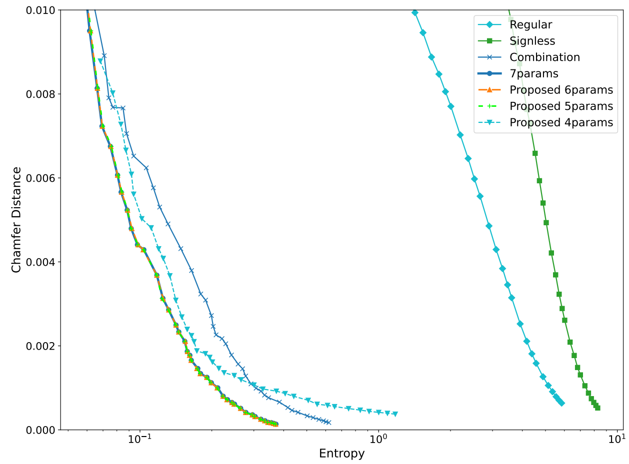


Figure 2: 3D reconstruction quality as a function of traffic under the different graph shift operators. Here, the performance curve of "7params" is overlapped with that of "proposed 6params" and "proposed 5params".

Proposed Scheme Variants: The variants of the proposed scheme are labeled as follows for clarity:

- The graph shift operator defined in Eq. (2) is referred to as "7params".
- The operators from Eqs. (5), (6), (7), and (8) are labeled "Proposed 6params", "Proposed 5params", "Proposed 4params", and "Proposed", respectively.
- The operator determined through learning from the training dataset is designated as "Proposed Learning".

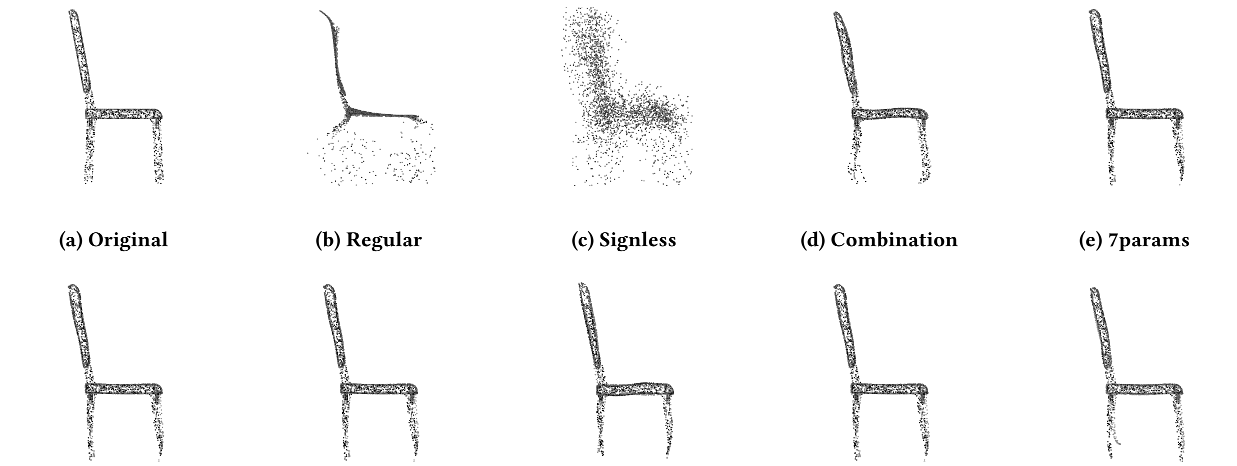
Measurement Setup: For the measurement of a rate-distortion curve, graph shift operators except "Proposed" and "Proposed Learning" were obtained by varying the quantization parameter r. For "Proposed" and "Proposed Learning", we varied both r and the transformation parameter e_-.

Execution Environment: The evaluations were performed using an Intel(R) Xeon(R) Silver 4108 CPU @ 1.80 GHz, an NVIDIA Quadro GV100 GPU, and Python 3.8.10.

4.2 Effect of Hyperparameter Reduction

In this section, we demonstrate how reducing the number of parameters in a hyperparameter set affects the compression performance of point clouds. For comparison, we used "7params," which has the most parameters, and the reduced-parameter schemes "Proposed 6params", "Proposed 5params", and "Proposed 4params". Fig. 2 displays the Chamfer Distance and entropy for each graph shift operator. The evaluation results reveal the following insights:

- "Proposed 6params" and "Proposed 5params" achieve better point cloud quality at the same entropy compared to well-known graph Laplacian operators
- "Proposed 6params" and "Proposed 5params" perform nearly as well as "7params".



(a) Original (b) Regular (c) Signless (d) Combination (e) 7params

(f) Proposed 6params (g) Proposed 5params (h) Proposed 4params (i) Proposed (j) Proposed Learning

Figure 3: Visual performance of the reconstructed point cloud in each graph shift operator. ere, the entropy of regular, signless, and other schemes is approximately 0.3, 0.9, and 0.1, respectively.

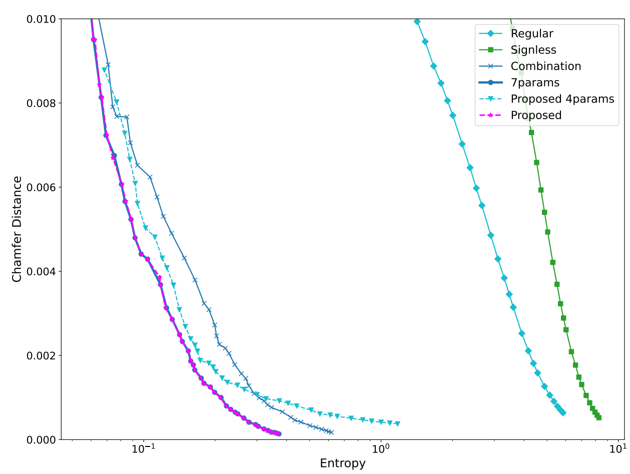


Figure 4: 3D reconstruction quality as a function of traffic under the different graph shift operators. Here, "Proposed" used decomposition-free hyperparameter search.

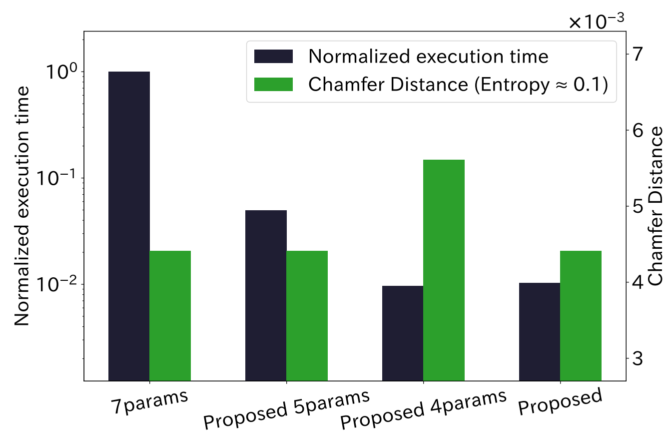


Figure 5: Normalized execution time and Chamfer Distance in "7params" and each proposed scheme. Here, the entropy of each schemes is 0.1, respectively.

- "Proposed 4params" shows inferior point cloud quality compared to "7params", and under certain conditions, performs worse than Combination in terms of entropy.

Fig. 3 visualizes the original point cloud alongside the reconstructed point cloud in each graph shift operator. Distortions are observable in the back and legs across all schemes. Additionally, the seat surface shows notable distortions with Combination and "Proposed 4params".

4.3 Impact of Decomposition-Free Parameter Search

This section explores the impact of the proposed decomposition-free parameter searches on the quality of the reconstructed point cloud, required traffic, and time complexity. For comparison, we used "7params" and two of the proposed schemes "4params" and "Proposed". Fig. 4 illustrates the relationship between Chamfer Distance and entropy for each graph shift operator. The evaluation reveals "Proposed", which realizes decomposition-free parameter search, enhances reconstruction quality over "Proposed 4params", underscoring the effectiveness of its parameter search approach.

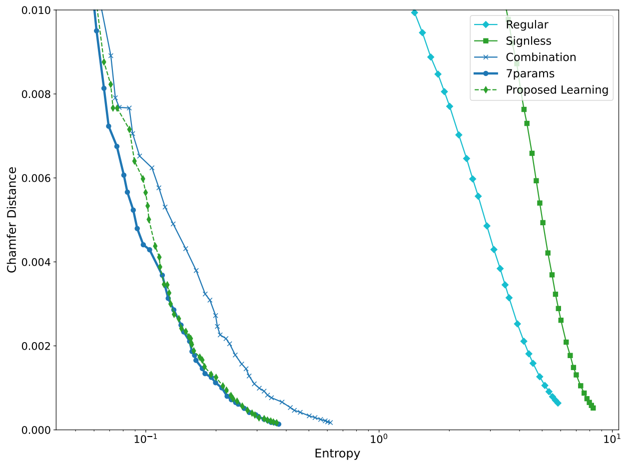


Figure 6: 3D reconstruction quality as a function of traffic under the different graph shift operators. Here, "Proposed Learning" used the trained hyperparameters from the training point cloud dataset.

It suggests that decomposition-free parameter search does not degrade the reconstructed point cloud quality. Therefore, the proposed scheme can serve as a viable alternative to conventional search methods, potentially offering computational efficiency without sacrificing rate-distortion performance.

We then evaluate the execution time across the proposed scheme variants to discuss the effect of the decomposition-free parameter search on the time complexity. Fig. 5 shows the normalized execution times and Chamfer Distances for each proposed scheme. Note that the execution time for "7params" and "Proposed 6params" are estimates based on the execution time for one parameter set, while those for other schemes are based on actual measurements. The measurement results show that "Proposed" achieves significant reductions in search time compared to computation-hungry schemes, e.g., "7params", without the degradation of 3D reconstruction quality.

In summary, by incorporating a decomposition-free search, "Proposed" achieves execution times similar to "Proposed 4params" but offers better point cloud compression performance. This demonstrates the efficiency of the proposed scheme in optimizing both the computational time and the reconstructed quality of point clouds.

4.4 Effect of Trained Parameter Set

Finally, this section evaluates the compression performance by applying the hyperparameters learned from a training dataset to other point clouds. We used "7params" as an ideal performance and "Proposed Learning" to discuss the effect of the learned hyperparameters. Fig. 6 shows the Chamfer Distance and entropy for each graph shift operator. It demonstrates that applying hyperparameters derived from the training point cloud dataset enhances point cloud compression performance over well-known graph Laplacian operators. This finding suggests that the same graph shift operator could be reused for different point clouds without additional computation costs.

5 CONCLUSION

In this paper, we propose a graph-based novel PCC scheme to simultaneously achieve rate-distortion optimization and expedite hyperparameter optimization. For this purpose, the proposed scheme reduces the number of hyperparameters by leveraging the inherent properties of the graph shift operator. Additionally, it realizes the hyperparameter search without eigenvalue decomposition, effectively enhancing both the rate-distortion performance of 3D point cloud compression and the time complexity of hyperparameter optimization. Evaluations verified that the proposed scheme significantly decreases the time needed for hyperparameter search while maintaining rate-distortion performance compared to computation-hungry existing schemes. Moreover, the proposed scheme demonstrated robust point cloud compression performance using hyperparameter sets derived from the training point cloud dataset, and thus new parameter searches can be eliminated for the compression.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number JP22H03582.

REFERENCES

[1] Ian F. Akyildiz and Hongzhi Guo. 2022. Holographic-type communication: A new challenge for the next decade. *ITU Journal on Future and Evolving Technologies* 3 (2022), 421–442. Issue 2.
[2] Gisle Bjøntegaard. 2001. Calculation of Average PSNR Differences between RD-curves. https://api.semanticscholar.org/CorpusID:61598325
[3] Siheng Chen, Chaojing Duan, Yaoqing Yang, Duanshun Li, Chen Feng, and Dong Tian. 2020. Deep Unsupervised Learning of 3D Point Clouds via Graph Topology Inference and Filtering. *IEEE Transactions on Image Processing* 29 (2020), 3183–3198.
[4] George Dasoulas, Johannes Lutzeyer, and Michalis Vazirgiannis. 2021. Learning Parametrised Graph Shift Operators. In *International Conference on Learning Representations*. ICLR, Virtual, 1–17.
[5] Haoqiang Fan, Hao Su, and Leonidas J Guibas. 2017. A Point Set Generation Network for 3D Object Reconstruction from A Single Image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Honolulu, HI, USA, 2463–2471.
[6] Google. 2024. Draco 3D data compression. https://google.github.io/draco/
[7] D Graziosi, O Nakagami, S Kuma, A Zaghetto, T Suzuki, and A Tabatabai. 2020. 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 (2020), 17.
[8] Bo Han, Yu Liu, and Feng Qian. 2020. ViVo: Visibility-aware mobile volumetric video streaming. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. Association for Computing Machinery, New York, NY, USA, 1–13.
[9] Euee S Jang, Marius Preda, Khaled Mammou, Alexis M Tourapis, Jungsun Kim, Danillo B Graziosi, Sungryeul Rhyu, and Madhukar Budagavi. 2019. Video-based point-cloud-compression standard in mpeg: From evidence collection to committee draft [standards in a nutshell]. *IEEE Signal Processing Magazine* 36, 3 (2019), 118–123.
[10] Rufael Mekuria and Lazar Bivolarsky. 2016. Overview of the MPEG activity on point cloud compression. In *Data Compression Conference (DCC)*. IEEE, UT, United States, 620–620.
[11] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José M. F. Moura, and Pierre Vandergheynst. 2018. Graph Signal Processing: Overview, Challenges, and Applications. *Proc. IEEE* 106, 5 (2018), 808–828.
[12] Jingshu Zhang, Yueru Chen, Guoqing Liu, Wei Gao, and Ge Li. 2024. Efficient Point Cloud Attribute Compression Framework using Attribute-Guided Graph Fourier Transform. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, Seoul, South Korea, 8426–8430.