Rate Control for Point Cloud Compression with Adaptive Updates to the Rate-Quantization Model

Yohei Hanaoka, Kohei Matsuzaki, Kyohei Unno†, Keisuke Nonaka
KDDI Research, Inc., Saitama, Japan
{yo-hanaoka, ko-matsuzaki, ky-unno, ki-nonaka}@kddi.com

Abstract—**In this paper, we propose a rate control method based on the Octree coding of Geometry-based Point Cloud Compression (G-PCC) for LiDAR point clouds. In conventional methods, the parameter estimation for the Rate-Quantization (R-Q) model relies only on frame-internal characteristics. While this approach facilitates the identification of general trends in the encoding efficiency of LiDAR point clouds and provides an approximation of the relationship between bitrate and quantization scale, accurate estimation of the R-Q model parameters for each frame remains a challenge. To address this issue, our method updates the R-Q model using temporal correlations among frames, specifically by employing adaptive Least Mean Square (LMS) with our dynamic control of its step size and adaptive bit allocation for geometry and attribute information. Experimental results demonstrate an improvement of 99.7% in error rate compared to conventional methods.**

Index Terms—**point cloud, compression, G-PCC, rate control.**

I. Introduction

3D point clouds are spatial data utilized in various fields, such as construction, autonomous driving, and extended reality, and the demand for real-time delivery of the data is growing [1]. A key method for achieving real-time delivery of 3D point clouds is LiDAR scanning, which enables the real-time capture of sparse 3D point clouds (hereinafter referred to as LiDAR point clouds). On the other hand, real-time delivery of LiDAR point clouds presents the challenge of large data sizes which typically require a bandwidth of several to tens of Mbits per second, making compression of LiDAR point clouds essential.

Recent approaches for compressing 3D point clouds include deep learning-based techniques [2] [3] [4], which exhibit advanced compression performance. However, these methods require substantial GPU computing power and are time-consuming, making them unsuitable for real-time transmission on portable devices. Alternatively, Geometry-based Point Cloud Compression (G-PCC) [5] is a lightweight coding method standardized by the Moving Picture Experts Group (MPEG), known for its international standards for video and audio data encoding. This method is suitable for real-time processing because of its low computational complexity. G-PCC enables lossy compression by quantization parameters for both geometry and attribute information. However, in practice, the encoding efficiency and the number of points in LiDAR point clouds can vary depending on the scene, even with the same sensor, leading to fluctuations in bitrate. Currently, the ability to achieve a desired bitrate is absent in reference software [6]. Moreover, in practical use cases, the network bandwidth may not always be constant. Therefore, to reliably deliver LiDAR point clouds, rate control methods capable of achieving the desired bitrate are necessary.

The G-PCC reference software [6] provides tools for encoding geometry, such as Octree and Predictive tree, which can be selected by the encoder [7]. Predictive tree is particularly suitable for 3D point clouds obtained from spinning LiDAR, as it utilizes the geometric features of laser scans from spinning LiDAR. On the other hand, Octree is independent of the scanning methods of LiDARs, allowing it to be widely utilized without restricting the type of LiDAR. This paper focuses on the generic applicability of the Octree.

Li et al. [8] proposed a rate control method using the Octree for G-PCC applied to LiDAR point clouds. They created a Rate-Quantization (R-Q) model that describes the relationship between bitrate and quantization parameters and proposed a procedure to update the model parameters using frame-internal predictions for each frame. Specifically, they estimate the parameters using the density of the point cloud for each frame. However, considering the signal characteristics of LiDAR point clouds, where their distribution changes with each frame, parameter estimation based solely on one frame-internal feature, density, is insufficient. Furthermore, in the Li et al.'s method, bit allocation for geometry and attribute is fixed throughout the sequence, which also makes accurate rate control difficult.

To address this issue, we focus on the fact that the bitrate fluctuates continuously and propose a method that updates the R-Q model using temporal correlations. Specifically, we utilize the adaptive Least Mean Square (LMS) [9] within the G-PCC rate control framework, updating model parameters for each frame based on information from the previous frame. Furthermore, based on our observation that the appropriate update step size for the LMS can differ from frame to frame, our method dynamically controls the step size. After that, the updated R-Q model is used to perform bit allocation for the next frame, assigning bits for geometry and attribute. These characteristics of our method realize the accurate rate control. The contributions of our research are as follows: 1) We propose an accurate rate control method that updates the R-Q model using temporal correlations, specifically by employing LMS and dynamically controlling its update step size. 2) We propose a method for appropriate bit allocation using the R-Q model updated for each frame.

† Current affiliation: Science and Technology Research Laboratories, NHK, Tokyo, Japan.

ISBN: 978-9-46-459362-4

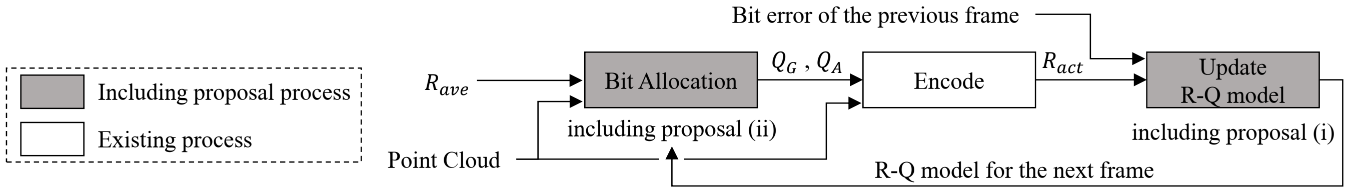


Fig. 1. Overview diagram of the proposed method.

II. Related Work

A. G-PCC

MPEG has standardized two main methods based on use cases: Video-based Point Cloud Compression (V-PCC) [10] and G-PCC [5]. V-PCC is particularly suited for encoding dense and dynamic point cloud data, such as that of people. In contrast, G-PCC is a technique that compresses 3D point clouds while retaining their 3D data structure, making it applicable to sparse point clouds obtained from LiDAR. G-PCC encodes geometry and attribute separately. One of the tools for compressing geometry is the Octree, which recursively subdivides the entire target space into an octree structure, enabling high-efficiency point cloud compression through arithmetic coding that uses the occupancy state of neighboring spaces as context. Additionally, several tools are available for attribute compression [7].

In G-PCC's lossy compression, the quantization parameters Q_G and Q_A control the quantization of geometry and attribute, respectively. The coordinates of 3D point clouds are transformed and duplicate points are merged before being encoded, based on Q_G. After encoding the geometry, attribute values are reattached to the reconstructed point cloud, followed by a transformation governed by Q_A [8]. Therefore, the bitrate of attribute is influenced by the quantization of the geometry.

B. Rate Control for Point Cloud Compression

Rate control is a technique that achieves a desired bitrate by adjusting factors that affect the bitrate, such as quantization parameters. Generally, in rate control for point cloud compression, the first step is to allocate bits to geometry and attribute while minimizing overall distortion to ensure that the target bitrate is not exceeded. Subsequently, the quantization parameters are derived from the rate model to achieve the allocated rate. In recent years, several rate control methods for point cloud compression have been proposed. Various rate-distortion models for optimal bit allocation have been introduced [11] [12] [13]. These methods are specific to rate control in V-PCC. Wang et al. [14] proposed a rate control method under the point cloud compression standard developed by the Audio Video Coding Standard Workgroup of China [15]. Furthermore, Zhang et al. [16] proposed a rate control for dense and dynamic human point clouds in G-PCC, which cannot be directly applied to rate control using LiDAR sensors targeted in this paper due to differences in the content and characteristics of the point cloud data.

For the rate control based on the G-PCC for LiDAR point clouds, several approaches have also been proposed. Hou et al. [17] proposed a rate control method using Predictive tree, specifically targeting spinning LiDAR. This method utilizes parameters specific to spinning LiDAR, limiting the types of LiDAR that can be employed. Li et al. [8] proposed a rate control method for LiDAR point clouds using Octree. This approach constructs an R-Q model and allocates bitrates to both geometry and attributes, adjusting the R-Q model parameters for each frame through frame-internal estimation to achieve the desired target bitrate.

This paper focuses on the generic applicability of Octree, building upon the method proposed by Li et al. [8]. Nonetheless, a challenge with Li et al.'s method is that it estimates model parameters using only frame-internal features, making it difficult to estimate an accurate R-Q model for each frame.

III. Proposed method

A. Overview

The overview of the proposed method is illustrated in Fig. 1. Initially, the target bitrate R_{tar} to be allocated for the current frame is determined based on the average target bitrate R_{ave}. Here, the average target bitrate R_{ave} can be expressed as $R_{ave} = R_T/N$, using the full frame level total target bitrate R_T and the number of frames N. The target bitrate R_{tar} for the current frame is allocated using the following model, in accordance with the method proposed by Li et al. [8]:

$$R_{tar} = R_{ave} + \frac{(R_T^L - R_{ave} \cdot N^L)}{SW}, \tag{1}$$

where R_T^L denotes the bitrate that can be allocated to the remaining frames, N^L is the number of remaining frames, and SW refers to the sliding window.

Once R_{tar} is determined, the quantization parameters for geometry and attribute, Q_G and Q_A, for the current frame are derived from the point cloud and the R-Q model. Then, bitrates for geometry R_G and attribute R_A are allocated accordingly. The R-Q model parameters start with initial values for the first frame and are then updated for subsequent frames after the second. After determining Q_G and Q_A, encoding of the point cloud is performed using G-PCC.

Finally, the actual bitrate R_{act} obtained from the encoding results and the computed bitrate R_{comp} are used to update the R-Q model for the next frame. The computed bitrate R_{comp} can be calculated based on Q_G used in the encoding and (5). For the subsequent frame, the bit allocation process is repeated based on the updated R-Q model.

In this process, our proposal comprises the following two points, each of which can be referenced in Fig. 1. Proposal (i):

To accurately estimate the parameters of the R-Q model for each frame, we propose a method that updates the R-Q model using temporal correlations, specifically by employing LMS and dynamically controlling its update step size. Proposal (ii): For optimal bit allocation for each frame, we propose a bit allocation method based on the R-Q model updated for each frame. The following sections provide a detailed description of each proposal.

B. R-Q Model Update using Temporal Correlation

The R-Q model proposed by Li et al. [8] is expressed as follows:

$$R_G = \alpha_G \ln Q_G + \beta_G, \tag{2}$$

$$R_A = \alpha_A(Q_G) Q_A^{\beta_A}, \tag{3}$$

$$\alpha_A(Q_G) = a \ln Q_G + b, \tag{4}$$

$$R_T = \alpha_G \ln Q_G + \beta_G + (a \ln Q_G + b)(cQ_G^d)^{\beta_A}, \tag{5}$$

where Q_G and Q_A are the quantization parameters for geometry and attribute, α_G and β_G are parameters for the R-Q model of geometry, while α_A and β_A are parameters for the R-Q model of attribute. As mentioned earlier, since R_A is influenced by Q_G, α_A becomes a function of Q_G as shown in (4). β_A, however, does not depend on Q_G and therefore is not a function of Q_G. a and b describe the relationship between α_A and Q_G, and R_T represents the total bitrate comprising both R_G and R_A. Additionally, following the approach of Li et al. [8], we define Q_A based on G-PCC CTC [18] as:

$$Q_A = cQ_G^d, \tag{6}$$

where c and d are the parameter of the relationship between Q_G and Q_A. The updates for α_G, β_G, a, and b using LMS [9] are carried out as follows:

$$\alpha_{G_{i+1}} = \alpha_{G_i} + \delta_{\alpha_G}(R_{act} - R_{comp}) \ln Q_G, \tag{7}$$

$$\beta_{G_{i+1}} = \beta_{G_i} + \delta_{\beta_G}(R_{act} - R_{comp}), \tag{8}$$

$$a_{i+1} = a_i + \delta_a (R_{act} - R_{comp})(cQ_G^d)^{\beta_A} \ln Q_G, \tag{9}$$

$$b_{i+1} = b_i + \delta_b (R_{act} - R_{comp})(cQ_G^d)^{\beta_A}, \tag{10}$$

where α_{G_i}, β_{G_i}, a_i, and b_i are the R-Q model parameters for the i-th frame. δ_{α_G}, δ_{β_G}, δ_a, and δ_b represent the step sizes for each parameter in the LMS. Notably, the parameter β_A has been found, from our preliminary validation, to exhibit little variation between frames; therefore, it is not updated.

Additionally, the optimal step size may vary between frames. 3D Point clouds obtained while moving, such as those from vehicle-mounted LiDAR, often capture rapidly changing environments, leading to significant variations in spatial distribution. This variability can cause fluctuations in encoding efficiency within Octree, resulting in changes to the R-Q model. When the R-Q model changes significantly, rate control errors may increase. Therefore, increasing the LMS update step size allows for better tracking of the previous frame's R-Q model, improving accuracy. Conversely, when spatial distribution changes are minimal, a large step size can lead to unnecessary parameter updates and reduced rate control precision. Thus, a uniform step size for LMS updates is inadequate; it should be adjusted for each individual frame.

To address this, we propose a method that enhances accuracy and enables stable rate control through Dynamic Step size Control (DSC). The step size is updated for each frame based on the following equation:

$$\delta_X = \begin{cases} \theta \delta_X^{base} & \text{if } \theta < \frac{|E_{cur}|}{\bar{E}_i}, \\ \frac{|E_{cur}|}{\bar{E}_i} \delta_X^{base} & \text{if } \frac{1}{\theta} < \frac{|E_{cur}|}{\bar{E}_i} < \theta, \\ \frac{1}{\theta} \delta_X^{base} & \text{if } \frac{|E_{cur}|}{\bar{E}_i} < \frac{1}{\theta}, \end{cases} \tag{11}$$

where δ_X denotes the step size of the parameter X, which can be any of the following parameters: α_G, β_G, a, or b. δ_X^{base} represents the base value of δ_X. θ is the threshold of the step size. θ serves as the maximum threshold for the step size, whereas the minimum threshold for the step size is defined as $1/\theta$. E_{cur} is the difference between the actual bitrate R_{act} obtained from encoding and the computed bitrate R_{comp}, and $\bar{E}_i$ denotes the cumulative average of the absolute values of the difference up to the i-th frame.

However, if the step size is determined solely by the magnitude of the rate error as described above, the following issue may arise: if the R-Q model is updated excessively, it is possible that a significant error will still be detected in the subsequent frame, which could lead to setting a large step size. If this occurs continuously, the rate may fluctuate around the target bitrate without stabilizing. To prevent this, we propose that the step size should also be reduced if unstable behavior around the target bitrate is detected based on the following condition:

$$\delta_X = \frac{1}{\theta} \delta_X^{base} \qquad \text{if } E_{cur} \cdot E_{pre} < 0, \tag{12}$$

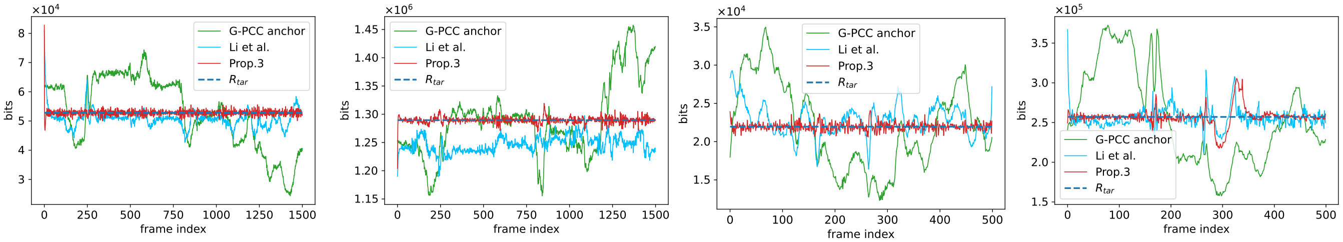
where E_{pre} is the difference between the actual bitrate and the computed bitrate from the previous frame. The step size control based on (11) is performed only when the condition in (12) is not met.

C. Adaptive Bit Allocation

As discussed in Section III-A, the appropriate R-Q model varies for each frame. On the other hand, Li et al. [8] allocate bits for geometry and attribute using pre-fitted R-Q model parameters, resulting in a uniform allocation across all frames. In other words, their method does not reflect the R-Q model updates in each frame. Even with accurate estimation of the R-Q model for each frame, poor accuracy in bit allocation for geometry and attribute can hinder effective rate control. We propose Adaptive Bit Allocation (ABA), which achieves optimal bit allocation by utilizing the parameters from the frequently updated R-Q model. The total bitrate for the i-th frame to be expressed as:

$$\frac{R_{tar_i}}{M_i} = \alpha_{G_i} \ln Q_G + \beta_{G_i} + (a_i \ln Q_G + b_i)(cQ_G^d)^{\beta_A}. \tag{13}$$

where R_{tar_i} denotes the target bitrate for the i-th frame, M_i is the number of points in the i-th input point cloud. From both (13) and (2), Q_G for the i-th frame can be determined, and



(a) ford_01_q1mm r01 (b) ford_01_q1mm r06 (c) qnxadas-motorway-join r01 (d) qnxadas-motorway-join r06

Fig. 2. Four examples of bitrate per frame.

TABLE I
EXPERIMENTAL RESULTS FOR THE ERROR RATE.

	G-PCC anchor	Li et al. [8]	Prop.1	Prop.2	Prop.3
CBR	8.20%	3.71%	2.89%	2.78%	**1.08%**
ABR	-	2.86%	2.04%	2.04%	**0.01%**

Q_A can be derived from (6). Hence, by utilizing the regularly updated R-Q models, we can achieve optimal bit allocation.

IV. EXPERIMENTS

A. Experimental Settings

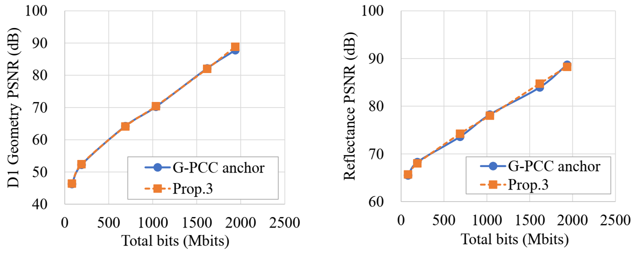
We implemented the proposed method on the latest software compliant with G-PCC version 1, specifically G-PCC TMC13-v12.10 [6]. We evaluated three patterns of the proposed method: LMS only (Prop.1), LMS and DSC (Prop.2), LMS, DSC, and ABA (Prop.3). We also performed reference experiments with G-PCC anchor (v12.10) and Li et al. [8]. The method of Li et al. was implemented and run on version 12.10. We utilized Octree and RAHT, conducting experiments under lossy conditions. The experimental data comprised the G-PCC's am-frame spinning class of LiDAR sequences, consistent with [8]. The average target bitrate R_{ave} was obtained from the experimental results of G-PCC anchor across rate points r01 to r06, in accordance with G-PCC CTC. The initial parameters of the R-Q model, as represented in (2) – (4), were retrieved using the results obtained earlier, following the same methodology as [8].

The evaluation metrics include the error rate of rate control, Bjontegaard delta rate (BD rate), and encoding time. For evaluating the error rate of rate control, we tested two methods: Constant Bit Rate (CBR) and Average Bit Rate (ABR) [19].

Furthermore, the parameters for the current validation were set as follows: the initial step sizes in LMS were $(\delta^{base}_{\alpha_G}, \delta^{base}_{\beta_G}, \delta^{base}_{a}, \delta^{base}_{b}) = (0.05, 0.5, 0.5, 0.5)$. The threshold θ for the step size in DSC was set to 1.5, and SW in (1) was set to 4. The specification of the PC used for the experiments was a 12th Gen Intel® Core™ i9-12900K CPU@3.20 GHz.

B. Experimental Results for the Error Rate

The experimental results for the error rate of rate control are shown in Table I. As previously mentioned, the average target bitrate R_{ave} was obtained by executing G-PCC anchor; thus, the ABR evaluation for G-PCC anchor is excluded. Prop.3 achieved the best accuracy with the CBR error rate of 1.08%. Additionally, for the ABR, Prop.3 attained the lowest error rate of 0.01%, representing an improvement of 99.7% compared to the method of Li et al. The error rate shows a tendency for improvement compared to the literature in [8], which is believed to be due to the upgrade of the G-PCC version. Prop.2 showed a 3.8% improvement in the CBR error rate compared to Prop.1, highlighting the effectiveness of DSC. The absence of improvement in ABA is likely due to ABR calculating the error rate based on the total sum of errors across all frames, which does not capture the subtle improvements observed in CBR. Moreover, Prop.3 showed a 61.2% improvement in the CBR error rate compared to Prop.2, indicating the effectiveness of ABA.



(a) D1 Geometry (b) Reflectance

Fig. 3. RD curve of ford_01_q1mm.

Table II compares the CBR error rates for various sequences and rate points between the method of Li et al. and Prop.3. Across all rate points and sequences, Prop.3 exhibits outstanding performance.

Figure 2 shows the rate transitions for the ford_01_q1mm and qnxadas-motorway-join sequences. It can be observed that the proposed method maintains stable control in both low-rate (r01) and high-rate (r06) bands.

C. Experimental Results for the BD Rate

Table III presents the results of BD rate and the ratio of Enc time for each method compared to the G-PCC anchor. The proposed method demonstrates gains of -0.8% for Geometry and -3.0% to -3.2% for Attribute. Figure 3 shows the RD curve of ford_01_q1mm. Additionally, the encoding time was found to be nearly equivalent to that of G-PCC anchor.

Table IV compares the BD rates of each point cloud between G-PCC anchor and Prop.3. This demonstrates the effectiveness of the proposed method across various datasets. Although the results of Li et al. show some deviation from the literature in [8], it is believed to be due to the impact of updating the G-PCC version, as mentioned earlier. Additionally, as

TABLE II
COMPARISON OF CBR ERROR RATE BETWEEN LI ET AL. AND THE PROP.3.

Point Cloud	Li et al. [8]						Prop.3					
	r01	r02	r03	r04	r05	r06	r01	r02	r03	r04	r05	r06
ford_01_q1mm	5.5%	2.8%	5.8%	4.0%	0.9%	3.3%	1.2%	0.9%	0.7%	0.4%	0.3%	0.3%
ford_02_q1mm	8.1%	3.5%	6.6%	4.4%	1.3%	3.7%	1.2%	0.9%	0.8%	0.5%	0.4%	0.3%
ford_03_q1mm	4.5%	2.2%	4.3%	3.4%	1.2%	2.1%	1.1%	0.9%	0.6%	0.4%	0.3%	0.3%
qnxadas-junction-approach	6.3%	1.7%	1.9%	2.8%	4.6%	2.8%	1.8%	1.1%	1.0%	0.9%	0.8%	0.8%
qnxadas-junction-exit	8.7%	3.8%	2.6%	3.2%	3.8%	3.3%	2.3%	1.2%	1.4%	1.6%	1.2%	1.3%
qnxadas-motorway-join	7.6%	4.6%	3.1%	3.2%	3.3%	3.4%	1.9%	1.8%	1.5%	1.4%	1.2%	2.4%
qnxadas-navigating-bends	5.6%	2.3%	1.8%	2.4%	3.4%	2.2%	2.1%	1.6%	1.2%	1.2%	1.2%	1.0%
Overall average	6.6%	3.0%	3.7%	3.3%	2.6%	3.0%	1.7%	1.2%	1.0%	0.9%	0.8%	0.9%

TABLE III
OVERALL AVERAGE OF BD RATE AND COMPLEXITY.

Method	Geom.BD-TotalRate		Attr.BD-TotalRate	Rate of
	D1	D2	Reflectance	Enc time
Li et al. [8]	1.8%	1.8%	-21.9%	94.2%
Prop.1	-0.8%	-0.8%	-3.2%	97.5%
Prop.2	-0.8%	-0.8%	-3.0%	98.1%
Prop.3	-0.8%	-0.8%	-3.0%	98.2%

TABLE IV
COMPARISON OF BD RATE BETWEEN G-PCC ANCHOR AND THE PROP.3.

Point Cloud	Geom.BD-TotalRate		Attr.BD-TotalRate
	D1	D2	Reflectance
ford_01_q1mm	-0.5%	-0.5%	-1.6%
ford_02_q1mm	-0.7%	-0.8%	-1.6%
ford_03_q1mm	-0.3%	-0.4%	-2.4%
qnxadas-junction-approach	-0.1%	0.0%	-2.0%
qnxadas-junction-exit	0.0%	0.1%	-1.7%
qnxadas-motorway-join	-3.9%	-3.9%	-9.9%
qnxadas-navigating-bends	-0.1%	-0.1%	-1.8%
Overall average	-0.8%	-0.8%	-3.0%

shown in Fig. 3, the curves for G-PCC Anchor and Prop.3 intersect at certain points, which may imply that the BD-rate calculation is not accurate [20]. However, the primary goal is the improvement of the error rate, which has been achieved.

V. CONCLUSION

In this paper, we propose a method for rate control based on G-PCC that updates the R-Q model parameters for each frame using temporal correlations, specifically by employing LMS and dynamically controlling its update step size. In addition, for optimal bit allocation for each frame, we propose a bit allocation method based on the R-Q model that is updated for each frame. Experimental results show an improvement of 99.7% in error rate compared to conventional methods.

ACKNOWLEDGMENT

These research results were obtained from the commissioned research (No. JPJ012368C06801) by National Institute of Information and Communications Technology (NICT), Japan.

REFERENCES

[1] "Transmission experiment using real-time codec compliant with the latest international standard of point cloud compression," https://www.kddi-research.jp/english/newsrelease/2023/012401.html, 2023.

[2] L. Huang, S. Wang, K. Wong, J. Liu, and R. Urtasun, "Octsqueeze: Octree-structured entropy model for lidar compression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1313–1323.

[3] Z. Que, G. Lu, and D. Xu, "Voxelcontext-net: An octree based framework for point cloud compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6042–6051.

[4] C. Fu, G. Li, R. Song, W. Gao, and S. Liu, "Octattention: Octree-based large-scale contexts model for point cloud compression," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 1, 2022, pp. 625–633.

[5] ISO/IEC, "ISO/IEC 23090-9:2023 information technology — coded representation of immersive media — part 9: Geometry-based point cloud compression."

[6] Geometry-based point cloud compression reference software, tmc13-12.10. [Online]. Available: https://github.com/MPEGGroup/mpeg-pcc-tmc13

[7] C. Cao, M. Preda, V. Zakharchenko, E. S. Jang, and T. Zaharia, "Compression of sparse and dense dynamic point clouds—methods and standards," *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1537–1558, 2021.

[8] L. Li, Z. Li, S. Liu, and H. Li, "Frame-level rate control for geometry-based lidar point cloud compression," *IEEE Transactions on Multimedia*, vol. 25, pp. 3855–3867, 2022.

[9] B. Widrow and M. E. Hoff, "Adaptive switching circuits," in *1960 IRE WESCON Convention Record, Part 4. New York, NY, USA: IRE*, 1960, pp. 96–104.

[10] ISO/IEC, "ISO/IEC 23090-5:2023 information technology — coded representation of immersive media — part 5: Visual volumetric video-based coding (v3c) and video-based point cloud compression (v-pcc)."

[11] Q. Liu, H. Yuan, R. Hamzaoui, and H. Su, "Coarse to fine rate control for region-based 3d point cloud compression," in *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2020, pp. 1–6.

[12] L. Li, Z. Li, S. Liu, and H. Li, "Rate control for video-based point cloud compression," *IEEE Transactions on Image Processing*, vol. 29, pp. 6237–6250, 2020.

[13] Q. Liu, H. Yuan, J. Hou, R. Hamzaoui, and H. Su, "Model-based joint bit allocation between geometry and color for video-based 3d point cloud compression," *IEEE Transactions on Multimedia*, vol. 23, pp. 3278–3291, 2020.

[14] Y. Wang, W. Gao, X. Mu, and H. Yuan, "Rate control optimization for joint geometry and attribute coding of lidar point clouds," in *2023 IEEE International Conference on Visual Communications and Image Processing (VCIP)*. IEEE, 2023, pp. 1–5.

[15] G. Li, W. Gao, and W. Gao, "AVS point cloud compression standard," in *Point Cloud Compression: Technologies and Standardization*. Springer, 2024, pp. 167–197.

[16] J. Zhang, J. Zhang, W. Ma, D. Ding, and Z. Ma, "Content-aware rate control for geometry-based point cloud compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 10, pp. 9550–9561, 2024.

[17] L. Hou, L. Gao, Q. Zhang, Y. Xu, J.-N. Hwang, and D. Wang, "Rate control for geometry-based lidar point cloud compression via multi-factor modeling," *IEEE Transactions on Broadcasting*, 2024.

[18] "Common Test Conditions for G-PCC," ISO/IEC JTC1/SC29/WG7, 147th MPEG meeting, Sapporo, Tech. Rep. N00944/MDS24178, July 2024.

[19] M. Tang, J. Wen, and Y. Han, "A generalized rate-distortion-λmodel based hevc rate control algorithm," *arXiv preprint arXiv:1911.00639*, 2019.

[20] A. Kenneth, B. Frank, O. Jens-Rainer, S. Andrew, S. Rickard, S. Jacob, S. Gary, and T. Alexis, "Summary information on bd-rate experiment evaluation practices," joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 output document JVET-R2016, 18th JVET meeting by teleconference, April 2020.