APSIPA Transactions on Signal and Information Processing, 2022, 11, e23

Original Paper

Content-Adaptive Level of Detail for Lossless Point Cloud Compression

Lei Wei[1*], Shuai Wan[1,2], Fuzheng Yang[3] and Zhecheng Wang[1]

[1]*School of Electronics and Information, Northwestern Polytechnical University, China*
[2]*School of Engineering, Royal Melbourne Institute of Technology, Australia*
[3]*School of Telecommunications Engineering, Xidian University, China*

ABSTRACT

The nonuniform distribution of points in a point cloud and their abundant attribute information (such as colour, reflectance, and normal) result in the generation of massive data, making point cloud compression (PCC) essential for related applications. The hierarchical structure of the level of detail (LOD) in a point cloud and the corresponding predictions are commonly used in PCC, whereas the current method of LOD generation is neither content adaptive nor optimized. Targeting lossless PCC, an LOD prediction error model is proposed in this work, based on which the prediction error is minimized to obtain the optimal coding performance. As a result, the process of generating LOD is optimized, where the smallest number of LOD levels that yields the minimum attribute bitrate can be found. The proposed method is evaluated on various standard datasets under common test conditions. Experimental results show that the proposed method achieves optimal coding performance in a content-adaptive way while significantly reducing the time required for encoding and decoding, i.e., ∼15.2% and ∼17.3% time savings on average for distance-based LOD, and ∼5.4% and ∼5.1% time savings for Morton-based LOD, respectively.

*Corresponding author: Lei Wei, l.wei@mail.nwpu.edu.cn.

Received 23 January 2022; Revised 29 April 2022
ISSN 2048-7703; DOI 10.1561/116.00000004

Keywords: Point cloud compression, level of detail, prediction error, bitrate.

JEL Codes: L82

1 Introduction

In recent years, the rapid development of 3D acquisition technologies has boosted the applications of point clouds, which have been widely employed in virtual reality, automatic driving, digital cities, etc. [23]. However, the large amount of data contained in a point cloud poses storage and transmission challenges. Therefore, point cloud compression (PCC) has attracted significant attention in both research and industry. How to explore the potential correlations among unstructured points remains challenging.

In current PCC approaches, such as the geometry-based point cloud compression (G-PCC) standard developed by the Moving Pictures Experts Group (MPEG), the geometry and attribute of a point cloud are separately compressed, the encoding is a sequential operation, which first processes geometry and then the attribute [13, 14, 21]. The attribute is predicted and encoded with the help of the reconstructed geometry information.

With regard to attribute compression, prediction and transform coding have been proposed to improve the accuracy of attribute reconstruction. For example, Gu *et al.* [17] proposed a graph-based 3D Point Cloud (3DPC) colour prediction method and treat the graph construction as the sparse optimization problem, where point cloud geometry information is used to guide the generation of sparse representation basis [18]. Zhang *et al.* [30] developed a graph transform (GT) method for encoding the attribute blocks of a point cloud. This approach requires that the point cloud be captured or arranged on regular grids, and the nearest neighbour points are used for predictive coding, followed by a discrete cosine transform (DCT) of the prediction error. In [5], a point cloud was sampled into a unified grid, and then the mesh was divided into blocks to directly apply a graph transform, followed by 3D block prediction. Cohen *et al.* [4] proposed a GT-based attribute compression approach that extends the k-nearest neighbours (KNN) method to generate increasingly efficient graphs, and this technique has a certain effect on reducing the number of isolated subgraphs. Shao *et al.* [26] proposed a method based on the k-d tree structure to make the dimensions of each subgraph consistent, thus avoiding the existence of isolated subgraphs. Hou [19] proposed to increase the regularity of 3DPC by reordering the geometry of 3DPC. De Queiroz and Chou [9] proposed a colour compression method for point clouds based on the hierarchical transform and arithmetic coding. The hierarchical transform is a levelled sub-band transform (similar to the adaptive change of the Haar wavelet) and has been applied to G-PCC. Another essential hierarchical solution in the field of PCC is level of detail (LOD).

In this paper, by establishing a mathematical prediction error model for LOD levels (LODs), an optimized solution approach for determining the number of LODs in a point cloud is proposed, and this leads to content-adaptive LOD generation for a given point cloud. The proposed work is well adapted to the different contents of various point clouds and achieves optimal performance while reducing the induced encoding and decoding complexity by avoiding unnecessary LODs.

The rest of this paper is organized as follows: Section 2 presents the related works. Section 3 describes the LOD in G-PCC. Section 4 analyses the relationship between the prediction error and attribute bitrate of a model and describes the proposed method for estimating the optimal number of LODs by establishing a mathematical prediction error model. Section 5 provides experimental results and analysis. The conclusion is drawn in Section 6, followed by references.

2 Related Works

LOD, which was first proposed by Clark in 1976 [3], has been commonly used in real-time 3D technologies in computer graphics. A discussion on various algorithms, problems, and solutions in the field of LOD can be found in [22, 24], focusing on meshes and rendering.

An LOD structure actually stores a set of copies of the original model at different resolutions, all of which have strong spatial correlations, as shown in Figure 1. Figure 1(a) is the original point cloud, and the remaining subfigures from left to right are the sampled point clouds showing different detail levels or resolutions (i.e., different LODs), where the sampling distance increases from Figure 1(b) to (d).

As a common solution for spatial division, LOD is widely employed to improve coding performance. It is noted that LOD not only contributes to attribute compression but also provides potential scalability. In G-PCC, an LOD generation method based on distance is used for the dense point clouds, and the coded nearest neighbour points are used for prediction [2, 25]. Kathariya *et al.* [20] proposed an LOD generation scheme based on a binary tree, this approach is useful for sparse point clouds, such as those acquired by light detection and ranging (LiDAR). In [8], by comparing the LOD generation methods of random sampling and grid sampling, an extended method for generating LOD by constraining the samples to regular discrete grids was proposed to optimize the coverage of the volume represented by each sample. Fan *et al.* [12] proposed an LOD generation method based on clustering, and this technique has both general topological applicability and good rate-distortion performance at low bitrates.

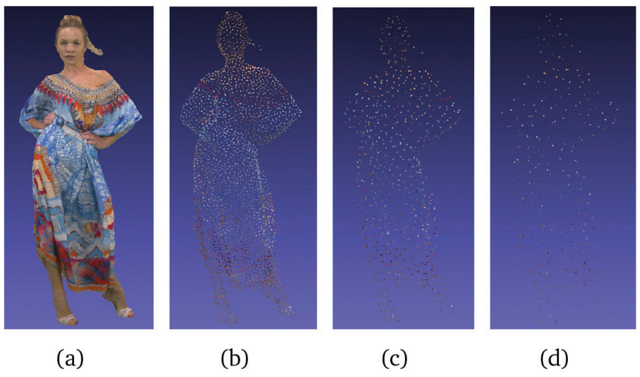

Figure 1: Different LODs for point cloud *longdress_vox10*. (a) Original point cloud; (b) the second LOD; (c) the third LOD; (d) the sixth LOD.

Generally, the influencing factors of an LOD generation algorithm include the number of LODs (denoted as N_LOD), the initial sampling distance, and the scaling factor used to derive the sampling distance for the LODs [11]. The existing LOD generation schemes have a common problem in that the above parameters are not adaptive to point clouds with different contents. The key factor of these methods is the number of LODs. Taking G-PCC as an example, how to set N_LOD remains an open question. In common test conditions (CTC), a large N_LOD is used for all point clouds [6]. It is, however, intuitive that different N_LOD should be applied for point clouds with different characteristics. If N_LOD is not properly selected, it is very likely that the resulting coding performance will not be optimized. On the other hand, if N_LOD is set too large, which may result in optimal coding performance, unnecessary LODs will be generated, increasing the time complexity without contributing to the coding performance.

Targeting the above problem, we propose a method for estimating the optimal N_LOD by minimizing the prediction error, which results in a minimized attribute bitrate for the lossless compression of a given point cloud. It is noted that the analysis in this paper is carried out based on the LOD generation strategy in G-PCC. Similarly, the proposed method can be extended to other LOD generation schemes in a straightforward way.

3 LOD in G-PCC

The LOD generation process is generally carried out in the following way. A point cloud is reorganized into a set of refinement levels $\{A_l\}_{(l=1,\ldots,N-1)}$ and a detail level L_N according to a set of distances $\{d_l\}_{(l=1,\ldots,N-1)}$ specified by the user. Note that the distances should satisfy $d_l < d_{l+1}$ [13, 14].

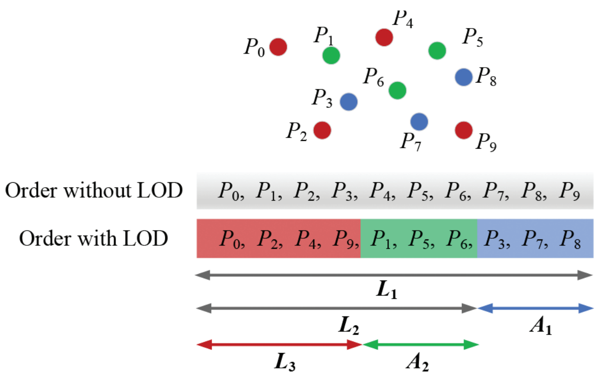
P_0
P_1
P_4
P_5
P_6
P_8
P_3
P_2
P_7
P_9
Order without LOD
P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9
Order with LOD
P_0, P_2, P_4, P_9, P_1, P_5, P_6, P_3, P_7, P_8
L_1
L_2
A_1
L_3
A_2

Figure 2: Level of detail generation process.

The process of LOD generation is carried out in an iterative way, as illustrated in Figure 2. Generally, the original point cloud can be regarded as L_1. By applying the initial sampling distance, the detail level L_2 and refinement level A_1 are obtained. Then, L_3 and A_2 are generated by sampling L_2. As a result, after two sampling cycles, the point cloud is decomposed into refinement levels A_1 and A_2 and a detail level L_3, based on which the point cloud is reorganized into $\{L_3, A_2, A_1\}$.

The LOD could reduce the correlations among the points in the point cloud and helps remove redundancy by prediction through the decomposition of the point cloud into several levels. At the same time, the hierarchical nature of LOD makes it naturally suitable for the spatially scalable coding of the point clouds, in which the stored copies vary from coarse to fine and can be transmitted or decoded according to the different needs of users.

As shown in Figure 2, after reorganizing the point cloud into LODs, the points are actually reordered in the LOD order. To describe the LOD-based coding process more clearly, we represent it as a pyramid structure in Figure 3,

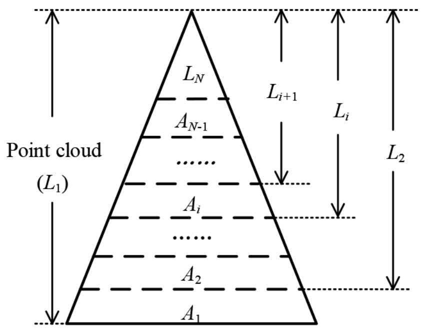

Figure 3: LOD pyramid structure.

where the points in the point cloud are sorted from top to bottom according to the LOD pyramid. By accumulating the detail levels and the corresponding refinement levels, the resulting LOD structure with different levels of detail can be applied to efficiently code point clouds while enabling spatial scalability. For example, L_{i+1} is the $i+1$th detail level, and the combination of L_{i+1} and A_i results in a denser detail level L_i.

Similar to predictive video coding, a point cloud uses prediction to remove the correlations among the points and applies a specific predictive method to exploit the corresponding redundancy. Predictive coding refers to the use of one or several coded samples, the prediction of the current sample value according to a prediction model or method, and the coding of the difference between the real value and the predicted value of a given sample to effectively improve the coding performance of the associated method [28].

In PCC, the common prediction method is to find several nearest coded points according to their Euclidean distances and to use the weighted average of their reconstructed values as the predicted value of the current point. G-PCC takes the same approach by finding the k nearest coded points. Because the computation of Euclidean distances introduces a large amount of complexity, Wei *et al.* [29] proposed a KNN search method that finds the reference points more quickly and more accurately by using the Morton code.

Due to the LOD structure, compared with the general prediction method, the prediction approach based on LOD has a different range of available reference points at each level. Regardless of whether a given point is at a detail level or refinement level, the prediction is carried out in the same way, where the difference lies in the available reference points. That is, after LOD generation, coding is carried out in LOD order, i.e., the points in the topmost detail level are coded first, followed by those in sequential refinement levels. When coding the points in the topmost detail level, only the coded points in this level are available for prediction, while the points in a refinement level can refer to the coded points in the previous detail level as well as those in the current refinement level.

Figure 4 illustrates the prediction process conducted with LOD by using the points in Figure 2 as an example. Specifically, the points in L_3 are coded first (the red points), followed by those in A_2 (the green points), and finally those in A_1 (the blue points). The points in L_3 can only use the coded points in this level as reference points, while the points in A_2 and A_1 have more candidates, i.e., the points in the previously coded LODs and the coded points in the current refinement level. For P_3 and P_6 in a refinement level, the reference points can be selected from a wider range and in an omnidirectional way. However, the prediction of P_4 can only refer to the coded points in L_3, and the prediction process is limited to the order determined by the coding order, i.e., with directionality. Apparently, the prediction of P_3 and P_6 tends to be more accurate than that of P_4.

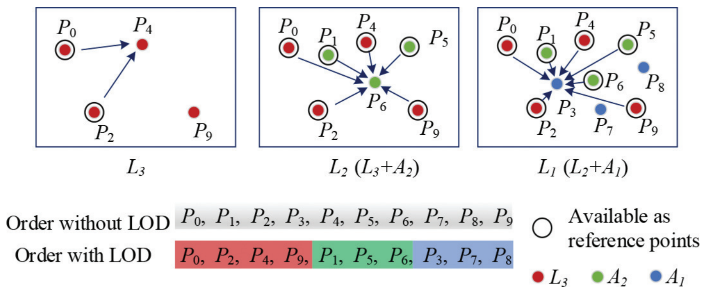
P_0
P_4
P_2
P_9
L_3
P_0
P_1
P_4
P_5
P_6
P_2
P_9
L_2 (L_3+A_2)
P_0
P_1
P_4
P_5
P_6
P_8
P_3
P_2
P_7
P_9
L_1 (L_2+A_1)
Order without LOD P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9
Order with LOD P_0, P_2, P_4, P_9, P_1, P_5, P_6, P_3, P_7, P_8
Available as reference points
L_3
A_2
A_1

Figure 4: Reference points for prediction in several LODs.

To facilitate the discussion in this manuscript, according to the different types of levels where the points to be predicted are located, prediction refers to prediction at the topmost detail level and prediction at the refinement level.

In the coding process, the reference points can only be selected from the coded points, so the impact brought by LOD is twofold. Taking the first step in the LOD generation process as an example, when the point cloud is sampled once, the detail level L_2 and the refinement level A_1 are obtained. The points in L_2 are coded first, but the process suffers from certain directionality when selecting reference points while limited by the coding order. The distance between adjacent points in L_2 is larger than the original distance (due to the increased LOD); this generally weakens the correlations between points and causes performance loss. Meanwhile, the available reference points for the prediction of the points in A_1 are located in various spatial directions (since all the points in L_2 would have already been coded and available for reference), which generally improves the accuracy of prediction and results in a performance gain. When the gain is greater than the loss, the LOD may improve the coding performance.

Afterward, L_2 can be further sampled to reduce the loss and obtain a new refinement level A_2 and a new detail level L_3; this is done in a similar manner as above. The entire LOD generation is performed iteratively, where L_N and A_{N-1} are obtained after the N-1th sampling cycle. Suppose that at this time, the overall performance gain in A_{N-1} is less than or equal to the overall performance loss in L_N; further LOD generation will not result in further gains to the coding performance, so the decomposition process can be stopped. Therefore, the overall performance of LOD generation is a trade-off.

It is seen from the above discussion that given a point cloud, different LOD structures lead to different prediction errors, as well as different coding efficiencies. It is noted that the performance gain and loss brought by LOD reach a balance when a certain number of LODs is applied. The distance between adjacent points in the remaining part (i.e., the detail level) is very large, so the correlation is very weak. Therefore, with further LOD generation, the impact is very limited, and at the same time, the number of remaining points is very small (declining in an exponential way with further LOD generation), which makes the impact of these points negligible. Hence, the coding performance remains almost the same as that when the number of LODs is optimal. The contributions of more LODs become rather slim or even cause performance loss in some cases. Additionally, unnecessary LODs increase the time complexity. It is therefore not appropriate to apply a large number of LODs.

In addition, since the distributions and characteristics of different point clouds are different, different numbers of LODs should be applied. How to generate the proper number of LODs for a specific point cloud is therefore essential for optimizing the coding performance of PCC.

4 Content-adaptive LOD with Optimization

To determine the optimal number of LODs for a point cloud, it is necessary to model the attribute bitrate of each level (including the refinement level and the detail level). However, LODs are not coded separately but together. It is cumbersome to obtain the corresponding independent bitrate of each level, so the prediction error is used instead.

4.1 The Relationship between the Prediction Error and Bitrate

In this subsection, we analyse the distribution of the prediction error of the model to determine the relationship between the error and the corresponding attribute bitrate, which will be verified through experiments. Since distance-based LOD is useful for dense point clouds, a few point clouds from G-PCC test class A are selected as the dataset for verifying the theoretical analysis. These point clouds are from the Static Objects and Scenes database [6, 7] and are named $redandblack\_vox10\_1550$, $soldier\_vox10\_0690$, $frog\_00067\_vox12$ and $facade\_00015\_vox14$. It is noted that as a common sense-based process, distance-based LOD is not suitable for sparse point clouds, such as those acquired by LiDAR. The reason lies in the fact that for sparse point clouds, the distances among points are generally very large and the correlations are very low. How to optimize the LOD generation for sparse point clouds remains an open question.

To investigate the distribution of the prediction error, experiments are carried out with N_LOD set to a certain value (i.e., 10 in this work), and the abovementioned point clouds are compressed using G-PCC for lossless compression. The other parameters are configured following the configuration scripts [6, 16].

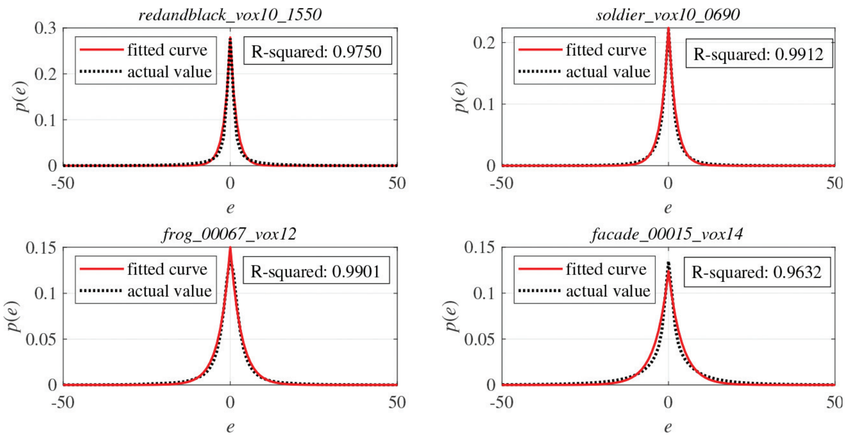

Figure 5: Distributions of the prediction error e.

The prediction error denoted as e is defined as follows,

$$e = a_i - \hat{a}_i,$$

where a_i is the original attribute value of point i, and $\hat{a}_i$ is the corresponding predicted attribute value. Statistics on the prediction errors of all points of the point cloud are collected to obtain its probability density function $p(e)$, where the results obtained for the point clouds in the analysis dataset are shown in Figure 5.

It can be observed from Figure 5 that the prediction errors follow the Laplacian distribution. The probability density function of the Laplacian distribution is

$$p(e) = \frac{1}{2\lambda}\exp(-\frac{|e|}{\lambda}), \tag{1}$$

where λ is the parameter of the distribution. The goodness of fit is measured by the coefficient of determination, i.e., R-squared. It is a statistical measure in regression models that determines the proportion of variance in the dependent variable that can be explained by the independent variables. In other words, R-squared shows how well the data fit the regression model. The maximum value of R-squared is 1. The closer the value of R-squared is to 1, the better the fit is; conversely, the smaller the value of R-squared, the worse the fit is. It is obvious that the distribution of prediction error fits the Laplacian distribution well.

Since the mean absolute error (MAE) metric is able to avoid the problem in which the errors offset each other, it is used to measure prediction error in this work. According to Equation (1), the entropy $H(e)$ is obtained as follows,

$$H(e) = \frac{E_e}{\lambda} + log(2\lambda), \tag{2}$$

where E_e is the MAE of the prediction error and λ is the distribution parameter of the prediction error.

Regarding LOD generation, different settings of the number of LODs will lead to different prediction errors, in which E_e and λ are different. On the basis of the maximum entropy principle, the prediction error satisfies $E_e = \lambda$, and $H(e)$ is maximal when the distribution is a Laplacian distribution. Accordingly, Equation (2) can be rewritten as,

$$MAX\{H(e)\} = 1 + log(2E_e). \tag{3}$$

It is observed from Equation (3) that E_e and $H(e)$ increase monotonically. It is therefore reasonable to convert the problem of the minimization of the entropy or bitrate to the minimization of E_e.

With N_LOD set to different values (i.e., from 1 to 5), the point clouds in the analysis dataset are lossless compressed. The prediction errors and the corresponding attribute bitrates are plotted in Figure 6, which shows the relationship between E_e and the attribute bitrate. It is concluded that the minimization of E_e is feasible for minimizing the attribute bitrate and is consistent with the above theoretical and empirical analysis. The curves of the actual data may not be strictly logarithmic since the maximum entropy is an ideal hypothesis, whereas the overall trend is consistent.

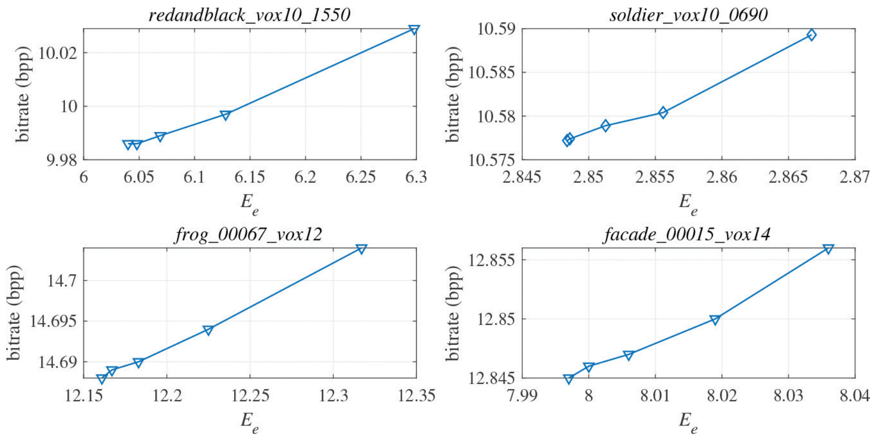

Figure 6: The relationship between E_e and the attribute bitrate.

In this paper, we propose a method for minimizing the MAE of the prediction error to minimize the attribute bitrate for lossless compression, and an LOD prediction error model is built to obtain the optimal value of N_LOD.

4.2 LOD Prediction Error Model

A point cloud is reorganized into a series of LODs $\{A_1, A_2, \cdots, A_{N-1}, L_N\}$ when N_LOD equals N. The MAE of the overall prediction error of the point cloud is denoted as E_{total}, which is the weighted sum of the MAE of all levels,

$$E_{total} = \sum_{i=1}^{N-1} s_{1,i} E_{A_i} + s_{2,N} E_{L_N}, \tag{4}$$

where the MAE of $\{A_i\}_{(i=1,\cdots,N-1)}$ is denoted as $\{E_{A_i}\}_{(i=1,\cdots,N-1)}$ and that of L_N is denoted as E_{L_N}, $s_{1,i}$ is the ratio of the number of points in A_i to the total number of points in the point cloud, and $s_{2,N}$ is the ratio of the number of points in L_N to the total number of points in the point cloud. Specially, when N equals 1, $E_{total} = E_{L_1}$. That is, L_1 is the point cloud itself.

During LOD generation, let r_i be the ratio of the number of points in A_i (denoted as $Num(A_i)$) to the number of points in detail level L_i (denoted as $Num(L_i)$), where this ratio is represented as $r_i = Num(A_i)/Num(L_i)$.

For the refinement level A_i, we have

$$s_{1,i} = \begin{cases} r_1, & \text{if } i = 1 \\ \prod_{k=1}^{i-1} (1 - r_k) r_i, & \text{if } i \geq 2 \end{cases}. \tag{5}$$

For the detail level L_i, we have

$$s_{2,i} = \begin{cases} 1, & \text{if } i = 1 \\ \prod_{k=1}^{i-1} (1 - r_k), & \text{if } i \geq 2 \end{cases}. \tag{6}$$

Equation (4) is then rewritten as follows,

$$E_{total} = r_1 E_{A_1} + \sum_{i=2}^{N-1} (\prod_{j=1}^{i-1} (1 - r_j) r_i E_{A_i}) + \prod_{k=1}^{N-1} (1 - r_k) E_{L_N}. \tag{7}$$

When the points are dense, they can be regarded as uniformly distributed in a small region. In the following analysis, it is assumed that the points are uniformly distributed. The ratio r_i is related to the scaling factor used for the derivation of the sampling distance for LOD generation. For a given set of points in space, in short, the sampling process extracts a point every a few points to obtain a new set. Thus, the minimum sampling period involves sampling once every other point, and the corresponding minimum sampling distance is the minimum distance between the current point and other points except the nearest adjacent points of the current point. Therefore, denoting

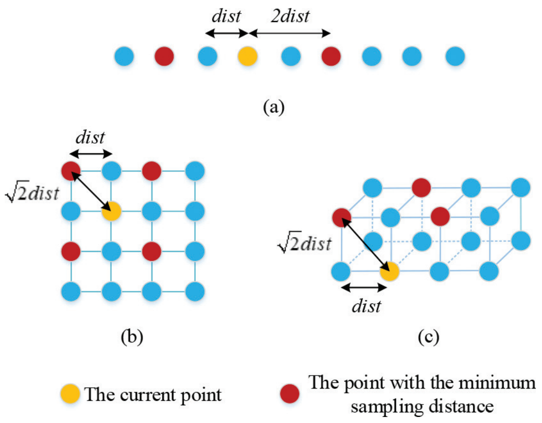

Figure 7: Minimum sampling distance: (a) one-dimensional, (b) two-dimensional, and (c) three-dimensional.

the minimum distance between the adjacent points as *dist*, the minimum sampling distance is generally $2dist$ for the one-dimensional case and $\sqrt{2}dist$ for the two-dimensional and three-dimensional cases, as shown in Figure 7.

During LOD generation, multiple sampling processes are carried out. A series of sampling distances are obtained by iteratively multiplying the initial sampling distance by the scaling factor. The greater the scaling factor is, the larger the subsequent sampling distance will be, and thus the sparser the sampling output will be, resulting in a decrease in prediction accuracy. Therefore, the minimum sampling distance should be used in each sampling process; that is, the minimum scaling factor should be selected every time [1]. Furthermore, for the sake of implementation simplicity and hardware usability, the scaling factor should be 2, that is, the minimum integer value that is larger than $\sqrt{2}$.

As in G-PCC, the scaling factor for the derivation of the sampling distance for LOD generation is set to 2 in this work [14]. In Figure 8, all points on the left is L_i, the distance between the adjacent points is the sampling distance of the lower detail level L_{i-1}, which is denoted as $dist_{i-1}$, and the upper detail

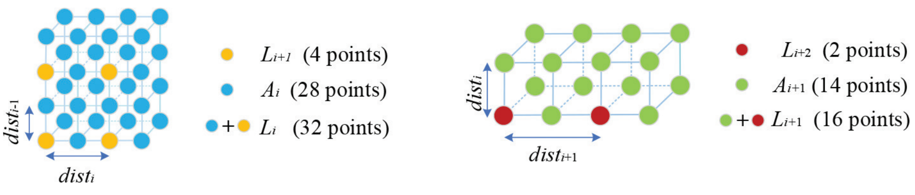

Figure 8: LOD generation through sampling ($p = 7/8$).

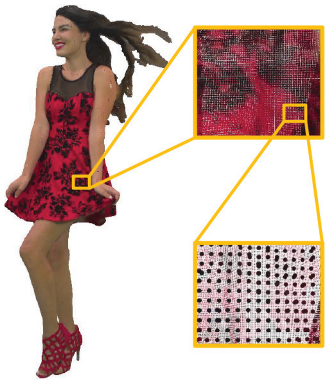

Figure 9: Points in a small region of the surface in the point cloud.

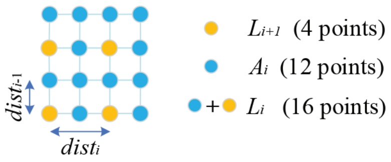

Figure 10: LOD generation through sampling (2D) ($p = 3/4$).

level L_{i+1} is obtained by sampling L_i by $dist_i = 2dist_{i-1}$, appearing as the set of yellow points; the remaining blue points belong to A_i. On the right of the figure, L_{i+1} includes all points, which is then sampled by $dist_{i+1} = 2dist_i$ to obtain L_{i+2}, shown as the set of red points; the remaining green points belong to A_{i+1}. When i is larger than 1, the ratio r_i of the adjacent LODs is a constant and denoted as p (with p=7/8). When i equals 1, the ratio r_1 is only related to the initial sampling distance (denoted as D_0 below).

In practice, all points are located on the surface of the object. The points within a small region can be regarded as a plane, as shown in Figure 9. In this instance, p equals $3/4$, as shown in Figure 10.

Before LOD generation in G-PCC, it is necessary to set the initial sampling distance D_0 for each point cloud, and r_1 is obtained from D_0. Then, r_1 is considered a known quantity. By substituting $r_i = p$ ($i > 1$) into Equation (7), E_{total} is expressed as a function of N_LOD (i.e., N),

$$E_{total}(N) = r_1 E_{A_1} + \sum_{i=2}^{N-1} p(1-r_1)(1-p)^{i-2} E_{A_i} + (1-r_1)(1-p)^{N-2} E_{L_N}. \quad (8)$$

The MAE of the prediction error at each level and r_1 are obtained from experiments are substituted into Equation (8) to obtain the estimated E_{total},

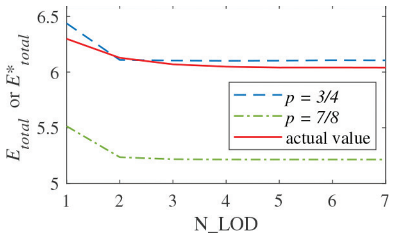

Figure 11: Comparison of E_{total} obtained with different values of p and the actual E^*_{total} of *redandblack_ vox10_ 1550.*

with p set to 7/8 and 3/4 under different values of N_LOD. E_{total} obtained using Equation (8) is compared with the actual MAE of the overall prediction error (denoted as E^*_{total}) obtained when the point cloud is decomposed using different N_LOD. The results of the point cloud *redandblack_ vox10_ 1550* are shown in Figure 11, and similar curves can be observed for other point clouds. When p is 3/4, E_{total} is more in line with the actual situation, which is consistent with the analysis above. Therefore, the value of p should be 3/4 for dense point clouds, and we model the prediction error using Equation (8) with $p = 3/4$.

4.3 The Optimized N_LOD

To derive the optimal value of N_LOD, its relationship with the prediction error is needed. The spatial correlation in the point cloud is related to the distances between pairs of adjacent points, and generally speaking, the smaller the distance is, the stronger the correlation is [27], which is removed by prediction. On one hand, for PCC without LOD, the prediction error is related to the distances between adjacent points. On the other hand, for LOD-based PCC, the prediction error is related to the sampling distance of each detail level. Therefore, the prediction error and the corresponding sampling distance are modelled in this subsection.

In Figure 12, (a) shows the relationship between the prediction error of the topmost detail level and the distance between adjacent points, and (b) shows the relationship between the prediction errors of the refinement level and the corresponding sampling distance. All the data are obtained from the experiment in Section 4.1. The triangle and circle markers are the actual values, and the lines are curves that are fitted according to these markers. With increasing distance, E_e also increases, but the slopes of the curves decrease. It is noted that the conclusion holds for all point clouds in the analysis dataset and is not limited to those shown in Figure 12.

In LOD-based predictive coding, according to the different types of LODs where the points to be predicted are located, each prediction is referred to as a

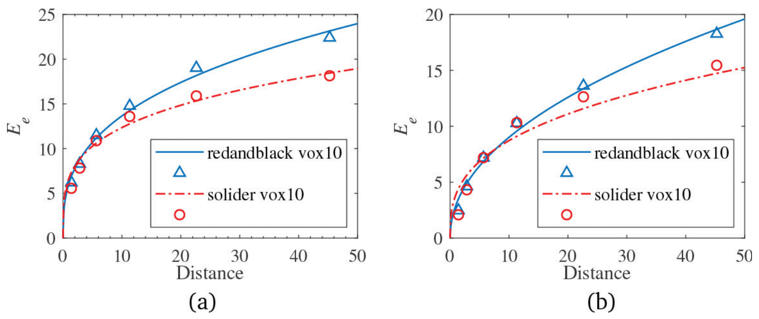

Figure 12: The relationship between the prediction error and distance: (a) the topmost detail level; (b) the refinement level.

prediction at the topmost detail level and a prediction at the refinement level. These two situations are analysed. First, for the prediction at the topmost detail level, assuming that the j-th detail level L_j is the topmost detail level, let $E_{L_j} = f(d_{1,j})_{(j=1,2,......,N)}$, where $d_{1,j}$ is the distance between the adjacent points in L_j and E_{L_j} is the MAE of the prediction error of L_j. Then, for prediction at the refinement level, let $E_{A_i} = g(d_{2,i})_{(i=1,2,......,N)}$, where $d_{2,i}$ is the sampling distance of the i-th refinement level A_i and E_{A_i} is the MAE of the prediction error of A_i. Finally, Equation (8) is further rewritten as follows,

$$\begin{aligned} E_{total}(N) = r_1 g(d_{2,1}) + \sum_{i=2}^{N-1} p(1-r_1)(1-p)^{i-2} g(d_{2,i}) \\ + (1-r_1)(1-p)^{N-2} f(d_{1,N}), \end{aligned} \tag{9}$$

where $d_{2,1}$ is the initial sampling distance D_0, $d_{2,i}$ is the sampling distance of A_i and L_{i+1} ($d_{2,i} = 2^{i-1}D_0$), and $d_{1,N}$ is the distance between adjacent points in L_N, with $d_{1,N} = d_{2,N-1}$.

After establishing the prediction error model, the functions $f(d_{1,j})$ and $g(d_{2,i})$ are fitted, as shown in Figure 12. They are expressed as the power functions $f(d_{1,j}) = a_1 d_{1,j}^{b_1}$ and $g(d_{2,i}) = a_2 d_{2,i}^{b_2}$. Then, we can obtain Equation (10).

$$\begin{aligned} E_{total}(N) = r_1 a_2 D_0^{b_2} + a_1(1-r_1)(1-p)^{N-2}(2^{N-2}D_0)^{b_1} \\ + \frac{(p(1-r_1)a_2 D_0^{b_2})((1-p)2^{b_2} - ((1-p)2^{b_1})^{N-1})}{(1-p)(1-(1-p)2^{b_2})}. \end{aligned} \tag{10}$$

From the perspective of spatial correlation, the distance between adjacent points in each detail level becomes larger along with the generation of LODs, resulting in a decrease in the corresponding spatial correlation. As a result, the performance loss increases, gradually mitigating the performance gain of

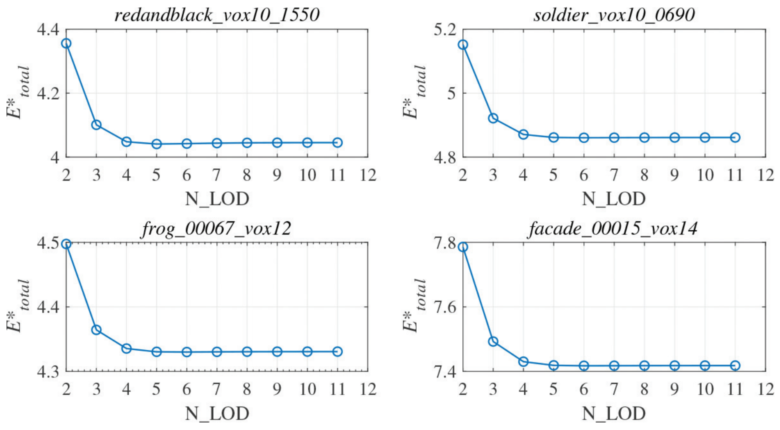

Figure 13: The relationship between E^*_{total} and N_LOD.

the corresponding refinement level, and eventually becomes almost equal to or even slightly larger than the gain.

At the same time, the number of remaining points (i.e., in the new detail level) is very small, declining in an exponential manner with further LOD generation, and the gap between the performance gain and loss is very limited, which makes the impact of these points negligible. Therefore, in the later stages of LOD generation, the coding performance remains almost unchanged. E^*_{total} obtained from the experiment in Section 4.1 is shown in Figure 13, and the curves illustrate the described trend well.

It can be seen that E^*_{total} is generally a convex function of N_LOD. With LOD generation, the coding performances are all improved in Figure 13.

However, there could also be other possibilities (highly unlikely to occur) in which the coding performance remains unchanged or even becomes worse after LOD generation. Especially for the latter, the trend of coding performance is opposite to the general situation, that is, E^*_{total} gradually increases with the increase in the number of LODs.

It is therefore necessary to see whether the derivative of the prediction error in Equation (10) is always approximately 0 to judge whether the coding performance changes after LOD generation. The partial derivative of $E_{total}(N)$ with respect to N (i.e., $\partial E_{total}(N)/\partial N$) is shown as follows,

$$\begin{aligned}\frac{\partial E_{total}(N)}{\partial N} = & -\frac{1}{1-p}\frac{(1-r_1)a_2D_0^{b_2}}{1-(1-p)2^{b_2}}((1-p)2^{b_2})^{N-1}\ln((1-p)2^{b_2}) \\ & + a_1D_0^{b_1}(1-r_1)((1-p)2^{b_1})^{N-2}\ln((1-p)2^{b_1}). \end{aligned} \tag{11}$$

If $\partial E_{total}(N)/\partial N$ is 0, the LOD-based coding will not change the performance, and the optimal N_LOD should be 1. If not 0, the concavity or convexity of $E_{total}(N)$ should be judged by calculating the second derivative (i.e., $\partial^2 E_{total}(N)/\partial N^2$). It is convex when the second derivative is greater than 0 and concave when it is less than 0.

For a concave $E_{total}(N)$, the minimum value of E_{total} is yielded when N_LOD = 1, i.e., the point cloud is best compressed without LOD generation; thus, the optimal N_LOD is 1.

For a convex $E_{total}(N)$, when the point cloud is decomposed into a certain number of LODs, the bitrate reaches its minimum. Continuing LOD decomposition beyond this point is not helpful for coding but increases the time complexity of the process. By calculating $\partial E_{total}(N)/\partial N = 0$, the value of N corresponding to the minimum value of $E_{total}(N)$ can be obtained (denoted as N_{op}).

Let

$$C_1 = \frac{1}{1-p}\frac{a_2}{1-(1-p)2^{b_2}}\ln((1-p)2^{b_2})$$

and

$$C_2 = \frac{a_1}{(1-p)2^{b_1}}\ln((1-p)2^{b_1}),$$

then, N_{op} is obatined as follows

$$N_{op} = \frac{1}{b_1-b_2}\log_2(\frac{C_1}{C_2}) - \log_2 D_0 + 1. \tag{12}$$

In general, voxelization is performed before point clouds are processed, where voxelization includes the process of position quantization, duplicate point removal, and attributes assignment to the remaining points. The voxels are unit cubes. Specifically, the locations of all the points within a voxel are quantized to the voxel centre, so the minimum distance between a pair of points is 1. D_0, as the sampling distance, must be greater than 1; then, $log_2 D_0$ is greater than 0. According to Equation (12), N_{op} increases when D_0 decreases in a logarithmic way. The greater D_0 is, the smaller N_{op} is. When D_0 doubles, N_{op} decreases by a value of one. The above discussion shows how to determine N_{op} given D_0. D_0 also has an impact on the coding performance of PCC. How to jointly determine the most proper value of D_0 based on N_{op} is one of our future work ideas.

In this paper, D_0 and p are given as known quantities; the fitting parameters a_1, b_1, a_2 and b_2 are obtained by fitting the prediction error according to different point clouds; and then N_{op} can be calculated by Equation (12). That is, when N_LOD is N_{op}, $E_{total}(N_{op})$ is the minimum, and the coding performance is optimized.

It is worth noting that the above model is based on the sampling distance of the LOD. However, a later version of G-PCC adopts a new LOD generation algorithm, which is based on the shift of Morton code (denoted as Morton-based LOD) compared to the traditional method (denoted as distance-based

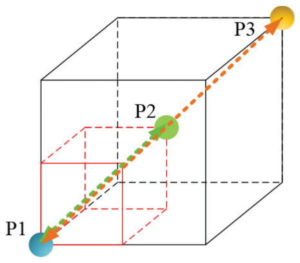

Figure 14: The distance of the corresponding vertices with different shift bits of Morton code.

LOD), where the Morton code is obtained by interleaving the binary values of the three-dimensional coordinates of each point [13]. Morton-based LOD is a fast implementation of distance-based LOD with an approximated result in LOD generation. Essentially, the idea is the same. Therefore, the model is also suitable for Morton-based LOD.

In the process of Morton-based LOD using the model for optimization, how to estimate the initial sampling distance (denoted as D_0') is the key. Considering that Morton-based LOD is an approximate implementation based on the method of distance, the maximum distance between the vertices of the cube is approximately regarded as the sampling distance of LOD.

According to the Morton-based LOD, the initial right shift bits are denoted as S_0, then the right shift bits on each coordinate are $S_0/3$, which is equivalent to enlarging each coordinate by $2^{S_0/3}$ times. After the shift operation, a cube is obtained. The maximum distance between the vertices is the diagonal distance, that is, the side length of the cube multiplied by $\sqrt{3}$, so the approximate initial sampling distance is $D_0' = 2^{(S_0-3)/3}\sqrt{3}$. For example, Figure 14 shows the sampling distance when the initial right shift bits are 3 and 6 respectively, in which the distance between P1 and P2 corresponds to the right shift 3 bits, and the distance between P1 and P3 corresponds to the right shift 6 bits. After the initial sampling distance is calculated, the optimal number of LODs can be obtained from Equation (12).

5 Experimental Results and Discussions

In this section, we conduct extensive experiments to evaluate the proposed method on two different LOD generation algorithms separately, which are distance-based LOD [14] and Morton-based LOD [13]. In the experiment, all the point clouds in test class A of the Static Objects and Scenes database [6, 7] except those in the analysis dataset have been employed as the test dataset; these data are widely used to evaluate the compression performance of various point clouds [6, 9, 10]. The proposed method is implemented in the G-PCC reference software TMC13 v9 and v14 [6, 15, 16]. A Hewlett-Packard PC with AMD Ryzen 5 PRO 2400G, 3.60 GHz CPU, and 16GB of RAM running a 64-bit Win10 Operating system is the hardware used in the experiments.

The experiments include: (1) a comparison between the optimized N_LOD obtained through actual coding and the optimal N_LOD calculated according to the proposed method; (2) an analysis of the improvements in the coding performance and the time complexity of the proposed method induced by the N_LOD; and (3) a discussion about the different characteristics of point clouds.

5.1 Performance of the Proposed Method

To verify the effectiveness of the proposed method, the optimized N_LOD (i.e., N_{op}) is calculated by using Equation (12). Note that the parameters in $f(d_{1,j}) = a_1 d_{1,j}^{b_1}$ and $g(d_{2,i}) = a_2 d_{2,i}^{b_2}$ are obtained through fitting, which is provided by a pre-analysis.

Specifically, some points are collected for pre-analysis purpose to balance the fitting accuracy and the computational complexity. The points used for pre-analysis, are obtained by intercepting some points (1/10 of all points, empirically) according to the Morton-based order. For most point clouds, these points can represent the characteristics of the point clouds relatively completely, and the Morton-based decimation is very quick while avoiding the problem of data sensitivity. Provided with the selected group of points, it is decomposed into five LODs, and the averaged prediction errors of the detail levels and refinement levels corresponding to different distances are used for fitting the parameters a_1, b_1, a_2 and b_2.

If $\partial E_{total}(N)/\partial N$ is always approximately 0 and has nothing to do with N_LOD, N_{op} is 1. Otherwise, the concavity or convexity of $E_{total}(N)$ should be judged by calculating the second derivative. If it is less than 0, which means that $E_{total}(N)$ is concave, N_{op} is 1; if it is greater than 0, i.e., $E_{total}(N)$ is convex, N_{op} is obtained by Equation (12).

Finally, N_{op} is rounded to obtain the operable version of the optimized N_LOD, which is denoted as $\hat{N}_{op}$ and shown in Table 1.

The actual optimized N_LOD (denoted as N^*_{op})) which is summarized from the bitrates obtained by compressing the test dataset with different N_LOD values, are compared with $\hat{N}_{op}$. One can see that they are almost identical. Specifically, for distance-based LOD, $N^*_{op} = \hat{N}_{op}$ for 14 point clouds in the test dataset, while $|N^*_{op} - \hat{N}_{op}| = 1$ for the other 4 point clouds, and for the Morton-based LOD, $N^*_{op} = \hat{N}_{op}$ for 16 point clouds in the test dataset, while $|N^*_{op} - \hat{N}_{op}| = 1$ for the other 2 point clouds. The experimental results show that this method is applicable to both algorithms.

Table 1: Optimized values of N_LOD from the proposed method and an actual test.

		Distance-based LOD		Morton-based LOD	
Dataset	N_G	$\hat{N}_{op}$	N^*_{op}	$\hat{N}_{op}$	N^*_{op}
basketball_player_vox11	12	**6**	**6**	**1**	**1**
boxer_viewdep_vox12	12	**5**	**5**	**1**	**1**
dancer_vox11	12	**7**	**7**	**1**	**1**
Egyptian_mask_vox12	11	**5**	**5**	**1**	**1**
Facade_00009_vox12	12	**6**	**6**	**2**	**2**
Facade_00064_vox11	13	4	3	**1**	**1**
Head_00039_vox12	13	**1**	**1**	**1**	**1**
House_without_roof_vox12	13	**1**	**1**	**1**	**1**
longdress_vox12	12	**5**	**5**	1	2
longdress_vox10	13	**6**	**6**	**2**	**2**
loot_viewdep_vox12	12	6	7	**1**	**1**
loot_vox10_1200	12	**6**	**6**	**2**	**2**
queen_0200	12	1	2	**1**	**1**
redandblack_vox12	12	**6**	**6**	**1**	**1**
Shiva_00035_vox12	12	**5**	**5**	**1**	**1**
soldier_vox12	12	5	6	1	2
Thaidancer_viewdep_vox12	12	**5**	**5**	**2**	**2**
ULB_Unicorn_vox13	12	**1**	**1**	**1**	**1**

Table 1 also shows the recommended values of N_LOD (denoted as N_G) in G-PCC CTC. For each point cloud, $\hat{N}_{op}$ is much less than N_G, reducing the complexity of both encoding and decoding.

5.2 Coding Performance and Time Complexity

The compared work is the general solution for LOD generation, where N_LOD is set to a very large value (denoted as N_G) so that the LOD generation can be executed up to the last point or with only a few points left.

In our work, however, we propose a content-adaptive number of LODs for a given point cloud. To the best of the authors' knowledge, the proposed work is the first attempt to determine the optimized numbers of LODs for point clouds that is both content-adaptive and computationally efficient.

By setting N_LOD to $\hat{N}_{op}$ and N_G individually, lossless compression is performed on the test dataset, and the results in terms of the attribute bitrate ratio and time complexity are summarized in Table 2.

Table 2: Comparison of attribute bitrate and time complexity.

	Distance-based LOD			Morton-based LOD		
	BR (%)	TC (%)		BR (%)	TC (%)	
Dateset	Attribute	TC_{Enc}	TC_{Dec}	Attribute	TC_{Enc}	TC_{Dec}
basketball_player_vox11	100.00	96.94	92.45	99.09	95.07	97.24
boxer_viewdep_vox12	100.00	96.42	95.76	99.67	96.48	96.65
dancer_vox11	99.99	97.21	94.17	98.85	96.06	98.15
Egyptian_mask_vox12	100.00	97.51	93.26	99.47	92.68	94.87
Facade_00009_vox12	100.00	97.36	94.62	99.74	95.54	94.19
Facade_00064_vox11	100.00	95.13	92.22	99.70	97.50	97.47
Head_00039_vox12	99.09	44.36	43.42	98.46	91.23	91.43
House_without_roof_vox12	99.79	44.01	41.76	99.13	95.41	95.37
longdress_vox12	100.00	96.58	94.33	100.20	93.45	93.31
longdress_vox10	99.99	93.85	92.89	99.83	96.34	95.31
loot_viewdep_vox12	100.01	96.68	93.39	100.00	94.72	95.67
loot_vox10_1200	99.99	98.36	97.56	100.00	95.57	94.84
queen_0200	99.99	45.20	44.29	98.19	95.08	93.62
redandblack_vox12	100.00	95.58	94.21	99.61	92.08	92.67
Shiva_00035_vox12	99.99	95.05	94.37	99.22	94.42	94.09
soldier_vox12	100.01	96.82	94.94	100.16	92.65	93.37
Thaidancer_viewdep_vox12	100.00	96.35	94.98	100.00	95.67	95.75
ULB_Unicorn_vox13	98.91	43.16	40.35	94.84	93.83	94.26
Average	**99.88**	**84.81**	**82.72**	**99.23**	**94.65**	**94.90**

The attribute time complexity (TC) in the table refers to the relative ratio of the encoding and decoding times of the attribute in the two cases above, and it is calculated as follows:

$$TC_{Enc} = EncT_{op}/EncT_G,$$
$$TC_{Dec} = DecT_{op}/DecT_G,$$

where $EncT_{op}$ is the attribute encoding time corresponding to $\hat{N}_{op}$, $EncT_G$ is the attribute encoding time corresponding to N_G, $DecT_{op}$ is the attribute decoding time corresponding to $\hat{N}_{op}$, and $DecT_G$ is the attribute decoding time corresponding to N_G.

The bitrate ratio (BR) is calculated as follows:

$$BR = R_{op}/R_G.$$

where R_G is the attribute bitrate when N_LOD is N_G, and R_{op} is the attribute bitrate when N_LOD is $\hat{N}_{op}$.

The attribute bitrate obtained by using the proposed N_LOD in this paper is slightly less than that obtained by using the N_LOD provided by G-PCC configuration scripts on average. Furthermore, the encoding time and decoding time of the attribute are significantly reduced, i.e., ~15.2% and ~17.3% time savings on average for distance-based LOD, and ~5.4% and ~5.1% time savings for Morton-based LOD, respectively. The latter saves less time, for the reason that Morton-based LOD is already a fast implementation of distance-based LOD. Nevertheless, further reduction in complexity is still achieved using the proposed method.

This method not only achieves the best lossless compression performance for the point cloud but also avoids conducting unnecessary LOD decomposition in advance, thus saving valuable attribute encoding and decoding time.

5.3 Different Characteristics of Point Clouds

From the experiments, the trend of the attribute bitrate versus N_LOD is summarized into three cases, as shown in Figure 15.

In the first case, which is the most common case, as shown in Figure 15(a), the bitrate decreases obviously after LOD generation and reaches its minimum when the point cloud is decomposed into a certain number of LODs. The coding performance is not further improved. Considering that the larger N_LOD is, the higher the time complexity will be, the value of N_LOD corresponding to the first iteration that yields the minimum bitrate is selected as the optimal number.

In very few cases, the bitrate hardly changes after LOD generation, as shown in Figure 15(b). Another rare possibility is that the bitrate increases

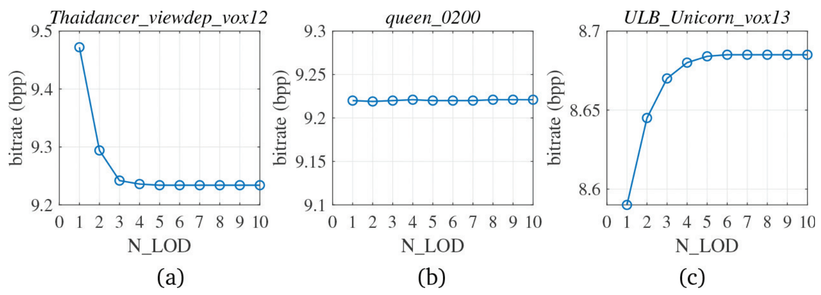

Figure 15: Attribute bitrates under different numbers of LODs: (a) *Thaidancer_ viewdep_ vox12*; (b) *queen_ 0200*; (c) *ULB_ Unicorn_ vox13*.

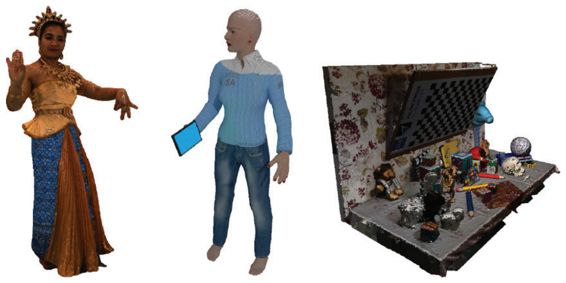

Figure 16: Three typical point clouds. From left to right: *Thaidancer_ viewdep_ vox12*, *queen_ 0200* and *ULB_ Unicorn_ vox13*.

along with LOD generation and reaches its maximum after a certain number of LODs are decomposed, as shown in Figure 15(c). In both cases, the optimized N_LOD corresponding to the minimum bitrate is 1.

Figure 16 provides the point clouds corresponding to the above three cases.

The most common type of point cloud is *Thaidancer_ viewdep_ vox12*. It is noticed that the implementation of the prediction process in coding order will be limited to a certain direction if LOD is not used. There are not only strongly correlated points but also weakly correlated points in the neighbourhood. When LOD generation is incorporated, however, some of the points (i.e., the points in the refinement levels) can select reference points from all directions, making the prediction more accurate and yielding a performance gain. As a cost, the other points (the points in the topmost detail level) are still limited by the coding order or directionality when selecting reference points. Furthermore, LOD increases the distances between adjacent points in the topmost detail level so that the correlations are reduced, resulting in a certain performance loss. When the gain yielded by the further generation of LODs cannot compensate for the induced loss, the generation of LODs is no longer useful. In other words, the optimal number of LODs is achieved by then. Overall, the gain and loss determine the optimal number of generated LODs, and this is a trade-off process.

It is noted that after reaching the optimal number of LODs, for most point clouds, with further LOD generation, the performance gain and loss are both very limited. At the same time, the number of remaining points is very small, declining exponentially with further LOD generation and making the impact of these points negligible. Therefore, the compression performance remains almost the same when the number of LODs is optimal and above, i.e., the BR stays at 100%. However, for a few point clouds, such as *longdress_ vox10* and *dancer_ vox11*, after reaching the optimal number of LODs, there are still certain correlations among adjacent points in the topmost detail level. If LOD generation is further carried out, with the increase in the distances among adjacent points in the topmost detail level, their correlations decrease greatly, while the improvement in prediction accuracy for the refinement levels is very limited, so the overall gain is less than the loss. In other words, for such point clouds, further LOD decomposition will lead to a very slight performance loss if the number of LODs is larger than the optimal value. This is the reason why the BR is 99.99% for *longdress_ vox10* and *dancer_ vox11*.

Very interestingly, some point clouds, such as *queen_ 0200*, have very similar attribute values within a large range, and the correlations among points are very strong. It is noted that *queen_ 0200* is a computer-generated point cloud, the texture of which is relatively simple, so the strong correlations among points are natural. For such point clouds, the coding result without LOD generation is already good since the nearest neighbours in the coding order already serve as good references for prediction purposes. The gain from LOD is therefore negligible, so the performance is almost the same with or without LOD.

As an example of the third case, the content of *ULB_ Unicorn_ vox13* is very complex, where only the very close points have correlations, and LOD generation has a negative effect on the prediction performance.

Nevertheless, the proposed model adapts to different characteristics of point clouds through a pre-analysis of the predicted error, and the optimal number of LODs for different point cloud contents can be found.

6 Conclusion

In this paper, by establishing a mathematical prediction error model for LODs, an optimized solution for determining the appropriate number of LODs is proposed, which leads to content-adaptive LOD generation for a given point cloud. The experimental results show that the optimized number of LODs calculated by the proposed method is consistent with the optimized number of

LODs obtained through actual coding. The proposed work is well adapted to the different contents of various point clouds, achieving optimal performance while reducing the attribute encoding and decoding complexity by avoiding unnecessary LODs.

This method is not only suitable for lossless compression but also useful for point cloud-based scalable coding because it effectively calculates prediction errors. Future work will include extending the proposed method to lossy compression and scalable coding.

Biographies

Lei Wei received the B.E. degree in information engineering from Xi'an Jiaotong University in 2007 and the M.E. degree from Xi'an Institute of Electromechanical Information Technology in 2010. He is currently working toward the Ph.D. degree in Northwestern Polytechnical University, Xi'an, China. His research interests include point cloud compression, 3D information processing and coding optimization.

Shuai Wan received the B.E. degree in telecommunication engineering and M.E. degree in communication and information system from Xidian University, Xi'an, China, in 2001 and 2004, respectively, and the Ph.D. degree in electronic engineering from Queen Mary, University of London in 2007. She is now a Professor in Northwestern Polytechnical University, Xi'an, China. Her research interests include point cloud compression, scalable/multiview video coding, video quality assessment, and hyperspectral image compression.

Fuzheng Yang received the B.E. degree in Telecommunication Engineering, the M.E. degree and the Ph.D. in Communication and Information System from Xidian University, Xi'an, China, in 2000, 2003 and 2005, respectively. He became a lecturer and an Associate Professor in Xidian University in 2005 and 2006, respectively. He has been a professor of communications engineering with Xidian University since 2012. During 2006–2007, he served as a visiting scholar and postdoctoral researcher in Department of Electronic Engineering in Queen Mary, University of London. His research interests include point cloud compression, video quality assessment, video coding and multimedia communication.

Zhecheng Wang received the B.E. degree from Zhengzhou University, China, in 2014, and the M.S. degree from the Technical University of Munich, Germany, in 2017. He is currently a Ph.D student in the School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China. His research interests include point cloud compression, video coding and multimedia communication.

References

[1] E. J. Candes and M. B. Wakin, "An Introduction To Compressive Sampling", *IEEE Signal Processing Magazine*, 25(2), 2008, 21–30, DOI: 10.1109/MSP.2007.914731.

[2] P. A. Chou, M. Koroteev, and M. Krivokuća, "A Volumetric Approach to Point Cloud Compression—Part I: Attribute Compression", *IEEE Transactions on Image Processing*, 29, 2020, 2203–16, DOI: 10.1109/TIP.2019.2908095.

[3] A. Clark, "Hierarchical Geometric Models for Visible Surface Algorithms", *Communications of the ACM*, 19(10), 1976, 547–54, DOI: 10.1145/360349.360354.

[4] R. A. Cohen, D. Tian, and A. Vetro, "Attribute Compression for Sparse Point Clouds Using Graph Transforms", in *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, 2016, 1374–8, DOI: 10.1109/ICIP.2016.7532583.

[5] R. A. Cohen, D. Tian, and A. Vetro, "Point Cloud Attribute Compression Using 3-D Intra Prediction and Shape-Adaptive Transforms", in *2016 Data Compression Conference (DCC)*, 2016, 141–50, DOI: 10.1109/DCC.2016.67.

[6] *Common Test Conditions for PCC*, "Standard ISO/IEC JTC1/SC29/WG11 MPEG, N19084, MPEG 3DG, Brussels, BE", 2020.

[7] E. d'Eon, B. Harrison, T. Myers, and P. A. Chou, *8i Voxelized Full Bodies - A Voxelized Point Cloud Dataset*, Standard ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG), WG11M40059, Geneva, CH, 2017.

[8] K. L. Damkjer and H. Foroosh, "Lattice-Constrained Stratified Sampling for Point Cloud Levels of Detail", in, Vol. 58, No. 8, 2020, 5627–41, DOI: 10.1109/TGRS.2020.2967880.

[9] R. L. de Queiroz and P. A. Chou, "Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform", *IEEE Transactions on Image Processing*, 25(8), 2016, 3947–56, DOI: 10.1109/TIP.2016.2575005.

[10] R. L. de Queiroz and P. A. Chou, "Transform Coding for Point Clouds Using a Gaussian Process Model", *IEEE Transactions on Image Processing*, 26(7), 2017, 3507–17, DOI: 10.1109/TIP.2017.2699922.

[11] *Efficient Implementation of the Lifting Scheme in TMC13*, "Standard ISO/IEC JTC1/SC29/WG11, M43781, MPEG 3DG, Ljubljana, SI", 2018.

[12] Y. Fan, Y. Huang, and J. Peng, "Point Cloud Compression Based on Hierarchical Point Clustering", in *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, Kaohsiung, 2013, 1–7, DOI: 10.1109/APSIPA.2013.6694334.

[13] *G-PCC Codec Description v12*, "Standard ISO/IEC JTC1/SC29/WG7 MPEG, N0251, MPEG 3DG, Virtual", 2021.

[14] *G-PCC Codec Description v6*, "Standard ISO/IEC JTC1/SC29/WG11 MPEG, N19091, MPEG 3DG, Brussels, BE", 2020.

[15] *G-PCC Test Model v14 User Manual*, "Standard ISO/IEC JTC1/SC29/WG7, N00094, MPEG 3DG, Virtual", 2021.

[16] *G-PCC Test Model v9 User Manual*, "Standard ISO/IEC JTC1/SC29/WG11 MPEG, N19083, MPEG 3DG, Brussels, BE", 2020.

[17] S. Gu, J. Hou, H. Zeng, and H. Yuan, "3D Point Cloud Attribute Compression via Graph Prediction", *IEEE Signal Processing Letters*, 27, 2020, 176–80, DOI: 10.1109/LSP.2019.2963793.

[18] S. Gu, J. Hou, H. Zeng, H. Yuan, and K.-K. Ma, "3D Point Cloud Attribute Compression Using Geometry-Guided Sparse Representation", *IEEE Transactions on Image Processing*, 29, 2020, 796–808, DOI: 10.1109/TIP.2019.2936738.

[19] J. Hou, "Permuted Sparse Representation for 3D Point Clouds", *IEEE Signal Processing Letters*, 26(12), 2019, 1847–51, DOI: 10.1109/LSP.2019.2949724.

[20] B. Kathariya, V. Zakharchenko, Z. Li, and J. Chen, "Level-of-Detail Generation Using Binary-Tree for Lifting Scheme in LiDAR Point Cloud Attributes Coding", in *2019 Data Compression Conference (DCC)*, Snowbird, UT, USA, 2019, 580–0, DOI: 10.1109/DCC.2019.00092.

[21] H. Liu, H. Yuan, Q. Liu, J. Hou, and J. Liu, "A Comprehensive Study and Comparison of Core Technologies for MPEG 3-D Point Cloud Compression", *IEEE Transactions on Broadcasting*, 66(3), 2020, 701–17, DOI: 10.1109/TBC.2019.2957652.

[22] D. Luebke, M. Reddy, J. D. Cohen, A. Varshney, B. Watson, and R. Huebner, *Level of Detail for 3D Graphics*, San Francisco, CA, USA: Morgan Kaufmann Publisher, 2002, 151–82.

[23] R. Mekuria, K. Blom, and P. Cesar, "Design, Implementation, and Evaluation of a Point Cloud Codec for Tele-Immersive Video", *IEEE Transactions on Circuits and Systems for Video Technology*, 27(4), 2017, 828–42, DOI: 10.1109/TCSVT.2016.2543039.

[24] J. Peng, C.-S. Kim, and C.-C. J. Kuo, "Technologies for 3D Mesh Compression: A Survey", *Journal of Visual Communication and Image Representation*, 16(6), 2005, 688–733.

[25] S. Schwarz *et al.*, "Emerging MPEG Standards for Point Cloud Compression", *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(1), 2019, 133–48, DOI: 10.1109/JETCAS.2018.2885981.

[26] Y. Shao, Q. Zhang, G. Li, and Z. Li, "Hybrid Point Cloud Attribute Compression Using Slice-based Layered Structure and Block-based Intra Prediction", in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, 1199–207.

[27] W. R. Tobler, "A Computer Movie Simulating Urban Growth in the Detroit Region", *Economic Geography*, 46, 1970, 234–40.

[28] S. Wan and F. Yang, *New Efficient Video Coding H.265/HEVC: Principle, Standard and Implementation*, Beijing, China: Publishing House of Electronics Industry, 2014, 94–104.

[29] L. Wei, S. Wan, Z. Sun, X. Ding, and W. Zhang, "Weighted Attribute Prediction Based on Morton Code for Point Cloud Compression", in *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, London, United Kingdom, 2020, 1–6, DOI: 10.1109/ICMEW46912.2020.9105953.

[30] C. Zhang, D. Florêncio, and C. Loop, "Point Cloud Attribute Compression with Graph Transform", in *2014 IEEE International Conference on Image Processing (ICIP)*, Paris, 2014, 2066–70, DOI: 10.1109/ICIP.2014.7025414.