Model-based Joint Bit Allocation between Geometry and Color for Video-based 3D Point Cloud Compression

Qi Liu, Hui Yuan, *Senior Member, IEEE*, Junhui Hou, *Senior Member, IEEE*, Raouf Hamzaoui, *Senior Member, IEEE*, and Honglei Su

***Abstract*—In video-based 3D point cloud compression, the quality of the reconstructed 3D point cloud depends on both the geometry and color distortions. Finding an optimal allocation of the total bitrate between the geometry coder and the color coder is a challenging task due to the large number of possible solutions. To solve this bit allocation problem, we first propose analytical distortion and rate models for the geometry and color information. Using these models, we formulate the joint bit allocation problem as a constrained convex optimization problem and solve it with an interior point method. Experimental results show that the rate-distortion performance of the proposed solution is close to that obtained with exhaustive search but at only 0.66% of its time complexity.**

***Index Terms*—Point cloud compression, bit allocation, rate-quantization (R-Q) model, distortion-quantization (D-Q) model, rate-distortion optimization (RDO).**

I. Introduction

With the rapid development of 3D scanning techniques [1], point clouds are now readily available and popular [2]. There are already many 3D point cloud applications in the fields of 3D modeling [3] [4], automatic driving [5], 3D printing [6], augmented reality [7], etc. Compared with traditional 2D images and videos, a 3D point cloud describes the 3D scene or object with the geometry information and the corresponding attributes (e.g., color, reflectance) [8]. To represent a 3D scene accurately, millions of points must be captured and processed. This huge data volume poses a severe challenge for efficient storage and transmission. In the past few years, major progress in both static and dynamic point cloud compression has been made [9] (see Section II).

To standardize 3D point cloud compression (PCC) technologies, the Moving Pictures Expert Group (MPEG) launched a call for proposals in 2017. As a result, three point cloud compression technologies were developed: surface point cloud compression (S-PCC) with software platform TMC1 [10] for static point cloud data, video-based point cloud compression (V-PCC) with software platform TMC2 [11] for dynamic content, and LIDAR point cloud compression (L-PCC) with software platform TMC3 [12] for dynamically acquired point clouds. Recently, L-PCC and S-PCC were merged under the name geometry-based point cloud compression (G-PCC) with software platform TMC13 [13].

In this paper, we focus on the V-PCC platform (TMC2) due to its excellent performance for compressing both static and dynamic point clouds. The main philosophy behind V-PCC is to leverage state-of-the-art video coders for PCC [11]. This is essentially achieved by decomposing each point cloud of a sequence of 3D point clouds into a set of patches, which are independently mapped to a 2D grid of uniform blocks. This mapping is then used to store the geometry and color information as one geometry image and one color image. The sequences of geometry images and color images corresponding to the dynamic point cloud are then compressed separately with a video coder, e.g., H.265/HEVC. Finally, the geometry and color videos, together with metadata (occupancy map for the 2D grid, auxiliary patch and block information) are used to reconstruct the dynamic 3D point cloud (see [14] for more details). Fig. 1 shows the main components of the V-PCC encoder.

The bitstream of a compressed 3D point cloud consists of two parts: geometry information and color information. For a given platform, the size of each part is controlled by a quantization parameter, which can take a large number of values. At the same time, quantization introduces distortion, which may affect the reconstruction quality. The aim of this paper is to find the pair of quantization parameter values (one for the geometry and one for color) that minimizes the reconstruction error subject to a constraint on the total number of bits. Since the number of candidates is very large, finding an optimal solution with exhaustive search is too time consuming

This work was supported in part by the National Natural Science Foundation of China under Grants 61571274 and 61871342; in part by the Shandong Provincial Key Research and Development Plan under Grant 2017CXGC1504; in part by the open project program of state key laboratory of virtual reality technology and systems,Beihang University, under Grant VRLAB2019B03; in part by Shenzhen Science and Technology Research and Development Funds under Grant JCYJ20170818103244664; in part by the Hong Kong Research Grants Council under grants 9042955 (CityU 11202320), and in part by the Shandong Provincial Natural Science Foundation, China under Grants ZR2018PF002. *Corresponding author: Hui Yuan.*

Qi Liu is with the School of Information Science and Engineering, Shandong University, Qingdao 266237, China. She is also with the Shenzhen Research Institute of Shandong University, Shenzhen 518057, China (Email: sdqi.liu@gmail.com).

Hui Yuan is with the School of Control Science and Engineering, Shandong University, Ji'nan 250061, China. He is also with the Shenzhen Research Institute of Shandong University, Shenzhen 518057, China (Email: huiyuan@sdu.edu.cn).

Junhui Hou is with the Department of Computer Science, City University of Hong Kong, Hong Kong (Email: jh.hou@cityu.edu.hk).

Raouf Hamzaoui is with the School of Engineering and Sustainable Development, De Montfort University, Leicester, UK (Email: rhamzaoui@dmu.ac.uk).

Honglei Su is with the School of Electronic Information, Qingdao University, Qingdao 266071, China (Email: suhonglei@qdu.edu.cn).

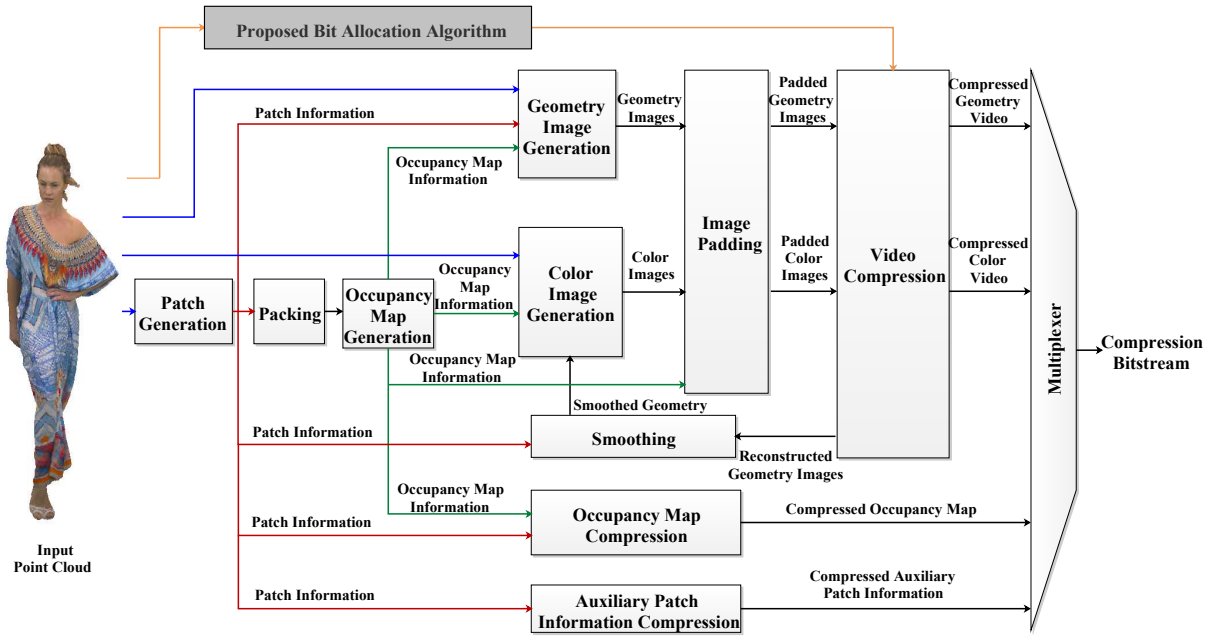


Fig. 1. General framework of the V-PCC encoder.

and infeasible in practice as it requires encoding and decoding the point cloud multiple times. In this paper, we address this challenge by proposing analytical models for the rate and distortion and using an interior point method [15] to efficiently solve the rate-distortion optimization problem.

The contributions of this paper are as follows.

1) We propose analytical models that characterize the rate and distortion of the geometry and color as functions of the V-PCC quantization steps. We show in particular that the color distortion can be expressed as the sum of two independent terms, one that depends on the geometry quantization step only and one that depends on the color quantization step only.
2) We exploit the proposed analytical models to formulate the bit allocation problem for V-PCC as a constrained convex optimization problem where the variables are the quantization steps of the V-PCC video coders.
3) We use an interior point method to efficiently solve the optimization problem.
4) We integrate our bit allocation algorithm into the latest test model of V-PCC (see Fig. 1).

To the best of our knowledge, our work is the first one to study the rate and distortion characteristics of the V-PCC coder in detail. The proposed rate and distortion models can be used not only for the joint bit allocation of the geometry and color information but also for other important tasks such as rate control and error protection for transmission over unreliable channels.

The remainder of this paper is organized as follows. In Section II, related work is briefly introduced. In Section III, we formulate the rate-distortion optimization problem for V-PCC and provide analytical models for the distortion and rate as a function of the quantization steps. In Section IV, we use these models to solve the rate-distortion optimization problem with an interior point method. In Section V, we present experimental results where we compare the accuracy and the time complexity of our approach to those of exhaustive search. Our results show that our approach has similar rate-distortion performance to exhaustive search with only 0.66% of its time complexity. Section VI gives our conclusions and suggests future work.

II. Related Work

To handle the irregular data structure of 3D point clouds, several effective transform techniques have been proposed. These include shape-adaptive discrete cosine transform [16], graph transforms [17] [18] [19] [20], region-adaptive hierarchical transforms [21]–[24], Gaussian process transforms [25] [26], and sparse representation [27] [28] [29]. Based on the previously mentioned graph transforms, Shao *et al.* [30] further combined a slice partition scheme and an intra prediction technique to improve the performance of attribute compression. Instead of compressing the irregular data directly, some researchers [31] [32] try to map the irregular data to a regular representation (e.g., a 2D plane) to simplify the task. Similar ideas were previously proposed to compress 3D human motion [33] and 3D facial expression [34]–[37]. All the described methods are mainly designed for static point clouds.

Since dynamic point clouds are becoming more and more important in practice, efficient compression methods for dynamic point clouds are also required. Because of the inter-frame redundancy of dynamic point clouds, motion estimation and motion compensation are the key technologies to effectively compress dynamic point clouds. Based on the V-PCC platform that is provided by MPEG, Li *et al.* [38] proposed using the geometry or auxiliary information to derive a better motion vector predictor to deal with the patch inconsistency

problem for both the geometry and attribute videos. Thanou, Chou and Frossard [39] focused on motion estimation by using a spectral graph wavelet descriptor. De Queiroz and Chou [40] proposed a motion compensation approach to encode dynamic voxelized point clouds. Anis, Chou and Ortega [41] simplified motion compensation by representing sets of frames in a consistently evolving high-resolution subdivisional triangular mesh.

In our previous work [8], a bit allocation approach for the Point Cloud Library (PCL)-based codec [31] [42] was proposed. The PCL-based encoder recursively subdivides the point cloud into eight subsets. This results in an octree data structure, where the position of each voxel is represented by its center whose attribute (color) is set to the average of the attributes of the enclosed points. Then the octree data is encoded by an entropy coder. In order to encode the color attributes, the octree attributes are mapped directly to a structured image grid using depth first tree traversal and then the image grid is encoded with the JPEG codec.

There are two main differences between this paper and our previous work [8]. First, since the two codecs are different, both the rate-distortion analysis and the modeling of the rate and distortion functions are different. In particular, in our previous paper [8], the rate is modeled as an exponential function of the maximum octree level and JPEG quality factor while in this paper it is modeled as a polynomial function of the quantization steps of the two video coders. Second, the objective function for the rate-distortion optimization, as well as the optimization variables are different. In this paper, the objective function is the overall distortion, which includes the distortion of both geometry and color information, and the optimization variables are the quantization steps for the geometry and color information. In [8], only the color distortion is minimized, and the optimization variables are the octree level and the JPEG quality factor.

III. Rate and Distortion Models Derivation

To efficiently solve the bit allocation problem, we formulate it as a constrained optimization problem by deriving rate and distortion models for the 3D point cloud. As the distortion of a 3D point cloud is determined by the coding distortion of both geometry and color information, the bit allocation problem can be expressed as

$$\begin{aligned} &\min_{(R_g,\,R_c)} D(R_g, R_c) \\ &\text{s.t.} \quad R_g + R_c \le R_T, \end{aligned} \tag{1}$$

where the distortion D of the reconstructed 3D point cloud is determined by the color distortion (D_c) and the geometry distortion (D_g); R_g and R_c are the geometry and color bitrate, respectively, and R_T is the target total bitrate. It is worth noting that the occupancy map and auxiliary information also consume bitrate resources. The occupancy map is a binary array that indicates whether a pixel position is occupied or not. The auxiliary information is used to store just a few encoder parameters. In general, both the occupancy map and the auxiliary information are compressed without loss. In addition, their bitrate cost is small and fixed for a given 3D point cloud. Therefore, we do not consider the bitrate of the occupancy map and auxiliary information in (1).

In the V-PCC codec, a geometry video is generated from the input point cloud and compressed with a state-of-the-art video coder (HEVC in the latest version of the reference software). The reconstructed geometry video is then used to generate the color video. Next, the color video is compressed by the same video coder. As the color and geometry distortions are controlled by the quantization parameters (QPs) of the video coder, the distortion models of the color video and the geometry video depend on the quantization characteristics of the video coder. We considered several candidate models, and used statistical analysis to select the model that best balances goodness of fit with simplicity, where simplicity was measured in terms of the number of parameters in the model. The statistical results showed that the linear model is the best candidate.

Since the geometry video is compressed before the color video, the linear model was directly used for the geometry distortion. Because color compression is based on the reconstructed geometry in the V-PCC encoder, the color distortion depends on the geometry distortion. We derived the dependency between the geometry and the color distortions by decomposing the color distortion into two parts based on the definition of distortion and the law of large numbers (see Appendix A). Next, we parameterized each part using a linear model based on the previous analysis.

The derived rate models can be explained similarly. In particular, we found that the influence of the geometry bitrate on the color bitrate is very small. Therefore, we assumed that the rate model for the color video depends only on the quantization parameter of the color video coder.

In the remainder of this section, we exploit the properties of V-PCC and use statistical analysis to propose distortion and rate models, so that the bit allocation problem (1) can be solved analytically.

A. Distortion Model

For the distortion D of the 3D point cloud, we used a linear combination of the geometry distortion D_g and color distortion D_c. That is,

$$D = \omega D_g + (1-\omega)\, D_c, \tag{2}$$

where ω is a weighting factor. For the geometry distortion and color distortion, we used symmetric point-to-point distortions based on the mean squared error (MSE) [43]. Let *A* and *B* denote the original point cloud and the reconstructed point cloud, respectively. The original point cloud is encoded with V-PCC using a given geometry and color quantization parameter pair and the default V-PCC encoder configuration [44], while the reconstructed point cloud is obtained with V-PCC decoding. Then D_g= max $(e_g^{B,A}, e_g^{A,B})$, where

$$e_g^{B,A} = \frac{1}{|B|} \sum_{b_i \in B} \|b_i n_A(b_i)\|_2^2, \tag{3}$$

$|B|$ denotes the number of points in B, $n_A(b_i)$ is the nearest neighbor of b_i in A, and $\|\|_2$ is the Euclidean norm. Similarly,

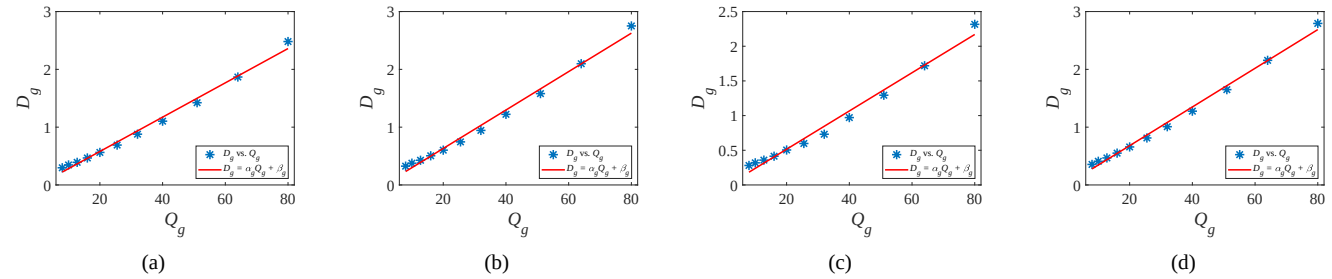


Fig. 2. Illustration of the relationship between the geometry distortion D_g and the geometry quantization step Q_g. (a) *Loot*, (b) *Longdress*, (c) *Queen*, (d) *Redandblack*.

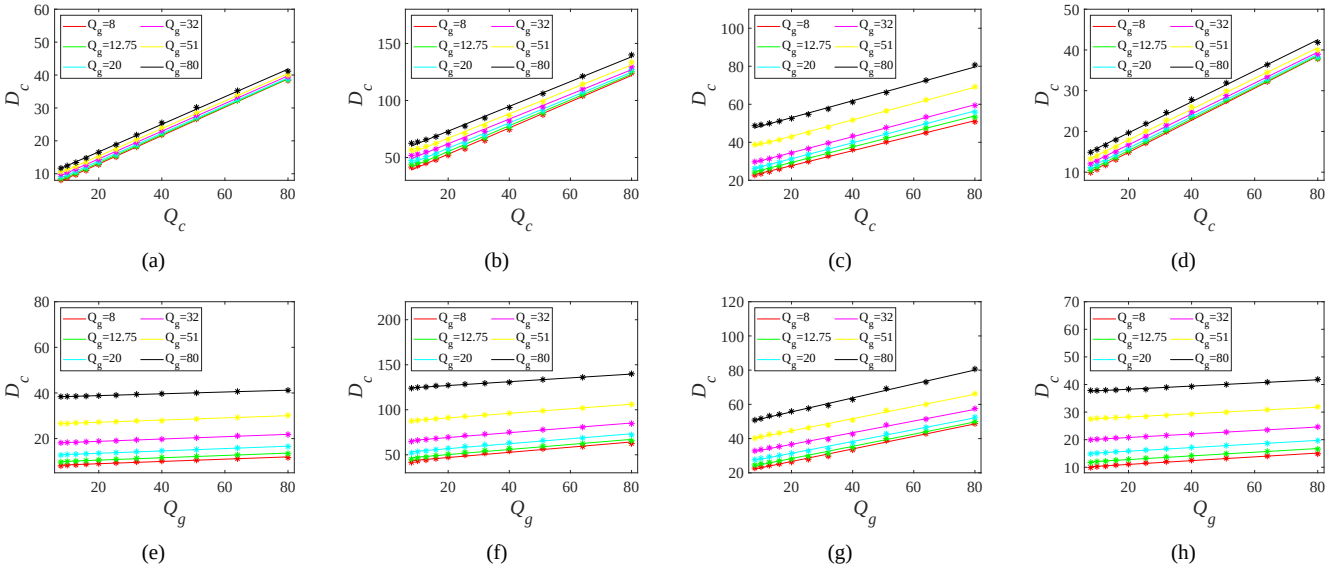


Fig. 3. Statistical results for $f_g(Q_g) = \alpha_{gc} Q_g + \beta_{gc}$ and $f_c(Q_c) = \alpha_{cc} Q_c + \beta_{cc}$. (a)-(d): relationship between D_c and Q_c for *Loot*, *Longdress*, *Queen*, and *Redandblack*, (e)-(f): relationship between D_c and Q_g for *Loot*, *Longdress*, *Queen*, and *Redandblack*.

$D_c = \max\,(e_c^{B,A}, e_c^{A,B})$, where

$$e_c^{B,A} = \frac{1}{|B|} \sum_{b_i \in B} |C(b_i) - C(n_A(b_i))|^2. \tag{4}$$

Here $C(x)$ is a color attribute of point x. For simplicity, we consider only the Y (luminance) component [45] in this paper. Because of the structure of the V-PCC encoder, D_g only depends on Q_g, the quantization step size of the geometry video coder. To obtain a relationship between D_g and Q_g, we conducted a statistical analysis on eight point clouds compressed according to the settings in Table VII. Fig. 2 shows D_g as a function of Q_g for the $Loot$, $Longdress$, $Queen$, and $Redandblack$ point clouds. The results suggest that a linear model

$$D_g = \alpha_g Q_g + \beta_g, \tag{5}$$

where α_g and β_g are model parameters is appropriate. Table I gives the squared correlation coefficient (SCC) and the root mean squared error (RMSE) between the actual data and the fitted data when the model parameters α_g and β_g are computed by the method of least squares. The SCCs are close to 0.99, while the maximum RMSE is only 0.09, confirming the accuracy of the model.

TABLE I
ACCURACY OF PROPOSED GEOMETRY DISTORTION MODEL (5)

Point Cloud	SCC	RMSE
Andrew	0.98	0.08
David	0.99	0.07
Longdress	0.99	0.08
Redandblack	0.99	0.07
Loot	0.99	0.07
Queen	0.98	0.09
Soldier	0.99	0.07
Basketballplayer	0.98	0.08
Average	0.99	0.08

Because color compression is based on the reconstructed geometry in the V-PCC encoder, the color distortion depends on the geometry distortion. In Appendix A, we show that for the original point cloud *A* and the corresponding reconstructed point cloud *B*, we have

$$e_c^{B,A} = f_g^{B,A}(Q_g) + f_c^{B,A}(Q_c), \tag{6}$$

where Q_c is the quantization step size of the color video coder, $f_g^{B,A}(Q_g)$ depends on Q_g only and $f_c^{B,A}(Q_c)$ depends on Q_c only. Similarly, $e_c^{A,B}$ can be decomposed as $e_c^{A,B} =$

TABLE II
ACCURACY OF PROPOSED COLOR DISTORTION MODEL (10)

Point Cloud	SCC	RMSE
Andrew	1.00	0.83
David	1.00	0.08
Longdress	1.00	1.82
Redandblack	1.00	0.36
Loot	1.00	0.28
Queen	1.00	0.75
Soldier	1.00	0.51
Basketballplayer	1.00	0.11
Average	1.00	0.60

TABLE III
ACCURACY OF PROPOSED DISTORTION MODEL (11)

Point Cloud	SCC	RMSE
Andrew	1.00	0.41
David	1.00	0.05
Longdress	1.00	0.90
Redandblack	1.00	0.18
Loot	1.00	0.14
Queen	1.00	0.40
Soldier	1.00	0.25
Basketballplayer	1.00	0.05
Average	1.00	0.26

$f_g^{A,B}(Q_g) + f_c^{A,B}(Q_c)$. Thus, D_c can be written as

$$\begin{aligned} D_c &= \max(e_c^{B,A}, e_c^{A,B}) \\ &= \max(f_g^{B,A}(Q_g) + f_c^{B,A}(Q_c), f_g^{A,B}(Q_g) + f_c^{A,B}(Q_c)) \\ &= f_g(Q_g) + f_c(Q_c). \end{aligned} \tag{7}$$

To obtain an analytical expression for $f_g(Q_g)$ and $f_c(Q_c)$, statistical experiments were conducted, as shown in Fig. 3. To investigate the exact expression of $f_c(Q_c)$, which characterizes the relationship between Q_c and D_c, the influence of Q_c on the distortion of the comparison color point cloud was statistically analyzed by setting Q_g to fixed values. As suggested in Fig. 3(a)-3(d), for a fixed Q_g, a linear model

$$f_c(Q_c) = \alpha_{cc} Q_c + \beta_{cc}, \tag{8}$$

where α_{cc} and β_{cc} are model parameters is appropriate. Similarly, to derive an analytical expression for $f_g(Q_g)$, the relationship between Q_g and D_c was statistically analyzed for a fixed Q_c. Fig. 3(e)-3(h) suggests that a linear model

$$f_g(Q_g) = \alpha_{gc} Q_g + \beta_{gc}, \tag{9}$$

where α_{gc} and β_{gc} are model parameters is suitable.

Consequently, based on (6), (8), and (9), the color distortion can be finally written as

$$\begin{aligned} D_c &= \alpha_{gc} Q_g + \beta_{gc} + \alpha_{cc} Q_c + \beta_{cc} \\ &= \alpha_{gc} Q_g + \alpha_{cc} Q_c + \beta_c, \end{aligned} \tag{10}$$

where $\beta_c = \beta_{gc} + \beta_{cc}$. The accuracy of (10) is verified in Table II, which shows that all SCCs are about 1.00, and the average RMSE is only 0.60. Finally, (2) can be rewritten as

$$\begin{aligned} D &= \omega D_g + (1-\omega) D_c \\ &= \omega(\alpha_g Q_g + \beta_g) + (1-\omega)(\alpha_{gc} Q_g + \alpha_{cc} Q_c + \beta_c) \\ &= a Q_g + b Q_c + c, \end{aligned} \tag{11}$$

where $a = \omega\alpha_g + (1-\omega)\alpha_{gc}$, $b = (1-\omega)\alpha_{cc}$, and $c = \omega\beta_g + (1-\omega)\beta_c$. Table III shows the SCC and RMSE between the actual D and the one provided by our model. In this table, the SCC and RMSE were calculated by setting the weighting factor ω to 0.5. We can see that the average SCC and RMSE are close to 1.00 and 0.26, respectively, which indicates that (11) is an accurate model.

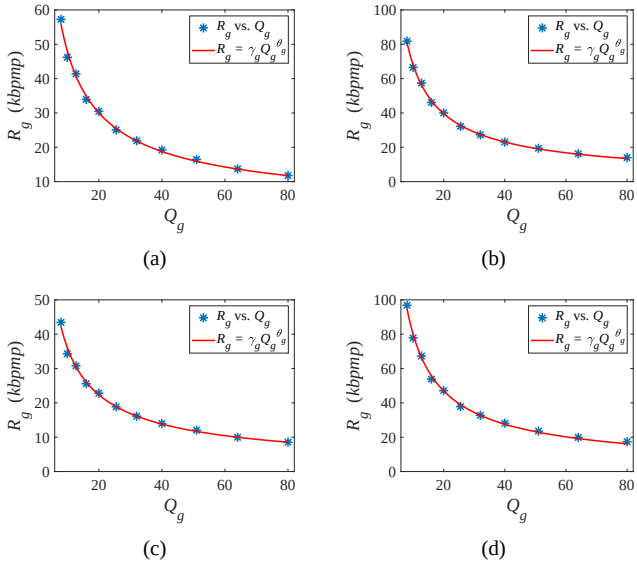


Fig. 4. Illustration of the relationship between R_g and Q_g. (a) *Loot*, (b) *Longdress*, (c) *Queen*, (d) *Redandblack*.

B. Rate Model

The total bitrate R is the sum of the geometry bitrate and color bitrate, i.e.,

$$R = R_g + R_c, \tag{12}$$

where the geometry bitrate R_g depends only on Q_g, whereas the color bitrate R_c depends on both Q_g and Q_c. For R_g, we used the Cauchy-based rate model [46]

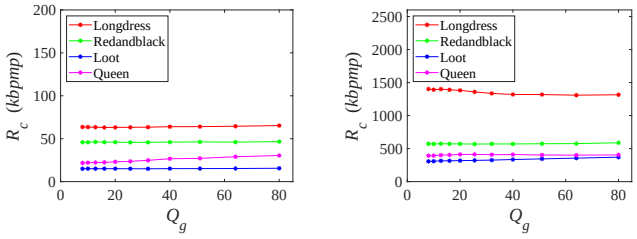
$$R_g = \gamma_g Q_g^{\theta_g}, \tag{13}$$

where γ_g and θ_g are model parameters. Because the bitrate of a 3D point cloud is relatively large, we used kilobits per million points (*kbpmp*) as the bitrate unit. Fig. 4 shows the results of statistical experiments to verify the accuracy of (13). From this figure, we can observe that the R_g model (13) is appropriate. This is confirmed by Table IV, which shows that the R-squared (R^2) value between the actual R_g and the fitted values is equal to 1.00. Because the bitrate of a 3D point cloud is typically very large, the RMSE seems to be large. Therefore, we also calculated the normalized RMSE to illustrate the fitting error effectively. The normalized RMSE ($NRMSE$) is defined as

$$NRMSE = \frac{RMSE}{Rate_{\max}}, \tag{14}$$

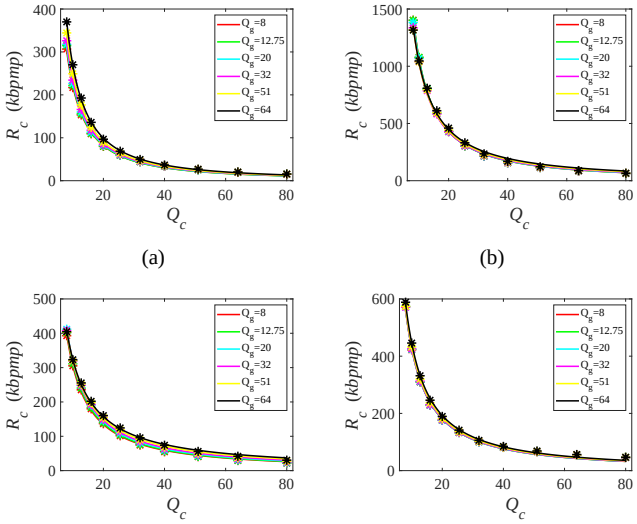
TABLE IV
VERIFICATION OF PROPOSED GEOMETRY RATE MODEL (13)

Point Cloud	R^2	RMSE (*kbpmp*)	$\text{Rate}_{\max}$ (*kbpmp*)	NRMSE
Andrew	1.00	1.16	81.70	0.01
David	1.00	1.14	69.68	0.02
Longdress	1.00	0.84	81.82	0.01
Redandblack	1.00	1.36	96.84	0.01
Loot	1.00	0.91	57.26	0.02
Queen	1.00	0.79	43.45	0.02
Soldier	1.00	1.19	64.27	0.02
Basketballplayer	1.00	0.68	44.74	0.02



(a) (b)

Fig. 5. Influence of Q_g on R_c. (a) Q_c=80, $Q_g = 8, 10, \ldots, 80$; (b) Q_c=8, $Q_g = 8, 10, \ldots, 80$.



(a) (b) (c) (d)

Fig. 6. Illustration of the relationship between R_c and Q_c ($R_c = \gamma_c Q_c^{\theta_c}$). (a) *Loot*, (b) *Longdress*, (c) *Queen*, (d) *Redandblack*.

where $Rate_{\max}$ is the maximum bitrate. Table IV shows that the NRMSE is as low as 0.01, which confirms that (13) is accurate.

To study the effect of Q_g on R_c, we compressed the color information with a fixed Q_c and multiple Q_gs. The results, shown in Fig. 5, indicate that the effect of Q_g on R_c is negligible. This is also confirmed by Fig. 6, which shows R_c as a function of Q_c for various Q_gs.

Thus, for simplicity, we can assume that R_c is only affected by Q_c. Moreover, Fig. 6 suggests that the model

$$R_c = \gamma_c Q_c^{\theta_c}, \tag{15}$$

TABLE V
VERIFICATION OF PROPOSED COLOR RATE MODEL (15)

Point Cloud	R^2	RMSE (*kbpmp*)	$\text{Rate}_{\max}$ (*kbpmp*)	NRMSE
Andrew	1.00	10.95	803.89	0.01
David	1.00	0.86	141.09	0.01
Longdress	1.00	22.40	1402.56	0.02
Redandblack	1.00	8.37	572.79	0.01
Loot	0.99	9.52	306.76	0.03
Queen	1.00	7.45	392.99	0.02
Soldier	0.99	15.31	493.52	0.03
Basketballplayer	1.00	14.66	287.94	0.02

TABLE VI
ACCURACY OF PROPOSED RATE MODEL (16)

Point Cloud	R^2	RMSE (*kbpmp*)	$\text{Rate}_{\max}$ (*kbpmp*)	NRMSE
Andrew	1.00	8.29	885.59	0.01
David	1.00	1.30	210.77	0.01
Longdress	1.00	20.15	1484.38	0.01
Redandblack	1.00	6.01	669.63	0.01
Loot	0.99	8.10	364.02	0.02
Queen	1.00	3.21	436.44	0.01
Soldier	0.99	12.04	557.79	0.02
Basketballplayer	1.00	3.71	332.68	0.01

where γ_c and θ_c are model parameters is appropriate to describe the relationship between Q_c and R_c. This is confirmed in Table V, which shows that the R^2 of the relationship between R_c and Q_c is larger than or equal to 0.99, while the NRMSE is always smaller than 0.03. Accordingly, (12) can be rewritten as

$$\begin{aligned} R &= R_g + R_c \\ &= \gamma_g Q_g^{\theta_g} + \gamma_c Q_c^{\theta_c}. \end{aligned} \tag{16}$$

Table VI validates (16) by showing that R^2 was close to 1.00, and the NRMSE was always smaller than or equal to 0.02. Finally, Fig. 7 illustrates the accuracy of models (16) and (11) by comparing the actual values to the values predicted by the models.

IV. MODEL-BASED OPTIMAL BIT ALLOCATION ALGORITHM

Based on the analysis in Section III, the optimal bit allocation problem (1) can be converted to the problem of finding the optimal solution of the constrained optimization problem

$$\begin{aligned} \min_{(Q_g, Q_c)} \quad & aQ_g + bQ_c + c \\ \text{s.t.} \quad & \gamma_g Q_g^{\theta_g} + \gamma_c Q_c^{\theta_c} \le R_T. \end{aligned} \tag{17}$$

To solve (17), we first need to determine the model parameters a, b, c, γ_g, θ_g, γ_c, and θ_c. This is done by encoding the 3D point cloud with three different pairs of quantization steps ($Q_{g,1}$,$Q_{c,1}$), ($Q_{g,2}$,$Q_{c,2}$), ($Q_{g,3}$,$Q_{c,3}$) and solving the systems of equations (18) and (19):

$$\begin{cases} D_1 &= aQ_{g,1} + bQ_{c,1} + c \\ D_2 &= aQ_{g,2} + bQ_{c,2} + c \\ D_3 &= aQ_{g,3} + bQ_{c,3} + c \end{cases} \tag{18}$$

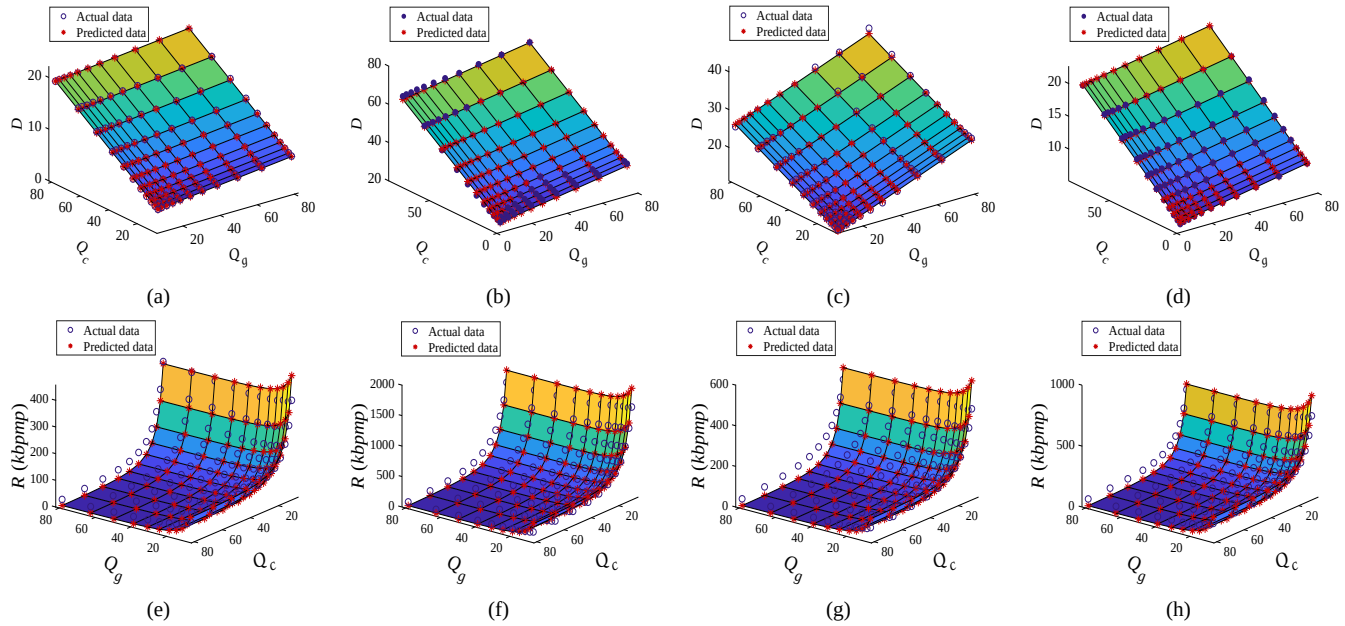


Fig. 7. Illustration of the accuracy of the rate model (16) and distortion model (11). (a)-(d): accuracy of the distortion model (11) for *Loot*, *Longdress*, *Queen*, and *Redandblack*, (e)-(f): accuracy of the rate model (16) for *Loot*, *Longdress*, *Queen*, and *Redandblack*.

$$\begin{cases} R_{g,1} &= \gamma_g Q_{g,1}^{\theta_g} \\ R_{g,2} &= \gamma_g Q_{g,2}^{\theta_g} \\ R_{c,1} &= \gamma_c Q_{c,1}^{\theta_c} \\ R_{c,2} &= \gamma_c Q_{c,2}^{\theta_c} \end{cases}, \tag{19}$$

where D_1, D_2, D_3 are the corresponding distortions and $R_{g,1}$, $R_{g,2}$, $R_{c,1}$, $R_{c,2}$ are the corresponding geometry and color bitrates, respectively. Because both the objective function and the constraint function in (17) are convex, the optimal quantization steps, $Q_{g,opt}$ and $Q_{c,opt}$ can be obtained with an interior point method called the barrier method [15]. The barrier method is simple and allows us to solve the optimization problem to a guaranteed accuracy [15]. In this method, the convex optimization problem is first converted to an unconstrained optimization problem using a logarithmic barrier function [15]:

$$\min_{(Q_g,Q_c)} (aQ_g + bQ_c + c) - \mu \log[-(\gamma_g Q_g^{\theta_g} + \gamma_c Q_c^{\theta_c} - R_T)], \tag{20}$$

where μ is the barrier parameter. The details of the barrier method are given in Algorithm 1. The output of the interior point method is subsequently rounded to obtain a solution that belongs to the finite set of discrete quantization steps used by the V-PCC coder. While rounding makes the solution practical for coding, it may lead to a slight violation of the constraint on the target bitrate.

Algorithm 1 Barrier method for the constrained optimization problem (17)

Input: a barrier parameter $\mu > 0$, a decline factor $\eta < 1$ and a desired level of accuracy $\epsilon > 0$.

Output: $(Q_{g,opt}, Q_{c,opt})$, an optimal solution to (17).

Initialization: $k = 0$, $(Q_{g,opt}, Q_{c,opt}) = (Q_g^{(k)}, Q_c^{(k)}) = (80, 80)$, $\mu^{(k)} = \mu$.

While $\mu^{(k)} \geq \epsilon$ **do**

1. compute $(Q_{g,opt}, Q_{c,opt})$ as solution to (20) using Newton's method initialized with $(Q_g, Q_c) = (Q_g^{(k)}, Q_c^{(k)})$;
2. update $k = k + 1$, $(Q_g^{(k)}, Q_c^{(k)}) = (Q_{g,opt}, Q_{c,opt})$, $\mu^{(k)} = \eta \mu^{(k-1)}$.

end While

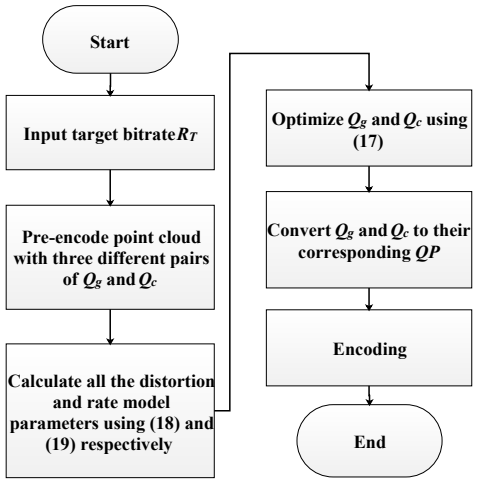


Fig. 8. Flow chart of the proposed bit allocation algorithm for geometry and color information.

Unlike exhaustive search, the proposed algorithm does not necessarily find an optimal solution to the original problem (1). This is not only due to rounding but also because our analytical rate and distortion models are only approximations. However, the experimental results (see Section V) show that the rate-distortion performance of the proposed algorithm is very close to that of exhaustive search.

The flowchart of the proposed bit allocation algorithm is shown in Fig. 8.

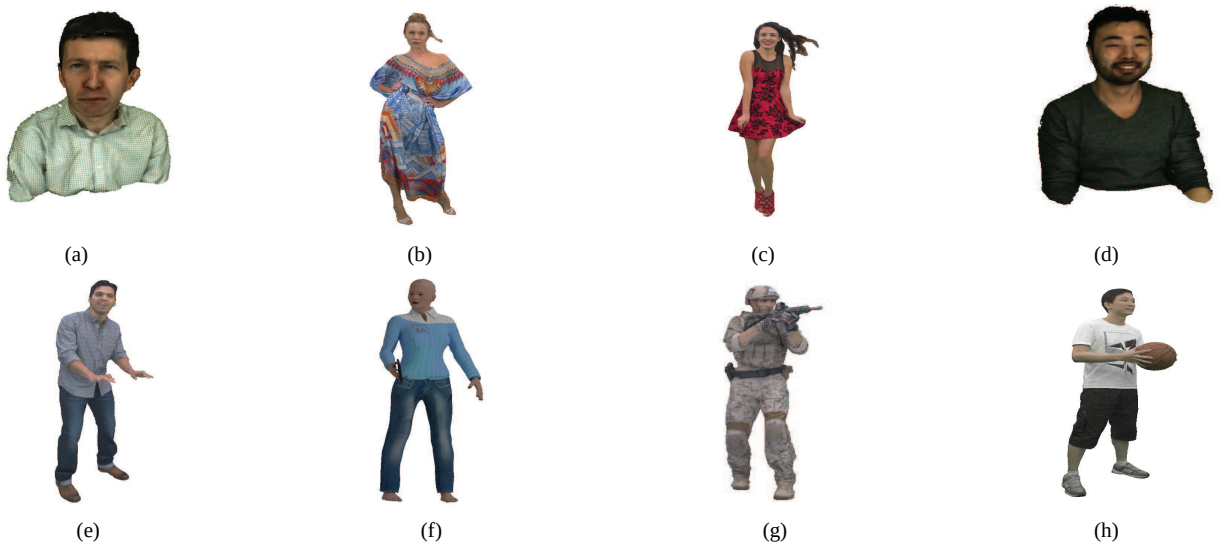


Fig. 9. 3D point cloud sequences used in the experiments. (a)*Andrew*, (b)*Longdress*, (c)*Redandblack*, (d)*David*, (e)*Loot* , (f)*Queen*, (g)*Soldier*, (h)*Basketballplayer*.

TABLE VII
DATASET

Point Cloud	Dataset	Type	Number of Frames	Default Configuration
Andrew	Microsoft Voxelized Upper Bodies	static	1	ctc-all-intra.cfg
David	Microsoft Voxelized Upper Bodies	static	1	ctc-all-intra.cfg
Longdress	Dynamic_Objects	dynamic	17	ctc-low-delay.cfg
Redandblack	Dynamic_Objects	dynamic	17	ctc-low-delay.cfg
Loot	Dynamic_Objects	dynamic	17	ctc-low-delay.cfg
Queen	Dynamic_Objects	dynamic	17	ctc-low-delay.cfg
Soldier	Dynamic_Objects	dynamic	17	ctc-low-delay.cfg
Basketballplayer	Dynamic_Objects	dynamic	17	ctc-low-delay.cfg

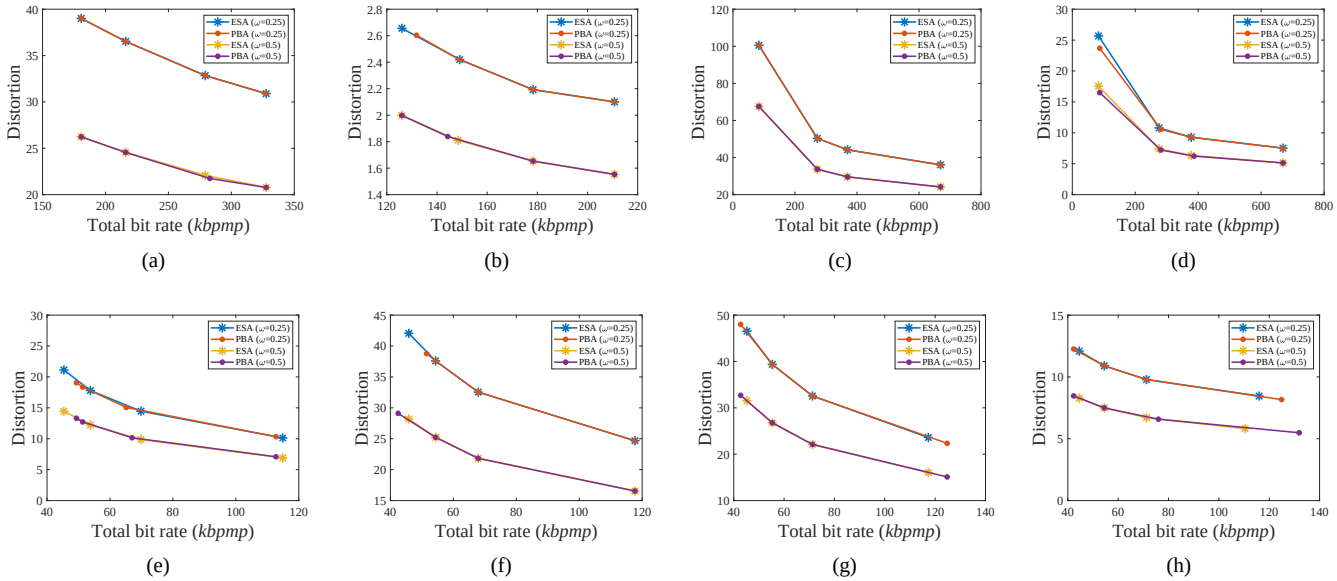


Fig. 10. R-D performance of the proposed algorithm (**PBA**) and exhaustive search (**ESA**). (a)*Andrew*, (b)*Longdress* , (c)*Redandblack*, (d)*David*, (e)*Loot*, (f)*Queen*, (g)*Soldier*, (h)*Basketballplayer*.

V. EXPERIMENTAL RESULTS

In this section, we evaluate the accuracy, rate-distortion performance, and time complexity of the proposed bit allocation algorithm. We implemented the proposed algorithm in the test model category 2 version 9.0 (TMC2V9) [11], which uses High Efficiency Video Coding Test Model Version 16.20 (HM16.20) [47] to compress the generated geometry and color video frames. The barrier parameter μ, decline factor η, and

TABLE VIII
BIT ALLOCATION ACCURACY FOR **ESA** AND **PBA** (ω=0.25) USING THE BITRATE ERROR (BE) AND THE QP ERROR (QPE).

Point Cloud	Target Bitrate ($kbpmp$)	**ESA**			**PBA**			BE(%)		ΔBE	QPE
		QP_g	QP_c	Bitrate ($kbpmp$)	QP_g	QP_c	Bitrate ($kbpmp$)	**ESA**	**PBA**		
Andrew	185	28	34	180.7	28	34	180.7	2.3	2.3	**0.0**	0
	220	26	33	216.1	26	33	216.1	1.8	1.8	**0.0**	0
	280	25	31	279.3	25	31	279.3	0.3	0.3	**0.0**	0
	335	23	30	327.7	23	30	327.7	2.2	2.2	**0.0**	0
Longdress	85	38	42	83.5	38	42	83.5	1.7	1.7	**0.0**	0
	280	26	34	272.0	26	34	272.0	2.9	2.9	**0.0**	0
	385	24	32	369.5	24	32	369.5	4.0	4.0	**0.0**	0
	675	22	28	669.3	22	28	669.3	0.8	0.8	**0.0**	0
Redandblack	85	36	40	83.4	40	38	87.3	**1.9**	2.7	**0.8**	6
	280	30	28	275.9	28	28	281.4	1.5	**0.5**	**1.0**	2
	385	26	26	377.9	26	26	377.9	1.8	1.8	**0.0**	0
	675	22	22	669.6	22	22	669.6	0.8	0.8	**0.0**	0
David	128	30	25	126.0	32	24	131.8	**1.6**	3.0	**1.4**	3
	150	31	23	149.1	31	23	149.1	0.6	0.6	**0.0**	0
	180	27	22	178.3	27	22	178.3	0.9	0.9	**0.0**	0
	212	22	22	210.8	22	22	210.8	0.6	0.6	**0.0**	0
Loot	46	36	38	45.4	40	36	49.3	**1.4**	7.3	**5.9**	6
	56	36	36	53.8	38	36	51.3	**4.0**	8.3	**4.3**	2
	72	32	34	69.9	36	34	65.2	**3.0**	9.5	**6.5**	4
	120	28	30	114.9	30	30	112.7	**4.2**	6.0	**1.8**	2
Queen	46	30	42	45.8	32	40	51.5	**0.3**	11.9	**11.6**	4
	56	30	40	54.4	30	40	54.4	2.9	2.9	**0.0**	0
	72	28	38	67.9	28	38	67.9	5.7	5.7	**0.0**	0
	120	22	34	117.7	22	34	117.7	1.9	1.9	**0.0**	0
Soldier	46	38	40	45.2	40	40	42.7	**1.7**	7.1	**5.4**	2
	56	38	38	55.3	38	38	55.3	1.2	1.2	**0.0**	0
	72	36	36	71.3	36	36	71.3	1.0	1.0	**0.0**	0
	120	34	32	117.3	30	32	124.8	**2.2**	4.0	**1.8**	4
Basketballplayer	46	34	36	44.5	36	36	42.3	**3.2**	8.1	**4.9**	2
	56	34	34	54.5	34	34	54.5	2.7	2.7	**0.0**	0
	72	32	32	71.3	32	32	71.3	1.0	1.0	**0.0**	0
	120	32	28	116.0	28	28	124.9	**3.3**	4.1	**0.8**	4
Average								**2.0**	**3.4**	**1.4**	**1.3**

level of accuracy ϵ were set to 0.1, 10^{-6}, and 10^{-10}, respectively. The quality evaluation software PC_error [48] was used to calculate the point-to-point distortion for both color and geometry. The performance of the proposed algorithm was evaluated on the eight 3D point cloud sequences [49] [50] shown in Fig. 9. The test data consist of two static point clouds and six dynamic point clouds (Table VII).

Because the color of the point clouds in Figs. 9(a)-9(c) is more diverse than that of the point clouds in Figs. 9(d)-9(h), we divided the point clouds in Fig. 9 into four groups, a complex static group (*Andrew*), a simple static group (*David*), a complex dynamic group (*Longdress* and *Redandblack*), and a simple dynamic group (*Loot*, *Queen*, *Soldier*, and *Basketballplayer*). The performance of exhaustive search was used as the benchmark to rate accuracy and time complexity. In exhaustive search, a 3D point cloud was first encoded by all the possible geometry and color quantization step pairs (ranging from 8 to 80), which correspond to QP values 22, 23, 24, ..., 42. Then the subset of admissible pairs (that is, those for which the bitrate is smaller than the target bitrate) was identified. Finally, the pair that gave the smallest distortion was selected from this subset.

In the proposed method, to derive the rate and distortion models, the point clouds were empirically pre-encoded with three geometry-color QP pairs (30, 40), (36, 30), and (38, 28). The distortion model parameters a, b and c were computed by solving (18), and the rate model parameters γ_g, θ_g, γ_c, and θ_c were obtained by solving (19). Then, given the target bitrate R_T, the optimal $Q_{g,opt}$ and $Q_{c,opt}$ were obtained by solving (17) using the interior point method.

A. Bit Allocation Accuracy of Proposed Algorithm

To evaluate the accuracy of the proposed bit allocation algorithm for the tested point clouds, we set different target bitrates according to the geometry and color characteristics of each sequence to cover different compression levels. For the complex static group, the common target bitrates were 185 kbpmp, 220 kbpmp, 280 kbpmp, and 335 kbpmp. For the simple static group, the target bitrates were 128 kbpmp, 150 kbpmp, 180 kbpmp, and 212 kbpmp. For the complex dynamic group, the target bitrates were 85 kbpmp, 280 kbpmp, 385 kbpmp, and 675 kbpmp. For the simple dynamic group,

TABLE IX
BIT ALLOCATION ACCURACY FOR **ESA** AND **PBA** (ω=0.5) USING THE BITRATE ERROR (BE) AND THE QP ERROR (QPE).

Point Cloud	Target Bitrate ($kbpmp$)	**ESA**			**PBA**			BE(%)		ΔBE	QPE
		QP_g	QP_c	Bitrate ($kbpmp$)	QP_g	QP_c	Bitrate ($kbpmp$)	**ESA**	**PBA**		
Andrew	185	28	34	180.7	28	34	180.7	2.3	2.3	**0.0**	0
	220	26	33	216.1	26	33	216.1	1.8	1.8	**0.0**	0
	280	25	31	279.3	24	31	282.9	**0.3**	1.0	**0.7**	1
	335	23	30	327.7	23	30	327.7	2.2	2.2	**0.0**	0
Longdress	85	38	42	83.5	38	42	83.5	1.7	1.7	**0.0**	0
	280	26	34	272.0	26	34	272.0	2.9	2.9	**0.0**	0
	385	24	32	369.5	24	32	369.5	4.0	4.0	**0.0**	0
	675	22	28	669.3	22	28	669.3	0.8	0.8	**0.0**	0
Redandblack	85	36	40	83.4	40	38	87.3	**1.9**	2.7	**0.8**	6
	280	30	28	275.9	28	28	281.4	1.5	**0.5**	**1.0**	2
	385	26	26	377.9	24	26	386.9	1.8	**0.5**	**1.3**	2
	675	22	22	669.6	22	22	669.6	0.8	0.8	**0.0**	0
David	128	30	25	126.0	30	25	126.0	1.6	1.6	**0.0**	0
	150	27	24	148.4	28	24	144.3	**1.1**	3.8	**2.7**	1
	180	27	22	178.3	27	22	178.3	0.9	0.9	**0.0**	0
	212	22	22	210.8	22	22	210.8	0.6	0.6	**0.0**	0
Loot	46	36	38	45.4	40	36	49.3	**1.4**	7.3	**5.9**	6
	56	36	36	53.8	38	36	51.3	**4.0**	8.3	**4.3**	2
	72	32	34	69.9	34	34	67.0	**3.0**	6.9	**3.9**	2
	120	28	30	114.9	30	30	112.7	**4.2**	6.0	**1.8**	2
Queen	46	30	42	45.8	32	42	42.5	**0.3**	7.7	**7.4**	2
	56	30	40	54.4	30	40	54.4	2.9	2.9	**0.0**	0
	72	28	38	67.9	28	38	67.9	5.7	5.7	**0.0**	0
	120	22	34	117.7	22	34	117.7	1.9	1.9	**0.0**	0
Soldier	46	38	40	45.2	40	40	42.7	**1.7**	7.1	**5.4**	2
	56	38	38	55.3	38	38	55.3	1.2	1.2	**0.0**	0
	72	36	36	71.3	36	36	71.3	1.0	1.0	**0.0**	0
	120	34	32	117.3	30	32	124.8	**2.2**	4.0	**1.8**	4
Basketballplayer	46	34	36	44.5	36	36	42.3	**3.2**	8.1	**4.9**	2
	56	34	34	54.5	34	34	54.5	2.7	2.7	**0.0**	0
	72	32	32	71.3	30	32	76.0	**1.0**	5.6	**4.6**	2
	120	24	30	110.4	26	28	131.9	**8.0**	9.9	**1.9**	4
Average								**2.2**	**3.6**	**1.5**	**1.3**

the target bitrates were 46 kbpmp, 56 kbpmp, 72 kbpmp, and 120 kbpmp. In practical applications, the target bitrate for geometry and color can be obtained by subtracting the bitrate of the occupancy map and the auxiliary information, which can be obtained by pre-encoding. To evaluate the bit allocation accuracy, we used the bitrate error (BE), defined as

$$BE = \frac{|B_{actual} - B_{target}|}{B_{target}} \times 100\%, \tag{21}$$

where B_{actual} is the actual bitrate and B_{target} represents the target bitrate. The lower the BE, the more accurate the algorithm. Because the proposed bit allocation algorithm allocates the bits for geometry and color components by selecting the QPs, QP error (QPE) was also used to measure the performance as follows:

$$QPE = |QP_{g,PBA} - QP_{g,ESA}| + |QP_{c,PBA} - QP_{c,ESA}|, \tag{22}$$

where $QP_{g,PBA}$ and $QP_{c,PBA}$ denote the geometry and color QPs obtained from the proposed algorithm, and $QP_{g,ESA}$ and $QP_{c,ESA}$ represent the geometry and color QPs obtained from exhaustive search.

Table VIII and Table IX show the BE and QPE of the proposed bit allocation algorithm (**PBA**) and exhaustive search (**ESA**) for different values of ω (0.25 and 0.5). Note that **ESA** produces an optimal solution but has a much higher computational cost. As shown in Table VIII and Table IX, the BE of **ESA** was as small as 0.3, while its average was 2.0% and 2.2% when ω was set to 0.25 and 0.5, respectively. For **PBA**, BE was also as low as 0.3 when ω was set to 0.25, while its average was about 3.4% and 3.6% when ω was set to 0.25 and 0.5, respectively. The average absolute difference in BE between **ESA** and **PBA** was only 1.4% and 1.5% for $\omega = 0.25$ and 0.5, respectively. On the other hand, the average QPE was only 1.3 for $\omega = 0.25$ and 0.5. In 57.8% of the cases, our algorithm (**PBA**) found the same solution as exhaustive search (**ESA**), as illustrated by the data shown in magenta in the tables, so the BE of the two algorithms was the same. In the remaining cases, our algorithm found a suboptimal solution (see the discussion at the end of Section

TABLE X
RATE-PSNR PERFORMANCE OF **ESA** AND **PBA**

Point Cloud	Target Bitrate ($kbpmp$)	**ESA**				**PBA**				BD-PSNR (dB)	
		Bitrate ($kbpmp$)		PSNR (dB)		Bitrate ($kbpmp$)		PSNR (dB)			
		$\omega=0.25$	$\omega=0.5$	$\omega=0.25$	$\omega=0.5$	$\omega=0.25$	$\omega=0.5$	$\omega=0.25$	$\omega=0.5$	$\omega=0.25$	$\omega=0.5$
Andrew	185	181	181	32.2	34.0	181	181	32.2	34.0		
	220	216	216	32.5	34.3	216	216	32.5	34.3		
	280	279	279	33.0	34.7	279	283	33.0	34.8	**0.0**	**0.0**
	335	328	328	33.3	35.0	328	328	33.3	35.0		
Longdress	85	84	84	28.1	29.9	84	84	28.1	29.9		
	280	272	272	31.1	32.9	272	272	31.1	32.9		
	385	370	370	31.7	33.4	370	370	31.7	33.4	**0.0**	**0.0**
	675	669	669	32.6	34.3	669	669	32.6	34.3		
Redandblack	85	83	83	34.1	35.8	87	87	34.5	36.2		
	280	276	276	37.9	39.6	281	281	37.9	39.7		
	385	378	378	38.5	40.3	378	387	38.5	40.3	**0.1**	**0.1**
	675	670	670	39.4	41.2	670	670	39.4	41.2		
David	128	126	126	44.2	45.9	132	126	44.3	45.9		
	150	149	148	44.6	46.3	149	144	44.6	46.3		
	180	178	178	45.0	46.8	178	178	45.0	46.8	**0.0**	**0.0**
	212	211	211	45.1	46.9	211	211	45.1	46.9		
Loot	46	45	45	34.9	36.7	49	49	35.4	37.2		
	56	54	54	35.7	37.5	51	51	35.6	37.3		
	72	70	70	36.6	38.3	65	67	36.4	38.2	**0.1**	**0.1**
	120	115	115	38.1	39.9	113	113	38.0	39.8		
Queen	46	46	46	31.9	33.7	51	42	32.3	33.6		
	56	54	54	32.4	34.2	54	54	32.4	34.2		
	72	68	68	33.0	34.8	68	68	33.0	34.8	**0.0**	**0.0**
	120	118	118	34.2	36.0	118	118	34.2	36.0		
Soldier	46	45	45	31.5	33.2	43	43	31.4	33.1		
	56	55	55	32.2	34.0	55	55	32.2	34.0		
	72	71	71	33.0	34.8	71	71	33.0	34.8	**0.0**	**0.0**
	120	117	117	34.4	36.2	125	125	34.7	36.4		
Basketballplayer	46	45	45	37.4	39.1	42	42	37.3	39.1		
	56	54	54	37.8	39.6	54	54	37.8	39.6		
	72	71	71	38.3	40.0	71	76	38.3	40.1	**0.0**	**0.0**
	120	116	110	38.9	40.6	125	132	39.1	40.9		
Average										**0.0**	**0.0**

III), which had a higher BE in 37.5% of the cases and a lower one in 4.7% of the cases.

B. Rate-Distortion Performance

In addition to bit allocation accuracy, the rate-distortion performance should also be taken into account. After determining the coding parameters with **ESA** and **PBA**, we compressed the point clouds and computed their geometry and color distortions using the PC_error reference software [48].

Fig. 10 shows the distortion $D = \omega D_g + (1-\omega)D_c$ as a function of the bitrate (in $kbpmp$) for **ESA** and **PBA**. We can see that the rate-distortion performance of the proposed algorithm was very close to that of exhaustive search.

In addition to the distortions (which are based on the MSE), we also computed the peak-signal-to-noise ratio (PSNR). In general, it is necessary to normalize with respect to the peak value when converting MSE into $PSNR$. However, the peak values of geometry and color are completely different. To calculate a meaningful $PSNR$ for the reconstructed 3D point cloud, the geometry and color values were both normalized to [0,1]. Hence, the $PSNR$ of the reconstructed 3D point cloud was calculated as:

$$PSNR = 10\log_{10}\left[\frac{1}{NMSE(NMSE_g, NMSE_c)}\right], \tag{23}$$

where $NMSE_g$ and $NMSE_c$ are the normalized geometry and color (Y channel) distortion (i.e., D_g and D_c), respectively and $NMSE(NMSE_g, NMSE_c) = \omega NMSE_g + (1-\omega)NMSE_c$.

Table X shows the PSNR of the two algorithms and the Bjøntegaard delta (BD)-PSNR [51] between their rate-PSNR curves. The results show that the performance of **PBA** is similar to that **ESA**. Interestingly, a 0.1 dB BD-PSNR gain was achieved by **PBA** for $Redandblack$ and $Loot$. The main reason is that **ESA** is optimal for the distortion $D = \omega D_g + (1-\omega)D_c$ but not necessarily optimal for the PSNR (23).

TABLE XI
COMPLEXITY COMPARISON FOR **ESA** AND **PBA**

Point Cloud	Encoding Time (s)		CQ(%)
	ESA	**PBA**	
Andrew	42304.35	285.25	0.67
David	45379.97	360.86	0.80
Longdress	798817.86	5199.87	0.65
Redandblack	936829.38	6120.76	0.65
Loot	954849.12	6196.04	0.65
Queen	1152024.40	7638.26	0.66
Soldier	1405917.34	9334.95	0.66
Basketballplayer	2533011.95	14436.01	0.57
Average			**0.66**

C. Complexity Comparison

We run the experiments on a PC with a 3.40 GHz Intel Core i7 Processor and 8.00 GB RAM and used the encoding time to evaluate the time complexity. The ratio between the encoding time of **PBA** and that of **ESA** was used to define the complexity quotient (CQ) as

$$CQ = \frac{T_{PBA}}{T_{ESA}} \times 100\%, \tag{24}$$

where T_{PBA} and T_{ESA} denote the encoding time of **PBA** and **ESA**, respectively. The time complexity of **ESA** and **PBA** mainly depends on the pre-encoding times. **ESA** needs to pre-encode the 3D point cloud for all possible combinations of QPs for the geometry and color components. Because both the geometry and color QP search range was [22, 42] with a search step size of 1, a 3D point cloud needs to be encoded $21 \times 21 = 441$ times to find the optimal Q_g and Q_c with **ESA**. In contrast, only three pre-encodings were required by **PBA** to compute the model parameters. As the time complexity of the interior point method is very small compared to the pre-encoding procedure (for example the time spent to obtain the optimal Q_g and Q_c by the interior point method was only 1.42 s for the $Longdress$ point cloud, while the pre-encoding required 5199.87 s), on average, the time complexity of **PBA** was only 0.66% of that of **ESA**, as shown in Table XI.

VI. CONCLUSION

This paper presented a model-based joint bit allocation algorithm for the V-PCC encoder. To reduce the time complexity of exhaustive search as well as preserve its rate-distortion performance, we first derived rate and distortion models for point clouds through theoretical analysis and statistical validation. Based on the derived rate and distortion models, the optimal bit allocation problem was formulated as a convex constrained optimization problem and solved by an interior point method. Model parameters were calculated by pre-encoding a 3D point cloud only three times. Experimental results showed that the bit allocation accuracy and the rate-distortion performance of the PBA were very close to those of exhaustive search at only 0.66% of its computational cost. As future work, we plan to use our rate and distortion models to develop rate control algorithm for 3D point clouds.

APPENDIX A

Based on (4), $e_c^{B,A}$ of the original point cloud A and its corresponding reconstructed point cloud B can be rewritten as

$$e_c^{B,A} = \frac{1}{|B|} \sum_{j=1}^{|B|} \left| \boldsymbol{C}_{v_j} - \boldsymbol{C}_{v_j^*} \right|^2, \tag{25}$$

where $\boldsymbol{C}_{v_j}$ denotes the color of a point v_j in the original point cloud A, v_j^* is the nearest neighbor of v_j in the reconstructed point cloud B, and $\boldsymbol{C}_{v_j^*}$ is the color of v_j^*. From the law of large numbers [52], the average of the reconstruction error is close to its expected value. Thus, (25) can be written as

$$e_c^{B,A} \approx E\{(\boldsymbol{C}_v - \boldsymbol{C}_{v^*})^2\}, \tag{26}$$

where $E\{.\}$ denotes the expectation operator, $\boldsymbol{C}_v$ is the random variable corresponding to the color of point v in A, and $\boldsymbol{C}_{v^*}$ is the random variable corresponding to the color of its nearest neighbor v^* in B. In the V-PCC encoder, the color information of the points in the original point cloud is first reassigned (recoloring procedure) based on the reconstructed geometry information [11] due to the number of point changes after reconstructing the distorted geometric information. Then the reassigned color information is compressed to obtain the final point cloud with compressed geometry and color information. Let $\boldsymbol{C}_{v^g}$ denote the reassigned color of the point in the reconstructed geometry point cloud, whose color information is not compressed while the geometry information has been compressed. Then (26) can be rewritten as:

$$\begin{aligned} e_c^{B,A} &\approx E\{(\boldsymbol{C}_v - \boldsymbol{C}_{v^g} + \boldsymbol{C}_{v^g} - \boldsymbol{C}_{v^*})^2\} \\ &= E\{(\boldsymbol{C}_v - \boldsymbol{C}_{v^g})^2\} + E\{(\boldsymbol{C}_{v^g} - \boldsymbol{C}_{v^*})^2\} \\ &\quad + 2E\{(\boldsymbol{C}_v - \boldsymbol{C}_{v^g})(\boldsymbol{C}_{v^g} - \boldsymbol{C}_{v^*})\}, \end{aligned} \tag{27}$$

where $E\{(\boldsymbol{C}_v - \boldsymbol{C}_{v^g})^2\}$ represents the color distortion induced only by Q_g, and $E\{(\boldsymbol{C}_{v^g} - \boldsymbol{C}_{v^*})^2\}$ represents the color distortion induced only by Q_c.

From the definition of $\boldsymbol{C}_v$, $\boldsymbol{C}_{v^g}$, and $\boldsymbol{C}_{v^*}$, we can conclude that the difference between $\boldsymbol{C}_v$ and $\boldsymbol{C}_{v^g}$ depends only on the geometry quantization, whereas the difference between $\boldsymbol{C}_{v^g}$ and $\boldsymbol{C}_{v^*}$ depends only on the color quantization. Thus, it is reasonable to assume that these two random variables are independent (see [53] [54] for a similar assumption in the context of 3D video coding). Consequently, $E\{(\boldsymbol{C}_v - \boldsymbol{C}_{v^g})(\boldsymbol{C}_{v^g} - \boldsymbol{C}_{v^*})\} = E\{(\boldsymbol{C}_v - \boldsymbol{C}_{v^g})\}E\{(\boldsymbol{C}_{v^g} - \boldsymbol{C}_{v^*})\}$. Since it is also reasonable to assume that $E\{\boldsymbol{C}_v\} \approx E\{\boldsymbol{C}_{v^g}\}$ and $E\{\boldsymbol{C}_{v^g}\} \approx E\{\boldsymbol{C}_{v^*}\}$, we can conclude that $E\{(\boldsymbol{C}_v - \boldsymbol{C}_{v^g})(\boldsymbol{C}_{v^g} - \boldsymbol{C}_{v^*})\} \approx 0$. Hence, (27) can be written as

$$e_c^{B,A} \approx f_g^{B,A}(Q_g) + f_c^{B,A}(Q_c), \tag{28}$$

where $f_g^{B,A}(Q_g) = E\{(\boldsymbol{C}_v - \boldsymbol{C}_{v^g})^2\}$ and $f_c^{B,A}(Q_c) = E\{(\boldsymbol{C}_{v^g} - \boldsymbol{C}_{v^*})^2\}$.

REFERENCES

[1] Y. Cui, S. Schuon, S. Thrun, D. Stricker, and C. Theobalt, "Algorithms for 3D shape scanning with a depth camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1039–1050, May 2013.

[2] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 1–4.

[3] E. Vlachos, A. S. Lalos, A. Spathis-Papadiotis, and K. Moustakas, "Distributed consolidation of highly incomplete dynamic point clouds based on rank minimization," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3276–3288, Dec 2018.

[4] J. Chen, C. Lin, P. Hsu, and C. Chen, "Point cloud encoding for 3D building model retrieval," *IEEE Transactions on Multimedia*, vol. 16, no. 2, pp. 337–345, Feb 2014.

[5] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.

[6] W. Oropallo, L. A. Piegl, P. Rosen, and K. Rajab, "Point cloud slicing for 3-D printing," *Computer-Aided Design and Applications*, vol. 15, no. 1, pp. 90–97, 2018.

[7] Y. Park, V. Lepetit, and W. Woo, "Multiple 3d object tracking for augmented reality," in *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, 2008, pp. 117–120.

[8] Q. Liu, H. Yuan, J. Hou, H. Liu, and R. Hamzaoui, "Model-based encoding parameter optimization for 3D point cloud compression," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2018, pp. 1981–1986.

[9] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li, *et al.*, "Emerging mpeg standards for point cloud compression," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 133–148, 2018.

[10] P. A. Chou, O. Nakagami, and y. Euee S. Jang, journal=ISO/IEC JTC1/SC29/WG11 MPEG N17340, "Point cloud compression test model for category 1 v1."

[11] 3DG, "V-PCC test model v9," *ISO/IEC JTC1/SC29/WG11 MPEG, W19085*, 2020.

[12] M. Khaled, "PCC test model category 3 v1," *ISO/IEC JTC1/SC29/WG11 MPEG N17349*, 2018.

[13] 3DG, "G-PCC codec description v5," *ISO/IEC JTC1/SC29/WG11 MPEG N18891*, 2019.

[14] H. Liu, H. Yuan, Q. Liu, J. Hou, and J. Liu, "A comprehensive study and comparison of core technologies for MPEG 3D point cloud compression," *IEEE Transactions on Broadcasting*, 2019, doi: 10.1109/TBC.2019.2957652, to be appeared.

[15] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University press, 2004.

[16] 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)*. IEEE, 2016, pp. 141–150.

[17] C. Zhang, D. Florencio, and C. Loop, "Point cloud attribute compression with graph transform," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 2066–2070.

[18] 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)*. IEEE, 2016, pp. 1374–1378.

[19] P. de Oliveira Rente, C. Brites, J. Ascenso, and F. Pereira, "Graph-based static 3D point clouds geometry coding," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 284–299, Feb 2019.

[20] S. Gu, J. Hou, H. Zeng, and H. Yuan, "3D point cloud attribute compression via graph prediction," *IEEE Signal Processing Letters*, vol. 27, pp. 176–180, 2020.

[21] R. L. de Queiroz and P. A. Chou, "Compression of 3D point clouds using a region-adaptive hierarchical transform," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3947–3956, 2016.

[22] G. P. Sandri, P. A. Chou, M. Krivokuća, and R. L. de Queiroz, "Integer alternative for the region-adaptive hierarchical transform," *IEEE Signal Processing Letters*, vol. 26, no. 9, pp. 1369–1372, 2019.

[23] D. Flynn and S. Lasserre, "Report on upsampled transform domain prediction in RAHT," *ISO/IEC JTC1/SC29/WG11 MPEG M49380*, 2019, Jul.

[24] M. Khaled, A. Tourapis, J. Kim, F. Robinet, V. Valentin, and Y. Su, "Lifting scheme for lossy attribute encoding in TMC1," *ISO/IEC JTC1/SC29/WG11 MPEG M42640*, 2018, Apr.

[25] P. A. Chou and R. L. de Queiroz, "Gaussian process transforms," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1524–1528.

[26] R. L. de Queiroz and P. A. Chou, "Transform coding for point clouds using a gaussian process model," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3507–3517, 2017.

[27] 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*, vol. 29, pp. 796–808, 2019.

[28] J. Hou, L.-P. Chau, Y. He, and P. A. Chou, "Sparse representation for colors of 3D point cloud via virtual adaptive sampling," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2926–2930.

[29] J. Hou, "Permuted sparse representation for 3D point clouds," *IEEE Signal Processing Letters*, vol. 26, no. 12, pp. 1847–1851, 2019.

[30] Y. Shao, Q. Zhang, G. Li, and Z. Li, "Hybrid point cloud attribute compression using slice-based layered structure and block-based intra prediction," *arXiv preprint arXiv:1804.10783*, 2018.

[31] 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*, vol. 27, no. 4, pp. 828–842, 2016.

[32] C. Tu, E. Takeuchi, C. Miyajima, and K. Takeda, "Compressing continuous point cloud data using image compression methods," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 1712–1719.

[33] J. Hou, L. Chau, N. Magnenat-Thalmann, and Y. He, "Compressing 3-D human motions via keyframe-based geometry videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 1, pp. 51–62, Jan 2015.

[34] J. Hou, L. Chau, M. Zhang, N. Magnenat-Thalmann, and Y. He, "A highly efficient compression framework for time-varying 3-D facial expressions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 9, pp. 1541–1553, Sep. 2014.

[35] J. Hou, L. Chau, Y. He, M. Zhang, and N. Magnenat-Thalmann, "Rate-distortion model based bit allocation for 3-D facial compression using geometry video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 9, pp. 1537–1541, Sep. 2013.

[36] J. Xia, D. T. P. Quynh, Y. He, X. Chen, and S. C. H. Hoi, "Modeling and compressing 3-D facial expressions using geometry videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 1, pp. 77–90, Jan 2012.

[37] X. Gu, S. J. Gortler, and H. Hoppe, "Geometry images," in *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3. ACM, 2002, pp. 355–361.

[38] L. Li, Z. Li, V. Zakharchenko, J. Chen, and H. Li, "Advanced 3D motion prediction for video-based dynamic point cloud compression," *IEEE Transactions on Image Processing*, vol. 29, pp. 289–302, 2019.

[39] D. Thanou, P. A. Chou, and P. Frossard, "Graph-based compression of dynamic 3D point cloud sequences," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1765–1778, 2016.

[40] R. L. de Queiroz and P. A. Chou, "Motion-compensated compression of dynamic voxelized point clouds," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3886–3895, 2017.

[41] A. Anis, P. A. Chou, and A. Ortega, "Compression of dynamic 3D point clouds using subdivisional meshes and graph wavelet transforms," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6360–6364.

[42] MPEG. (2017) Test code PCC in 3DG. [Online]. Available: https://github.com/RufaelDev/pcc-mp3dg

[43] C. T. P. C. Rufael Mekuria, Zhu Li, "Evaluation criteria for pcc (point cloud compression)," *ISO/IEC JTC1/SC29/WG11 MPEG, N16332*, 2016.

[44] 3DG, "V-PCC test model v9," *ISO/IEC JTC1/SC29/WG11 MPEG N19085*, 2020.

[45] R. Mekuria, S. Laserre, and C. Tulvan, "Performance assessment of point cloud compression," in *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2017, pp. 1–4.

[46] F. Cen, Q. Lu, and W. Xu, "Efficient rate control for intra-frame coding in high efficiency video coding," in *2014 International Conference on Signal Processing and Multimedia Applications (SIGMAP)*. IEEE, 2014, pp. 54–59.

[47] JCT-VC. (2020) HEVC test model. [Online]. Available: Available: https: //hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware

[48] S. Schwarz and D. Flynn, "Common test conditions for point cloud compression," *ISO/IEC JTC1/SC29/WG11 MPEG, N18175*, 2019.

[49] MPEGC. (2020) MPEG point cloud datasets-cfp. [Online]. Available: Available: http://mpegfs.int-evry.fr/MPEG/PCC/DataSets/pointCloud/CfP/datasets

[50] ——. (2020) MPEG point cloud datasets-AnimatedPC-Microsoft. [Online]. Available: Available: http://mpegfs.int-evry.fr/MPEG/PCC/DataSets/AnimatedPC-Microsoft

[51] G. Bjøntegaard, "Calculation of average PSNR differences between RD curves," *ITU-T SC16/Q6, VCEG-M33*, 2001, Apr.

[52] S. Idele, "The law of large numbers: Some issues of interpretation and application for beginners," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 28, no. 3, pp. 209–217, 1979.
[53] H. Yuan, Y. Chang, J. Huo, F. Yang, and Z. Lu, "Model-based joint bit allocation between texture videos and depth maps for 3-D video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 4, pp. 485–497, 2011.
[54] L. Fang, N.-M. Cheung, D. Tian, A. Vetro, H. Sun, and O. C. Au, "An analytical model for synthesis distortion estimation in 3d video," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 185–199, 2014.

Qi Liu received the B.S. degree from Shandong Technology and Business University, Shandong, China, in 2011 and the M.S. degrees from the School of Telecommunication Engineering, Xidian University, Xi'an, China, in 2014. Currently, she is pursuing the Ph.D degree with the Information Science and Engineering from Shandong University, Shandong, China. Her research interests include point clouds coding and point clouds processing.

Hui Yuan (S'08-M'12-SM'17) received the B.E. and Ph.D. degree in telecommunication engineering from Xidian University, Xi'an, China, in 2006 and 2011, respectively. From 2011.04 to now, he works as a lecturer (2011.04-2014.12), an associate Professor (2015.01-2016.08), and a full professor (2016.09-), at Shandong University (SDU), Jinan, China. From 2013.01-2014.12, and 2017.11-2018.02, he also worked as a post-doctor fellow (Granted by the Hong Kong Scholar Project) and a research fellow, respectively, with the department of computer science, City University of Hong Kong (CityU). His current research interests include video/image/immersive media processing, compression, adaptive streaming, and computer vision, etc.

Junhui Hou received the B.Eng. degree in information engineering (Talented Students Program) from the South China University of Technology, Guangzhou, China, in 2009, the M.Eng. degree in signal and information processing from Northwestern Polytechnical University, Xian, China, in 2012, and the Ph.D. degree in electrical and electronic engineering from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2016. He has been an Assistant Professor with the Department of Computer Science, City University of Hong Kong, since 2017. His research interests fall into the general areas of visual computing, such as image/video/3D geometry data representation, processing and analysis, semi/un-supervised data modeling, and data compression and adaptive transmission

Dr. Hou was the recipient of several prestigious awards, including the Chinese Government Award for Outstanding Students Study Abroad from China Scholarship Council in 2015, and the Early Career Award (3/381) from the Hong Kong Research Grants Council in 2018. He is currently serving as an Associate Editor for IEEE Transactions on Circuits and Systems for Video Technology, The Visual Computer, an Area Editor for Signal Processing: Image Communication, the Guest Editor for the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. He also served an Area Chair of ACM International Conference on Multimedia (ACM MM) 2019 and 2020, and IEEE International Conference on Multimedia & Expo (IEEE ICME) 2020. He is a senior member of IEEE.

Raouf Hamzaoui (M02CSM07) received the M.Sc. degree in mathematics from University of Montreal, Montral, QC, Canada, in 1993; the Dr.rer.nat. degree from University of Freiburg, Freiburg im Breisgau, Germany, in 1997; and the Habilitation degree in computer science from University of Konstanz, Konstanz, Germany, in 2004. He was an Assistant Professor with the Department of Computer Science, University of Leipzig, Leipzig, Germany, and with the Department of Computer and Information Science, University of Konstanz. He joined De Montfort University, Leicester, U.K., in 2006, where he is currently a Professor in Media Technology. His research interests include image and video coding, multimedia communication systems, error control systems, and machine learning.

Honglei Su (M'19) received the B.A.Sc degree from Shandong University of Science and Technology, Qingdao, China in 2008 and the Ph.D. degree from Xidian University, Xi'an, China in 2014. From 2014.09 to now, he works as an Assistant Professor with the School of Electronic Information, Qingdao University, Qingdao, China. From 2018.03-2019.03, he also worked as a visiting scholar with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests include perceptual image processing, immersive media processing, computer vision, etc.