A Comparative Measurement Study of Point Cloud-Based Volumetric Video Codecs

Mengyu Yang, Zhenxiao Luo, Miao Hu, *Member, IEEE*, Min Chen, *Fellow, IEEE*, and Di Wu, *Senior Member, IEEE*

***Abstract*—Volumetric video is fully three-dimensional and immersive, which shows great potential in various application scenarios. However, the naive delivery of volumetric video over the Internet requires a huge amount of bandwidth and compute resources. A volumetric video should be compressed by a specific codec before transmission. In spite of the existence of a few pioneering efforts on prototype design, there is very limited work to investigate the efficiency and practicality of existing volumetric video codecs. In this paper, we conduct a comparative measurement study on five representative point cloud-based volumetric video codecs, including 3 conventional codecs (e.g., Draco, G-PCC, V-PCC) and 2 neural-based codecs (e.g., PU-GCN+, MPU+), on six real volumetric video datasets. We investigate the applicability of the above codecs in different application scenarios, and examine how the features of point cloud (e.g., quality, texture, geometric and scene complexity) affect the quality of compressed volumetric video from multiple perspectives. In addition, we also study the impact of users' viewing behaviors on the coding and delivery efficiency with Quality of Experience (QoE) metrics. Our results shed a number of important insights, which provide useful guidelines for optimizing the design of future volumetric video streaming systems.**

***Index Terms*—Volumetric video, point cloud compression, super-resolution.**

I. INTRODUCTION

DIFFERENT from 2D video and traditional 360-degree video, volumetric video enables users to watch a video with six-degrees-of-freedom (6DoF). Such feature of volumetric video has attracted tremendous attention from a number of fields, such as education, entertainment, and so on [1], [2], [3], [4]. Generally, a volumetric video can be represented by either *point cloud* [5], [6] or *polygon mesh* [7].

Manuscript received 18 November 2022; revised 20 January 2023; accepted 1 February 2023. Date of publication 22 February 2023; date of current version 6 September 2023. This work was supported in part by the National Natural Science Foundation of China under Grant U1911201, Grant U2001209, and Grant 62072486; in part by the Science and Technology Planning Project of Guangdong Province under Grant 2021A0505110008; and in part by the Natural Science Foundation of Guangdong under Grant 2021A1515011369. *(Corresponding author: Di Wu.)*

Mengyu Yang, Zhenxiao Luo, Miao Hu, and Di Wu are with the School of Computer Science and Engineering and the Guangdong Key Laboratory of Big Data Analysis and Processing, Sun Yat-sen University, Guangzhou 510006, Guangdong, China (e-mail: yangmy36@mail2.sysu.edu.cn; luozhx6@mail2.sysu.edu.cn; humiao5@mail.sysu.edu.cn; wudi27@mail.sysu.edu.cn).

Min Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, Guangdong, China (e-mail: minchen@ieee.org).

Digital Object Identifier 10.1109/TBC.2023.3243407

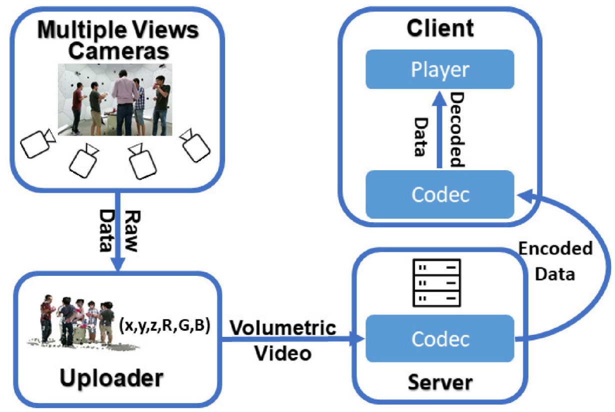


Fig. 1. Illustration of a typical volumetric video streaming system.

Due to the advantages of easy acquisition and high flexibility, point cloud has been adopted as the primary format of volumetric video. However, the transmission of point clouds is extremely challenging in today's Internet. For example, a point cloud for entertainment typically contains over one million points per frame. If not compressed, a total bandwidth of 3.6 Gbps is required to achieve a playback at 30 frames per second (FPS) [8]. Therefore, it is *a must* to perform deep compression over point clouds before transmission.

Commonly, a typical volumetric video streaming system can be illustrated by Fig. 1, in which the codec is responsible for compression and decompression of volumetric video content (e.g., point cloud). The codecs in the existing point cloud-based volumetric video streaming systems (e.g., ViVo [9], Groot [10], Yuzu [11]) can be divided into two main categories, namely, *conventional codecs* (e.g., V-PCC [8], G-PCC [8], Draco [12]) and *neural-based codecs* (e.g., MPU [13], PU-GCN [14]).

Conventional codecs use the data projection technique to represent a point cloud with a 2D projection structure or a 3D-tree structure. For example, V-PCC projects 3D models into a 2D space and compresses them with 2D video compression methods. Draco and G-PCC use 3D-tree structures, such as octree [15] and kd-tree [16], to compress point clouds. Instead, neural-based codecs employ the point cloud super-resolution method to compress a point cloud. With the rapid progress of deep learning techniques, especially deep learning-based super-resolution (SR) [13], [14], [17], neural-based codecs have also attracted tremendous attention in recent years.

Although quite a few volumetric streaming systems have been built, there is very limited work to study the efficiency and practicality of different volumetric video codecs. A few important questions need to be further investigated: *First*, it is desirable to clearly understand the application scenarios of different codecs. For example, if the encoding and decoding operation of a codec takes too much time, such a codec is not suitable for the scenario of live volumetric video streaming. Thus, we need to comprehensively measure the performance of these codecs in terms of compression ratio, speed, and video quality. *Second*, it is unclear how the features (e.g., video quality, texture and geometric complexity) of a volumetric video content affect the codec performance and delivery efficiency. *Third*, user viewing behaviors will dramatically change the Quality of Experience (QoE) of volumetric videos. It is expected to investigate the performance of volumetric video codecs with real user viewing traces and understand how user viewing behaviors affect the performance of volumetric video codecs.

In this paper, we are among the first to systematically measure 5 representative point cloud-based volumetric video codecs from multiple perspectives (e.g., compression ratio, encoding/decoding time, geometric distance, color-PSNR, QoE, etc). We use 6 real volumetric video datasets with different characteristics and examine the applicability of the above codecs in different application scenarios. We also study how the features of volumetric videos and user viewing behaviors affect the performance of different codecs.

Overall, our main observations and their implications can be summarized as below:

- A simple implementation of all existing codecs can hardly meet the latency requirement of live volumetric video streaming. It is required to conduct more performance-oriented optimization for codecs. For on-demand volumetric video streaming, G-PCC is more suitable for the delivery of high-quality point clouds, while Draco outperforms other codecs if the size of a point cloud is small or there is sufficient bandwidth.
- The texture and scene complexity as well as the noise level of point clouds have significant effects on the compression performance of codecs. On the contrary, the geometric complexity has little impact. We also observe that different codecs show diverse sensitivity to different features, revealing the necessity to configure codecs according to the attributes of a point cloud.
- A user's viewing behavior greatly affects the delivery performance of volumetric videos. Especially for videos with complex scenes (e.g., multiplayer scenes), it is hard to maintain high QoE during viewing. For videos with simple scenes (e.g., single-person scenes), G-PCC and Draco outperform others. And for videos with complex scenes, the neural-based codecs are more robust to different user viewing behaviors than conventional codecs.

The remainder of this paper is organized as follows. Section II surveys related work. In Section III, we describe the methodology of our measurements. The measurement results are described and analyzed in Section IV. Section V concludes this paper and points out our future work.

II. Related Work

In this section, we briefly survey related work in the fields of virtual reality video, point cloud compression and the design of volumetric video streaming systems.

Virtual reality video: There are different 3D data representations to deliver 3DoF (or 6DoF) virtual reality (VR) experience to users. A 360-degree video formatted in an equirectangular projection is currently the most popular VR video format. A 360-degree video [18], [19] allows users to control the viewing direction in three dimensions (e.g., yaw, pitch and roll) providing a 3DoF viewing experience. However, the immersive experience brought by the 3DoF interactive mode is limited, so how to allow users to freely explore the VR world with 6DoF interactive mode has become a research hotspot. To achieve 6DoF interactive mode, one way is to synthesize target views by providing multiple source views (e.g., depth and texture) according to the user's viewpoint [20], and the other way is to model the target scene with volumetric data (e.g., point cloud or polygon mesh) [21], [22]. The video generated by the second method is composed of a series of 3D models, which is called volumetric video. Point cloud-based volumetric video has the advantages of high flexibility and easy acquisition [9], [10], so it has become the primary research object of 6DoF VR video.

Point cloud compression: Conventional codec approaches for point cloud compression represent video data based on either 2D projection structure or 3D-tree structure. By adopting the representation of 2D projection structure, V-PCC (Video-based Point Cloud Compression) [8] projects the original 3D point cloud into the 2D space, and compresses them with 2D video compression methods. While for 3D-tree-based compression approaches, 3D data structures (e.g., kd-tree [16], octree [15] or trisoup [8]) are used in representative algorithms including Draco [12] and G-PCC (Geometry-based Point Cloud Compression) [8]. Draco is based on the kd-tree data structure to compress grid data and point clouds. Draco first encodes a rough representation of the full point cloud and then builds multiple refinement layers for efficient compression. While for G-PCC, it represents the point cloud using octree or trisoup data structures and performs arithmetical encoding on different attributes.

Recently, the neural-based techniques have been proved to be able to enhance the codec performance, which can be divided into two categories: the *optimization-based approach* [23], [24], [25] and the *deep learning-based approach* [13], [14], [17], [26]. Optimization-based approaches treat upsampling point clouds as an optimization problem while deep learning-based methods focus on feature extraction and expansion of point clouds. For deep learning-based methods, features are extracted from patches on the point cloud surface, and can be finally converted to the output 3D coordinates.

There existed a few works that analyzed point cloud compression methods [22], [27], [28], [29], but their objective was

(a) Pose (b) Office (c) Haggling (d) Pizza (e) Longdress (f) Soldier

Fig. 2. Rendered samples from 6 datasets.

TABLE I
VOLUMETRIC VIDEO DATASET

Index	Name	Description	Quality	Frames	Number of Points(K)	Required Bandwidth(Gbps)
1	Pose	One person without a background	Low density and Noisy	300	120	0.43
2	Office	One person with a background	Low density and Noisy	300	300	1.08
3	Haggling	Three people without a background	Low density and Noisy	300	250	0.90
4	Pizza	Five people with a background	Low density and Noisy	300	300	1.08
5	Longdress	One person without a background	High density and Noise-free	300	800	2.88
6	Soldier	One person without a background	High density and Noise-free	300	1000	3.59

to measure the overall performance of compression methods, without considering specific application scenarios for volumetric videos. Moreover, SR-based point cloud compression methods [11] have seldom been measured in the literature.

Volumetric video system design: A number of volumetric video streaming systems have been built, which have proven the possibility of volumetric video transmission [9], [10], [21], [30], [31], [32], [33]. Both ViVo [9] and Groot [10] constructed a transmission system with the parallel acceleration of conventional compression and optimization methods based on visual characteristics. However, they all suffered from a small compression ratio (e.g., less than 8) and required high bandwidth for video transmission. AITransfer [31] proposed a neural adaptive transmission scheme to extract and transfer features of each frame in a video. Unfortunately, AITransfer cannot handle color attributes of a point cloud. Yuzu [11] was the first to employ point cloud super-resolution methods to improve user experience based on the PU-GAN [17] and MPU [13] models. However, Yuzu only conducted experiments on small datasets(e.g., 100K points per frame) and has not compared its performance with conventional point cloud codecs.

The detailed comparison of different volumetric video codecs is still missing in the literature, which is the focus of our work in this paper.

III. Measurement Setup

In this section, we will briefly introduce our measurement settings, including datasets, metrics, video codecs, and measurement methodology.

A. Datasets

In our measurement, we use 6 volumetric video datasets as listed in Table I and the rendered samples are shown in Fig. 2. The point clouds in Datasets 1-4 [5] are generated by Microsoft Kinect sensors [34], which cover a variety of scenes and multiple types of human objects. The point clouds in Datasets 1-4 are usually sparse and noisy. On the contrary, the point clouds in Datasets 5-6 are captured by 42 synchronized RGB cameras [6], which are denser and noise-free. In the experiments, we select 300 frames per dataset.

B. Metrics

We use multiple types of metrics to evaluate the performance and efficiency of different volumetric video codecs, including:

- *Compression Ratio (CR)*, which is defined as the ratio of video size before compression to video size after compression.
- *Encoding Time (ET)/Decoding Time (DT)*, which is defined as the time required to perform encoding (or decoding) operation on the video content.
- *Chamfer distance (CD)*, which relates to pair-wise geometry distance of the closest points in two point clouds [27]. The chamfer distance between point cloud $\mathbf{S}_1$ and $\mathbf{S}_2$ is defined as:

$$\mathrm{CD}(\mathbf{S}_1, \mathbf{S}_2) = \frac{1}{2|\mathbf{S}_1|}\sum_{p\in\mathbf{S}_1}\min_{q\in\mathbf{S}_2}\|p-q\|_2^2 + \frac{1}{2|\mathbf{S}_2|}\sum_{q\in\mathbf{S}_2}\min_{p\in\mathbf{S}_1}\|p-q\|_2^2. \tag{1}$$

  A higher value of CD means that there exists higher distortion in the geometry.
- *CD-PSNR*, which is the peak signal of geometry over the chamfer distance distortion [27], [35]:

$$\text{CD-PSNR}(\mathbf{S}_1, \mathbf{S}_2) = 10 \times log_{10}\frac{D_m^2}{\mathrm{CD}(\mathbf{S}_1, \mathbf{S}_2)}, \tag{2}$$

  where

$$D_m = max\big\{(x_{max}-x_{min}), (y_{max}-y_{min}), (z_{max}-z_{min})\big\}, \tag{3}$$

  x_{max}, x_{min}, y_{max}, y_{min}, z_{max}, z_{min} are the maximum and minimum coordinates of the point cloud along with three axes respectively [36]. The value of CD-PSNR is positively correlated to decoding quality of geometric attribute.

- *Color-PSNR*, which represents the YUV difference between the closest points in the log scale [36], [37]:

$$\text{Color-PSNR} = \frac{6 \times \text{PSNR}_y + \text{PSNR}_u + \text{PNSR}_v}{8}, \quad (4)$$

where

$$\text{PSNR}_y = 10 \times log_{10}\left(\frac{255^2}{d_{MSE_y}(\mathbf{S}_1, \mathbf{S}_2)}\right), \quad (5)$$

and d_{MSE_y} is the Mean Square Error (MSE) of luminance values over all closest point pairs. PSNR_u and PSNR_v can be calculated by replacing the luminance values in Eq. (5) with chrominance and chroma values. When color attribute is better decoded, a higher value of Color-PSNR can be obtained.
- *Hybrid-PSNR (i.e., H-PSNR)*:

$$\text{H-PSNR} = \alpha \cdot \text{CD-PSNR} + (1 - \alpha) \cdot \text{Color-PSNR}, \quad (6)$$

which provides a global quality score by combining color-based and geometry-based metrics. It is the weighted sum of *Color-PSNR* and *CD-PSNR*, and the setting of the weights α is based on the measurement results of [38].
- *QoE*, which is a metric defined in [11] to evaluate the quality of user experience for volumetric video streaming. The QoE definition is as follows:

$$\text{QoE} = \sum_i Q_i - \sum_i \mu_f(\sigma) I_i^{frame} - \sum_i \mu_s(\sigma_i) I_i^{stall}, \quad (7)$$

where

$$Q_i = w_1(\sigma_i) \times d_i \times r_i - w_2(\sigma_i) \times m_i, \quad (8)$$

$$I_i^{frame} = \|Q_i - Q_{i-1}\|, \quad (9)$$

Q_i is the visual quality of each frame relating to the density d_i, upsampling ratio r_i and geometric distortion m_i of point clouds as well as the distance σ_i from the viewer, I_i^{frame} is the quality change between frame $i-1$ to frame i, and I_i^{stall} is the stall of frame i. The weights μ_f, μ_s, w_1 and w_2 are parameterized on σ_i and set according to the settings in [11].

C. Point Cloud-Based Volumetric Video Codecs

Existing point cloud-based volumetric video codecs can be generally divided into two categories, namely, *conventional codecs* and *neural-based codecs*. In this measurement, we mainly choose 5 state-of-the-art codecs [11], [22] for comparison, including 3 conventional codecs and 2 neural-based codecs. These 5 codecs are representative as they are also commonly used in point cloud-based volumetric video streaming systems [9], [11], [33], [39]. The description and settings of each codec are given as below:

- *Draco*, which was developed by Google [12] and has been widely adopted by existing systems. It controls the bit depth of different attributes by setting quantization parameters, and the encoder parameters are set according to [27].
- *G-PCC*, which was developed by MPEG [40] for the compression of static volumetric contents [41]. Lifting transformation [8] is used as the transforming tool and octree is set as the default data structure. The value of *pq_scale* and *qp* parameters corresponds to different compression effects and we set their value following the instructions from the MPEG's common test condition [42].
- *V-PCC*, which was developed by MPEG for the compression of dynamic volumetric contents [43]. The V-PCC encoder employs HEVC [8], [44] to encode the generated 2D videos using *all-intra* mode.[1] The values of *geoQP* and *texQP* indicate geometry and texture quantization parameters, respectively. And we also set the parameters according to the MPEG's common test condition [42].
- *PU-GCN+*, which is an extension of PU-GCN [14]. It adopts the SR technique to compress and decompress a point cloud and extends PU-GCN with more optimizations (e.g., extracting patches based on octree, nearest neighbor interpolation, merging SR input with the output [11], [32]). The model size of PU-GCN+ is around 1.75MB.
- *MPU+*, which is an extension of MPU [13]. Compared to PU-GCN+, MPU+ also uses spherical kernel function (SKF) [11], [45] to accelerate convolutions when extracting features. The model size of MPU+ is around 1.08 MB.

TABLE II
ENCODER PARAMETERS FOR FIVE CODECS AND EACH CODEC HAS FIVE SETS OF PARAMETERS CORRESPONDING TO FIVE BITRATE LEVELS

Codec	Parameters	R1	R2	R3	R4	R5
Draco	*qt*	6	7	8	9	10
	qp	6	7	8	9	10
G-PCC	*pq_scale*	0.25	0.5	0.75	0.875	0.9375
	qp	46	40	34	28	22
V-PCC	*geoQP*	32	28	24	20	16
	texQP	42	37	32	27	22
PU-GCN+	*up_ratio*	21	17	9	5	3
MPU+	*up_ratio*	21	17	9	5	3

In our comparison, we select 5 different settings (i.e., R1, R2, ..., R5) for each codec when compressing point clouds into various quality. From *R1* to *R5*, the bitrate of each codec increases sequentially. For conventional codecs, their parameter selection is based on the works mentioned above. For PU-GCN+ and MPU+, we select five upsampling ratios (*up_ratio*) similar to [11]. The parameter settings are shown in Table II.

D. Methodology

We measure the performance of all codecs on a computer equipped with an AMD Ryzen 5 1600X 6-core 3.60GHz CPU and an NVIDIA GeForce GTX 1080Ti GPU. The volumetric video player is designed with the Point Cloud Library(PCL) [46] and written in C++. PU-GCN+ and MPU+ are implemented using TensorFlow. We use a model pre-trained on the PU1K dataset [17] to upsample all point clouds for different datasets. The pre-trained model can be

[1]The encoder of V-PCC employs the all-intra mode to archieve fair comparison because the other 4 codecs cannot compress the inter-frame redundant information of point cloud-based volumetric videos.

generalized to different point clouds with similar upsampling results.

We first evaluate the performance of different codecs for different application scenarios. For each codec, we apply five settings to process all 6 datasets in Table I. For each metric, we calculate the average value for all frames. Before performing the experiments, all point cloud files are converted into binary format.

Next, we proceed to investigate the influence of different attributes of a point cloud on the codec performance including *color, geometric* and *scene complexity* as well as *noise level*. For the experiments on color attribute, we make some deformation on the longdress dataset by using interpolation to change the texture complexity without changing its geometric attribute. To characterize the color attribute, we capture several screenshots of those point clouds and use texture complexity calculated by the entropy of the grey-level co-occurrence matrix [47]. For the experiment on the geometric attribute, we perform surface reconstruction of the longdress dataset and simplify the mesh to various degrees in order to obtain point clouds with different geometric complexity. In our measurement, the number of face elements ranges from 130 to 2,050,000. When measuring targeted scene complexity, we vary the scene complexity of a point cloud by stitching different numbers of objects in the longdress dataset. The more objects a scene contains, the higher the scene complexity is. In order to generate point clouds with different noise levels, we add Gaussian noise to their geometry elements with standard deviations from 0.5 to 20. A larger noise deviation leads to more severe quality degradation. When exploring the influence of one attribute, we strictly control the size and the other three attributes of the generated point cloud to be consistent. In the measurement, we choose two sets of parameters that can best show changes, corresponding to the levels of *R1* and *R5*.

Finally, we also study the performance of these five codecs in on-demand volumetric video streaming scenarios with real user viewing traces. We first collect viewing traces of eight users by letting them watch 6 videos (as shown in Table I) freely and record viewport positions and orientations for each frame. Then we calculate QoE when using different codecs under two bandwidth settings [11], namely, 25Mbps and 75Mbps. In the measurement, we configure encoder parameters to be at the *R4* level, as five codecs have similar decoding quality with parameters at this level.

For conventional codecs, we apply the optimization method *Distance Visibility (DV)* in ViVo [9] to establish the mapping between the distance and the *point density level (PDL)*. When a user's viewpoint is far away, objects in the field of view will become small and adjacent 3D points in the point cloud may be projected to the same 2D pixel. As a result, point cloud density can be reduced while the visual quality of a user can still be maintained. The maximum value of PDL is set as 1.0 in default, which means that the upper limit of point cloud density is the raw data density. For neural-based codecs, the upsampling ratio is equal to the product of the compression ratio and PDL when the value of PDL is less than 1.0. When the value of PDL is equal to 1.0, the most suitable model will be selected from the pre-trained models to minimize the resolution change.

IV. Comparative Study and Analysis

In this section, we describe our measurement results and compare the performance of different codecs.

A. Practicality and Performance of Volumetric Video Codecs

For all five codecs, we compare their performance with five different settings using 6 real datasets. Fig. 3(a) shows the compression ratio of different codecs. According to the results, V-PCC and G-PCC perform much better than others. For G-PCC at the *R1* level, a soldier dataset with a size of 4.5GB can be compressed into 3.5MB. However, the compression ratio of Draco, MPU+ and PU-GCN+ at high-bitrate levels (e.g., *R4* and *R5*) are less satisfactory. It is difficult to meet the requirement of transmitting a large volumetric video with compression ratio of less than 6.0.

Fig. 3(b) shows the value of Hybrid-PSNR of different codecs and Fig. 4 shows the decoding quality of each codec on different datasets in more details. It can be found that the quality of volumetric video after decoding varies with codec type and attributes of original videos before encoding. For high-quality video datasets (e.g., longdress, soldier), all codecs can perform well in decoding to obtain geometric attributes at high-bitrate level. However, for low-quality video datasets (e.g., pose, office, haggling, pizza), V-PCC and G-PCC exhibit significant performance degradation, while Draco and neural-based codecs can properly decode geometric attributes. Such results are caused by the projection mechanism of V-PCC and the sensitivity of the predictor of G-PCC to the density of a point cloud. For color attributes, it is obvious that for all 5 codecs, the decoding quality of the soldier dataset is the best, which is due to simple color attribute of this dataset.

When the bitrate is low (e.g., at the *R1* level), the point cloud quality has no significant effect on the decoding effect of color attributes. However, at the *R5* level, decoding color attributes on high-quality datasets is significantly better than that on low-quality datasets. For example, Draco can decode color attributes losslessly for longdress and soldier datasets at a high bitrate, but the value of Color-PSNR cannot exceed 40 for low-quality datasets. It is also caused by noises that interfere with the work of the color prediction module in the codec, resulting in performance degradation. In Section IV-B, we will delve deeper into the impact of point cloud attributes on the codec.

According to the above results, the compression quality and efficiency of Draco and G-PCC will change significantly with the change of quantization parameters while V-PCC and neural-based methods are relatively stable. Therefore, when selecting parameters for G-PCC and Draco, a dedicated tradeoff should be considered.

Fig. 3(c) and Fig. 3(d) plot the encoding and decoding time of different codecs. The encoding time of V-PCC is the longest among all the codecs, which takes around 2 minutes to compress a single frame. It is because that V-PCC encodes point clouds based on decomposing them into multiple patches with

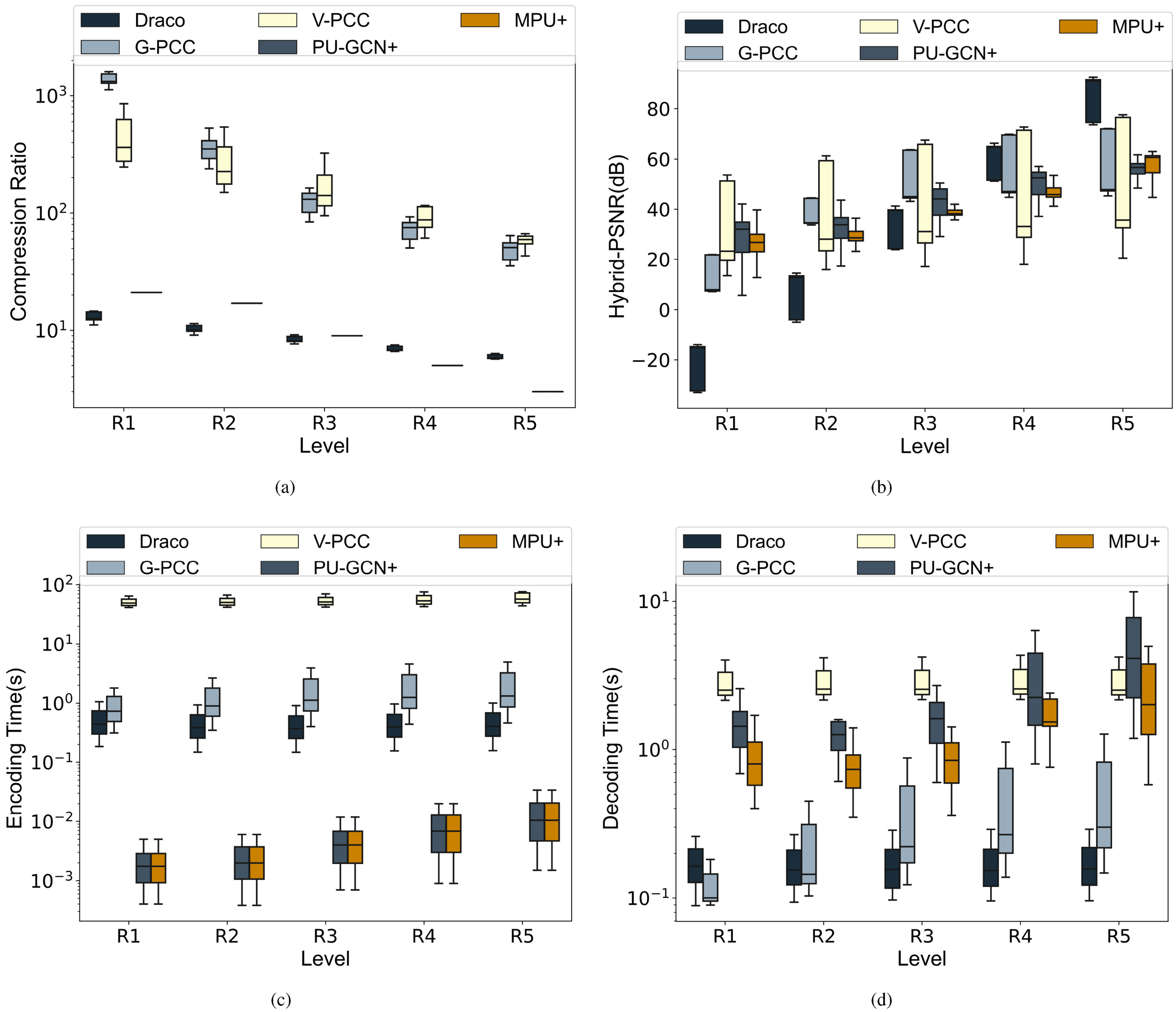
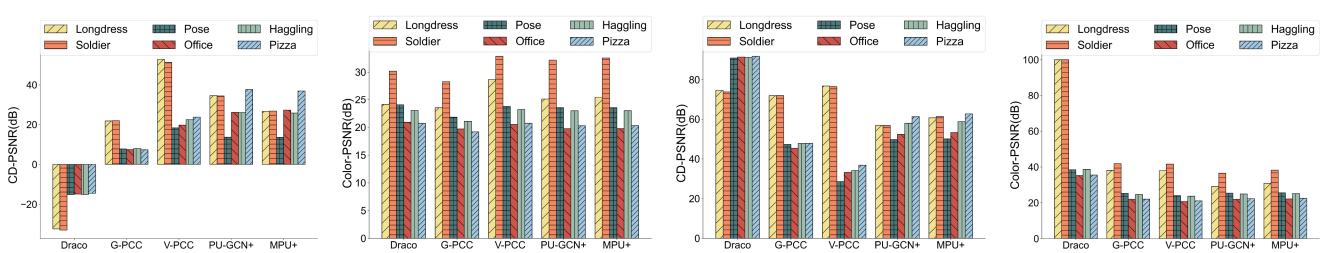


Fig. 3. Performance of different codecs on 6 datasets with 4 metrics: (a) Compression Ratio, (b) Hybrid-PSNR, (c) Encoding Time, (d) Decoding Time.



(a) CD-PSNR result at *R1* level (b) Color-PSNR result at *R1* level (c) CD-PSNR result at *R5* level (d) Color-PSNR result at *R5* level

Fig. 4. Measurement results of CD-PSNR and Color-PSNR at *R1* and *R5* levels.

surface normal information, which is very time-consuming. Instead, MPU+ and PU-GCN+ are highly efficient in encoding with random downsampling as the encoding method, which are able to encode thousands or even tens of thousands of frames in one minute. For decoding time, Draco performs the best and can reach 30FPS for small-size point clouds (such as 300k points/frame). However, with the increase of point cloud size, all five codecs perform poorly in decoding.

Particularly, PU-GCN+ and V-PCC are the worst in terms of decoding time. Moreover, in 5 groups of experiments, the time consumption of V-PCC seldom changes under different parameter settings. The results indicate that the bottleneck of V-PCC in terms of time consumption is the conversion between point cloud and patches. In addition, MPU+ is significantly better than PU-GCN+ in terms of decoding efficiency with the acceleration gains brought by SKF, which quantizes the local

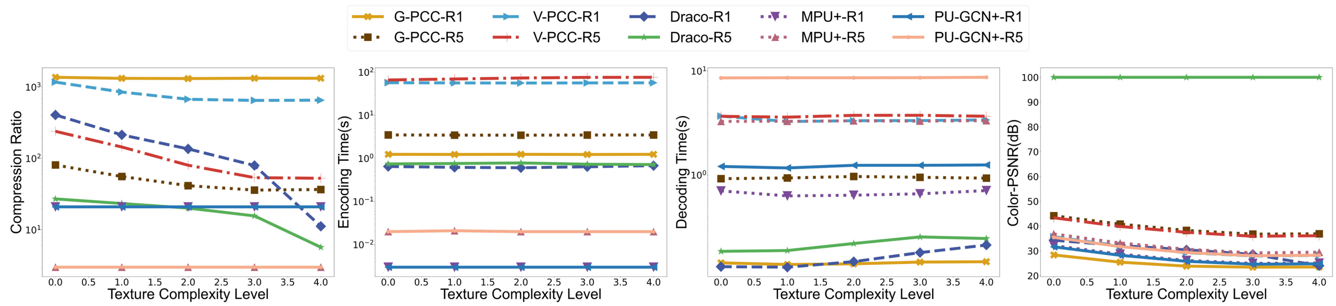


Fig. 5. Impact of color attribute on different codecs.

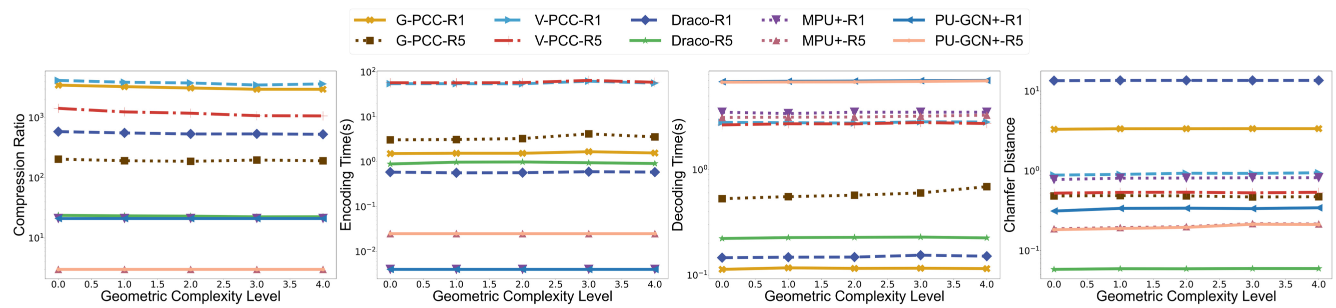


Fig. 6. Impact of geometric attribute on different codecs.

3D space of point clouds and specifies a learnable parameter to convolve the points falling in each sub-space [45]. It can reduce the computational overhead without reducing precision.

In summary, G-PCC is most suitable for the transmission of high-quality volumetric videos in on-demand streaming applications due to its high compression ratio and excellent decoding quality. Instead, Draco is suitable for delivering volumetric videos with a small data volume or with sufficient bandwidth, as it has a limited compression ratio and maximum decoding efficiency. For live streaming, all the codecs cannot meet the latency requirement. For Draco, its encoding speed is only 10FPS for a small dataset of 300K points/frame. Because neural-based codecs use downsampling as the encoding method, they can process hundreds of frames per second. However, the long single-frame decoding time (e.g., a few seconds) limits their application, which needs further optimization.

B. Impact of Different Attributes on Codec Performance

Fig. 5 shows the influence of color attribute (e.g., texture complexity) on the codec performance. According to the results, Draco at the *R1* level shows the maximum performance attenuation with the increase of texture complexity. When the value of texture complexity of a point cloud increases from 0.48 to 0.56, its compression ratio drops to 1/30 of the original size and the decoding time doubles, which indicates its poor performance in compressing color attribute. Apart from Draco, the compression ratio of V-PCC at the *R5* level also decreases with the increase of texture complexity. On the contrary, PU-GCN+ and MPU+ perform much more stably in terms of compression ratio and encoding/decoding time with the nearest neighbor interpolation to upsample the color attribute. For Color-PSNR, Draco at the *R5* level does not exhibit fluctuation and performs well on encoding/decoding complex textures. The performance of other codecs drop by about 5 dB with the change of color attribute. It is because that the quantization parameter of Draco is large enough to encode and decode complex textures without information loss. However, for other codecs, parameter settings at the *R5* level do not correspond to their lossless mode,[2] so it will cause a certain loss of information after quantization.

Fig. 6 and Fig. 7 show the impact of geometric and scene complexity on the codec performance, respectively. When exploring the geometric complexity, only a single geometry is included in the point cloud. We change geometric complexity of the point cloud by varying the number of face elements. As shown in Fig. 6, the performance of all five codecs does not show a significant change when the geometric complexity increases. However, when the scene complexity increases, there exhibits significant fluctuation in terms of codec performance. According to Fig. 7, V-PCC is most sensitive to scene complexity. When the number of objects in the point cloud is increased from 1 to 5, the chamfer distance between the decoded point cloud and the original one quickly increases to 10^5 times, which means that there exists significant geometric distortion. This is due to the fact that the V-PCC codec is projection-based, and a point cloud with complex scene (e.g., multiple objects) requires more 2D layers to be generated, thus increasing the difficulty of encoding and decoding.

It is also worth mentioning that the slight performance fluctuation of MPU+ and PU-GCN+ is caused by the improved

[2] G-PCC and V-PCC can achieve lossless compression via parameter adjustment, but the compression performance will greatly degrade (e.g., the compression ratio will drop to around 10.0). MPU+ and PU-GCN+ cannot achieve lossless compression.

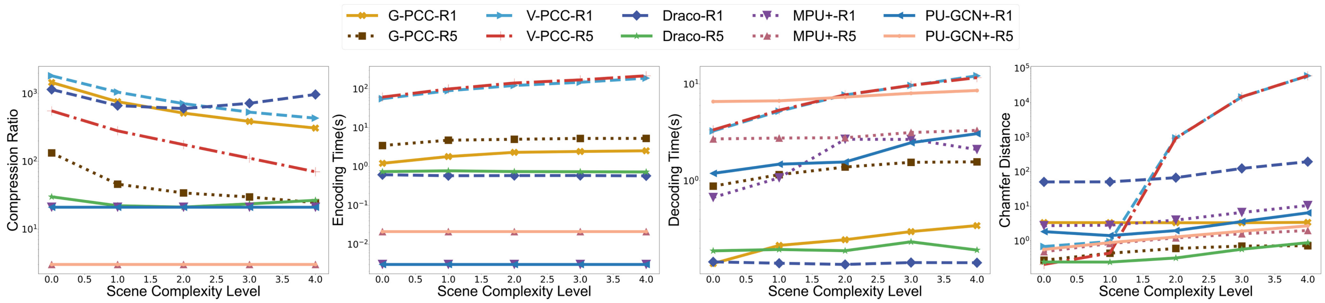


Fig. 7. Impact of scene complexity on different codecs.

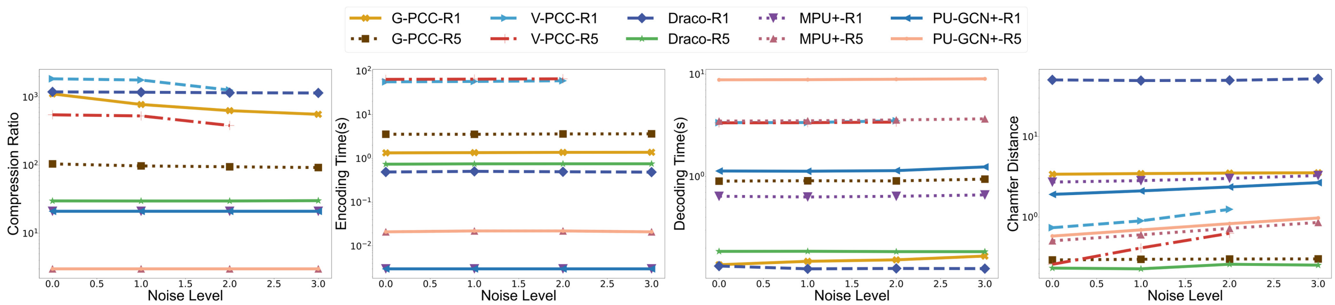


Fig. 8. Impact of noise level on different codecs.

patch generation method. When generating patches through octree, if the number of points in a patch is too small, patches will be discarded. It results in that the shape and density of a point cloud will affect the number of patches. In this experiment, we increase the number of objects without changing the size of the point cloud file, which results in significant changes in the density of the point cloud and patches divided by the octree. Therefore, the decoding time of MPU+ and PU-GCN+ will fluctuate.

Fig. 8 plots the effects of noise intensity on codec performance. It is observed that V-PCC fails to encode the point cloud when noise intensity reaches the highest level (standard deviation of 20). This is due to the high noise intensity, which affects the normal estimation of a point cloud and thus interferes with patch generation. When noise intensity is low, the compression ratio and decoding quality of V-PCC will also degrade with the addition of noise. On the contrary, the performance of Draco is the most stable when handling data with noise. There is little change on compression performance or point cloud quality after decoding.

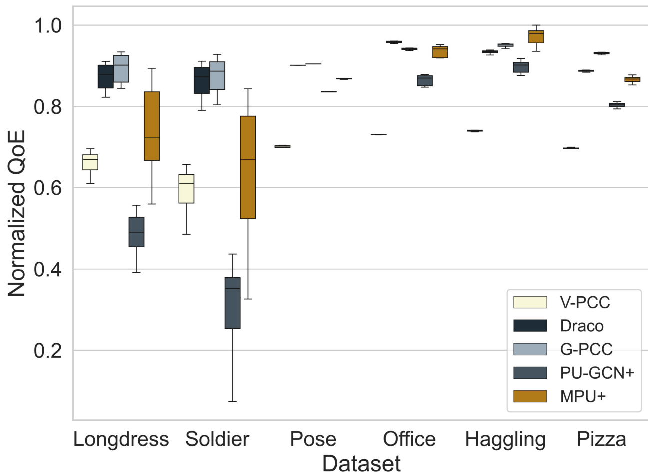
In summary, Draco is the most instable one when color attribute changes. The performance of V-PCC becomes the worst with the increase of scene complexity and noise level. Besides, the compression ratio, encoding time and decoding time of PU-GCN+ and MPU+ do not fluctuate significantly when these four attributes change. This is because their downsampling and upsampling methods are independent of the point cloud content, which enables them to maintain stable performance when the video content changes. The above measurement results provide useful guidelines on how to select a better codec (or determine encoder parameters) according to the attributes of a point cloud.

C. Impact of User Viewing Behaviors on QoE Changes

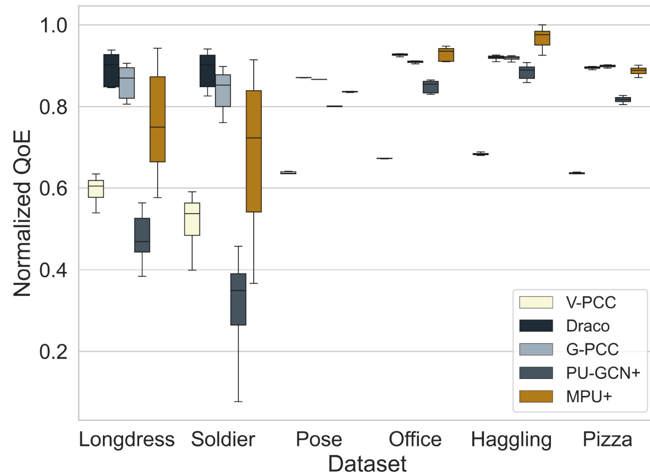
Fig. 9 shows the QoE result for each dataset and codec. For high-quality datasets with simple scene (e.g., longdress, soldier), G-PCC and Draco outperform others with excellent decoding quality and low decoding delay. When the bandwidth increases, the QoE of Draco also increases. The reason is that, when the bandwidth is sufficient, Draco's low compression ratio is no longer a performance bottleneck. For datasets with complex scene (e.g., office, haggling, pizza), MPU+ and PU-GCN+ show significant improvement. It attributes to the QoE gain brought by point cloud upsampling algorithm. On the contrary, V-PCC with long decoding delay is inferior to the others in all six experiments.

Fig. 10 shows the result of visual quality, which is an important component of QoE. It is obvious that both MPU+ and PU-GCN+ outperform conventional codecs on 6 datasets, especially on datasets with complex scenes. Through analysis of collected user trajectories, it is found that, when viewing videos with complex scenes, a user's range of motion at z-axis is 160% larger than that when viewing videos with simple scenes. Videos with complex scenes can provide users with richer information and more viewport options, however, they also pose a higher requirement for resolution adaptation. As neural-based methods can control the flexibility of video resolution by adjusting the upsampling ratio, they are able to enhance a user's QoE significantly, especially for volumetric videos with complex scenes. As shown in Fig. 11, with the decrease of viewing distance of a user, the upsampling operation on a point cloud can effectively increase the density of a point cloud, thereby improving the visual quality.

Based on our measurements, we provide the pros and cons of different codecs as follows:



(a) 25Mbps



(b) 75Mbps

Fig. 9. Normalized QoE over different bandwidth: (a): 25Mbps; (b): 75Mbps.

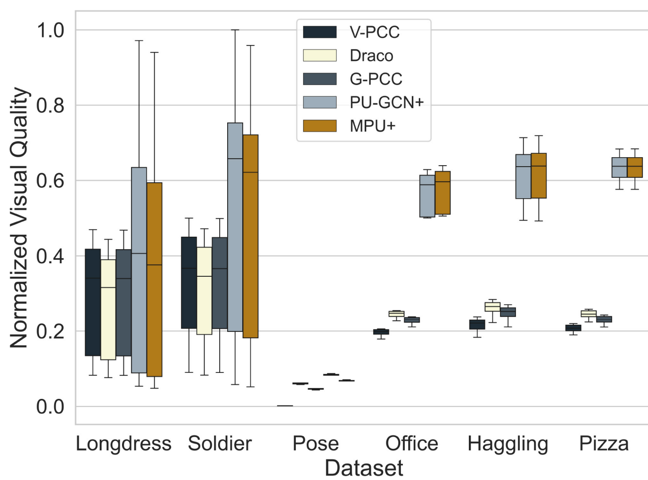


Fig. 10. Normalized visual quality on different codecs and datasets.

(a)

(b)

Fig. 11. Screenshots of point clouds of haggling dataset from different viewpoints decoded by different kinds of codecs: (a) point clouds decoded by conventional codecs; (b) point clouds decoded by neural-based codecs.

- *Draco:* The compression ratio of Draco is limited while its maximum decoding efficiency is suitable for delivering videos with small size or with sufficient bandwidth. It can reach 30FPS when the size of point cloud does not exceed 300K points/frame. Besides, its compression ratio will increase significantly as the color complexity of a point cloud decreases. In addition, Draco can encode and decode geometric attributes of both low-quality and high-quality point clouds well and be insensitive to changes in geometric and scene complexity as well as noise level.
- *G-PCC:* G-PCC takes both high compression ratio and efficiency into account. However, when dealing with low-quality point clouds, the decoding quality will be degraded. Those features make it more suitable for transmitting high-quality point clouds with large data volume in on-demand scenarios. When the quantization factor is 0.25, its compression ratio can even reach 1000, but the quality is poor. As the quantization factor increases, its compression performance and quality will also change significantly. Therefore, it is necessary to find a trade-off between these two factors during real usage.
- *V-PCC:* V-PCC compresses point clouds based on 2D video compression technology, which has pros and cons. The advantage is that it can achieve a high compression ratio with the help of 2D video codecs and reuse the architecture of 2D video transmission systems. However, V-PCC needs to convert point clouds into patches for compression, which leads to long encoding and decoding delay. Besides, the increase in noise level and scene complexity also greatly degrades its performance.
- *PU-GCN+:* Due to its short encoding latency and stable performance, PU-GCN+ has potential to be used in live streaming applications. The performance can be well maintained for point clouds with different attributes. For videos with complex scene, PU-GCN+ can adjust the point cloud density according to a user's viewing distance and improve user viewing quality. However, its decoding operation is too time-consuming and the nearest neighbor interpolation method used by PU-GCN+ cannot well deal with complex texture information.

- *MPU+:* Similar to PU-GCN+, MPU+ also has low encoding latency, stable performance and excellent ability to improve the visual quality of volumetric videos. Meanwhile, with the help of SKF, MPU+ can effectively reduce the decoding time, and show much better performance than PU-GCN+ when streaming point clouds with large data volume. However, its decoding delay is still too high.

D. Insights and Discussion

We briefly summarize all the important insights obtained from our comparative study and discuss their implications as below:

- Existing conventional codecs can hardly meet the requirement of live volumetric video streaming due to their long encoding time, while neural-based codecs show much better performance in encoding.
- Selecting an appropriate bitrate according to the content of a point cloud and the characteristics of the codec can improve the coding efficiency. Different codecs exhibit diverse sensitivity to different point cloud features. It is efficient to select an appropriate bitrate for a video by tile or chunk based on codec-specific characteristics and video content analysis.
- A point cloud with complex scenes has a higher requirement for the selection of codec in the streaming system. It results in more sophisticated user viewing behaviors and more severe fluctuation in size between frames. Since neural-based codecs can maintain high viewing quality of the video by upsampling point clouds, they exhibit superior performance in the transmission of point clouds with complex scenes.
- When designing a neural-based point cloud video streaming system, the user experience can be effectively improved by preset models at the client side. The neural-based codecs exhibit satisfactory generalization capability, especially for the geometric attribute of point clouds, which means the overhead of video transmission and model training can be reduced by preset models.

Conventional codecs have been studied extensively in the past few years, but there are still some obstacles that prevent their wide application in point cloud-based volumetric video streaming.

Instable color compression ratio: Our measurements show that the compression ratio of Draco decreases rapidly as the color complexity of a point cloud increases. For example, when Draco compresses point clouds with low color complexity at the R1 bitrate level, its compression ratio can be as high as 712. But when it deals point clouds with high color complexity, the compression ratio drops significantly to 11, which indicates its inefficiency when dealing with color attribute. It is because the quantization operation in Draco is used only for floating-point attributes. If integer attributes are provided, Draco will treat them as pre-quantized. Therefore, the color attribute of a point cloud will not be quantized in Draco, which leads to its low compression ratio.

Low performance due to 3D to 2D projection: V-PCC needs to extract surface normals before projecting a point cloud from the 3D space to the 2D plane. This is why V-PCC will experience a decrease in efficiency or even incompressibility when dealing with noisy data. Meanwhile, When the scene complexity of a point cloud increases, 2D layers projected by the V-PCC encoder will also increase, resulting in a decrease in coding performance. To solve these problems, preprocessing (e.g., denoising) before encoding point cloud is necessary. Moreover, some work [48] has improved the efficiency of V-PCC by optimizing the process of 2D projection.

Lack of inter-frame compression capability: V-PCC can use 2D codecs to perform inter-frame compression, but tree-based codecs such as Draco and G-PCC cannot exploit temporal redundancy in a point cloud. There have been quite a few studies have aimed to achieve inter-frame compression of tree-based codecs. Mekuria et al. [1] adapted point cloud registration technology to extract rigid transformation between two subsets of points. Recently, optimizing the inter-frame predictor with deep learning [49], [50] has become a feasible direction. Besides, using technologies such as point cloud scene flow [51] to extract motion vectors between frames is also a promising research direction.

Long processing and inference delay: The time consumed by a neural-based codec can be divided into two parts, namely, the *processing time* and the *inference time*. Processing time includes the time of extracting patches and uniform downsampling. For example, if we replace the most time-consuming farthest point sampling with voxel downsampling, the processing time can be significantly reduced. Further, if we improve the way to generate patches with octree, we can simply remove the step of uniform downsampling and reduce the total time to 1/300 of the original time. In the inference stage, feature extraction consumes a large portion of time. The key function of feature extraction is to establish topological relationships between scattered points. The current k-nearest neighbors (kNN) method is inefficient to provide the above function. It is better to find other efficient methods to replace the kNN method. Yuzu [11] applies SKF which partitions a 3D space into multiple volumetric bins to extract neighborhood information more efficiently. However, the method still suffers from poor performance when dealing with point clouds with large data volume. On one hand, we can speed up inference through optimization methods such as parallelization, and on the other hand, we can introduce a simpler network structure to perform low-complexity inference on simple geometric or color attributes.

Inference problem for color attribute: Existing upsampling methods cannot well handle the color attribute of point clouds. From our measurement results, we can find the interpolation method for calculating color attributes also has limited effects especially for point clouds with complex textures. Therefore, a color model is needed to colorize the high-resolution point cloud with a low-resolution colored one. We can feed each patch to a coloring module after upsampling, whose depth can be changed based on the color complexity of that patch. Moreover, the geometric model can form a pipeline with the coloring model to further speed up the processing. In addition to the *feature extraction-feature expansion* upsampling structure adopted by most methods, the *dense generator-spatial refiner* structure shown in Dis-PU [52] also deserves attention.

After rough upsampling, a patch can be fed into a geometry refinement network and a color refinement network at the same time to speed up the processing.

Lack of capability to analyze redundant information: Existing neural-based codecs only perform random downsampling when compressing point clouds, which may result in many important point features being discarded [53]. To solve this problem, it is necessary to analyze and utilize the characteristics of a point cloud itself when performing downsampling. Recent researches show learning-based point cloud downsampling methods [54], [55], [56] can improve application performance compared to traditional methods(e.g., random sampling and farthest point sampling). However, these methods are only applicable for small-scale and colorless point clouds, so it is of great research value to downsample large-scale and colored point clouds with the help of learning-based methods. Besides, how to make better use of the inter-frame redundancy of volumetric video for compression also needs attention. Yuzu [11] reduces computational overhead on the decoding side by caching and reusing similar patches between frames. This method can be further improved with point cloud registration [32], [57] or video-based point cloud upsampling framework [58] for better compression performance.

V. Conclusion

In this paper, we conduct an in-depth comparative measurement study on five point cloud-based volumetric video codecs, including 3 conventional codecs and 2 neural-based codecs. For each codec, we analyze its performance on six real datasets with different characteristics. Our results reveal the effectiveness of different codecs under various application scenarios and the impact of point cloud features, such as texture, geometric complexity and so on. We also study the effect of user viewing behaviors on the codec performance with real user viewing traces. Our results are useful in helping the design of volumetric video streaming systems. In the next step, we plan to conduct more comprehensive measurements in terms of neural-based codecs and consider how to improve their efficiency with the state-of-the-art neural enhancement techniques.

References

[1] R. Mekuria, C. Blom, and P. C. Garcia, "Design, implementation, and evaluation of a point cloud codec for tele-immersive video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 4, pp. 828–842, Apr. 2017.

[2] S. Schwarz and M. Pesonen, *Real-Time Decoding and AR Playback of the Emerging MPEG Video-Based Point Cloud Compression Standard*, IBC, London, U.K., 2019.

[3] "Volumetric video market by volumetric capture & content creation, application, and geography—Global forecast to 2026." MarketsandMarkets. 2020. [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/volumetric-video-market-259585041.html

[4] S. F. Langa, M. M. Climent, G. Cernigliaro, and D. R. Rivera, "Toward hyper-realistic and interactive social VR experiences in live TV scenarios," *IEEE Trans. Broadcast.*, vol. 68, no. 1, pp. 13–32, Mar. 2022.

[5] H. Joo et al., "Panoptic studio: A massively multiview system for social interaction capture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 190–204, Jan. 2019.

[6] T. M. Eon, B. Harrison, and P. A. Chou, "8i voxelized full bodies—A voxelized point cloud dataset," ISO, Geneva, Switzerland, ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006, Jan. 2017.

[7] A. Collet et al., "High-quality streamable free-viewpoint video," *ACM Trans. Graph.*, vol. 34, pp. 1–13, Jul. 2015.

[8] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, "An overview of ongoing point cloud compression standardization activities: Video-based (V-PCC) and geometry-based (G-PCC)," *APSIPA Trans. Signal Inf. Process.*, vol. 9, p. e13, Apr. 2020.

[9] B. Han, Y. Liu, and F. Qian, "ViVo: Visibility-aware mobile volumetric video streaming," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.*, London, U.K., Sep. 2020, pp. 1–13.

[10] K. Lee, J. Yi, Y. Lee, S. Choi, and Y. M. Kim, "GROOT: A real-time streaming system of high-fidelity volumetric videos," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.*, London, U.K., Sep. 2020, pp. 1–14.

[11] A. Zhang, C. Wang, B. Han, and F. Qian, "YuZu: Neural-enhanced volumetric video streaming," in *Proc. 19th USENIX Symp. Netw. Syst. Des. Implement. (NSDI)*, Renton, WA, USA, Apr. 2022, pp. 137–154.

[12] "Draco." Google. Accessed: Jan. 10, 2022. [Online]. Available: https://google.github.io/draco

[13] Y. Wang, S. Wu, H. Huang, D. Cohen-Or, and O. Sorkine-Hornung, "Patch-based progressive 3D point set upsampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 5951–5960.

[14] G. Qian, A. Abualshour, G. Li, A. K. Thabet, and B. Ghanem, "PU-GCN: Point cloud upsampling using graph convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11683–11692.

[15] D. Meagher, "Geometric modeling using octree encoding," *Comput. Graph. Image Process.*, vol. 19, pp. 129–147, Jun. 1982.

[16] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, pp. 509–517, Sep. 1975.

[17] R. Li, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "PU-GAN: A point cloud upsampling adversarial network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct./Nov. 2019, pp. 7202–7211.

[18] Y. Zhou, L. Tian, C. Zhu, X. Jin, and Y. Sun, "Video coding optimization for virtual reality 360-degree source," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 1, pp. 118–129, Jan. 2020.

[19] M. Dasari, A. Bhattacharya, S. Vargas, P. Sahu, A. Balasubramanian, and S. R. Das, "Streaming 360-degree videos using super-resolution," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2020, pp. 1977–1986.

[20] J. M. Boyce et al., "MPEG immersive video coding standard," *Proc. IEEE*, vol. 109, no. 9, pp. 1521–1536, Sep. 2021.

[21] Y. Liu, B. Han, F. Qian, A. Narayanan, and Z.-L. Zhang, "Vues: Practical mobile volumetric video streaming through multiview transcoding," in *Proc. 28th Annu. Int. Conf. Mobile Comput. Netw.*, New York, NY, USA, 2022, pp. 514–527.

[22] E. Zerman, C. Ozcinar, P. Gao, and A. Smolic, "Textured mesh vs coloured point cloud: A subjective study for volumetric video compression," in *Proc. 12th Int. Conf. Qual. Multimedia Exp. (QoMEX)*, 2020, pp. 1–6.

[23] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Computing and rendering point set surfaces," *IEEE Trans. Vis. Comput. Graph.*, vol. 9, no. 1, pp. 3–15, Jan.–Mar. 2003.

[24] C. Dinesh, G. Cheung, and I. V. Bajić, "3D point cloud super-resolution via graph total variation on surface normals," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2019, pp. 4390–4394.

[25] C. Dinesh, G. Cheung, and I. V. Bajic, "Super-resolution of 3D color point clouds via fast graph total variation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 1983–1987.

[26] S. Ye, D. Chen, S. Han, Z. Wan, and J. Liao, "Meta-PU: An arbitrary-scale upsampling network for point cloud," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 9, pp. 3206–3218, Sep. 2022.

[27] C.-H. Wu, C.-F. Hsu, T.-K. Hung, C. Griwodz, W. T. Ooi, and C.-H. Hsu, "Quantitative comparison of point cloud compression algorithms with PCC arena," *IEEE Trans. Multimedia*, early access, Feb. 28, 2022, doi: 10.1109/TMM.2022.3154927.

[28] A. Javaheri, C. Brites, F. Pereira, and J. Ascenso, "Point cloud rendering after coding: Impacts on subjective and objective quality," *IEEE Trans. Multimedia*, vol. 23, pp. 4049–4064, Nov. 2020.

[29] 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 Trans. Broadcast.*, vol. 66, no. 3, pp. 701–717, Sep. 2020.

[30] F. Qian, B. Han, J. Pair, and V. Gopalakrishnan, "Toward practical volumetric video streaming on commodity smartphones," in *Proc. 20th Int. Workshop Mobile Comput. Syst. Appl. (HotMobile)*, Santa Cruz, CA, USA, Feb. 2019, pp. 135–140.

[31] Y. Huang, Y. Zhu, X. Qiao, Z. Tan, and B. Bai, "AITransfer: Progressive AI-powered transmission for real-time point cloud video streaming," in *Proc. ACM Multimedia Conf. (MM)*, Oct. 2021, pp. 3989–3997.

[32] A. Zhang, C. Wang, B. Han, and F. Qian, "Efficient volumetric video streaming through super resolution," in *Proc. 22nd Int. Workshop Mobile Comput. Syst. Appl. (HotMobile)*, Feb. 2021, pp. 106–111.

[33] J. Li, C. Zhang, Z. Liu, R. Hong, and H. Hu, "Optimal volumetric video streaming with hybrid saliency based tiling," *IEEE Trans. Multimedia*, early access, Feb. 23, 2022, doi: 10.1109/TMM.2022.3153208.

[34] Z. Zhang, "Microsoft Kinect sensor and its effect," *IEEE Multimedia*, vol. 19, no. 2, pp. 4–10, Feb. 2012.

[35] Y. Xu, Q. Yang, L. Yang, and J.-N. Hwang, "EPES: Point cloud quality modeling using elastic potential energy similarity," *IEEE Trans. Broadcast.*, vol. 68, no. 1, pp. 33–42, Mar. 2022.

[36] E. Dumic and L. A. da Silva Cruz, "Point cloud coding solutions, subjective assessment and objective measures: A case study," *Symmetry*, vol. 12, no. 12, p. 1955, 2020.

[37] C.-H. Wu, X. Li, R. Rajesh, W. T. Ooi, and C.-H. Hsu, "Dynamic 3D point cloud streaming: Distortion and concealment," in *Proc. ACM NOSSDAV*, 2021, pp. 98–105.

[38] I. Viola, S. Subramanyam, and P. Cesar, "A color-based objective quality metric for point cloud contents," in *Proc. QoMEX*, 2020, pp. 1–6.

[39] Y. Gao, P. Zhou, Z. Liu, B. Han, and P. Hui, "FRAS: Federated reinforcement learning empowered adaptive point cloud video streaming," 2022, *arXiv:2207.07394*.

[40] S. Schwarz et al., "Emerging MPEG standards for point cloud compression," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 133–148, Mar. 2019.

[41] "Geometry-based point cloud compression." MPEG. 2021. [Online]. Available: https://github.com/MPEGGroup/mpeg-pcc-tmc13

[42] S. Schwarz, P. A. Chou, and I. Sinharoy, "Common test conditions for point cloud compression," ISO, Geneva, Switzerland, ISO/IEC JTC1/SC29 Joint WG11 (MPEG)input document N17345, 2018.

[43] "Video-based point cloud compression." MPEG. 2022. [Online]. Available: https://github.com/MPEGGroup/mpeg-pcc-tmc2

[44] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[45] H. Lei, N. Akhtar, and A. Mian, "Spherical kernel for efficient graph convolution on 3D point clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3664–3680, Oct. 2021.

[46] "Point cloud library (PCL)." Accessed: Jan. 22, 2022. [Online]. Available: http://pointclouds.org/

[47] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[48] E. Lopes, J. Ascenso, C. Brites, and F. Pereira, "Adaptive plane projection for video-based point cloud coding," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, 2019, pp. 49–54.

[49] P. Gomes, "Graph-based network for dynamic point cloud prediction," in *Proc. 12th ACM Multimedia Syst. Conf.*, 2021, pp. 393–397.

[50] A. Akhtar, Z. Li, and G. Van der Auwera, "Inter-frame compression for dynamic point cloud geometry coding," 2022, *arXiv:2207.12554*.

[51] X. Liu, C. R. Qi, and L. J. Guibas, "FlowNet3D: Learning scene flow in 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 529–537.

[52] R. Li, X. Li, P.-A. Heng, and C.-W. Fu, "Point cloud upsampling via disentangled refinement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 344–353.

[53] Q. Hu et al., "Learning semantic segmentation of large-scale point clouds with random sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8338–8354, Nov. 2022.

[54] I. Lang, A. Manor, and S. Avidan, "SampleNet: Differentiable point cloud sampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7575–7585.

[55] O. Dovrat, I. Lang, and S. Avidan, "Learning to sample," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2755–2764.

[56] J. Yang et al., "Modeling point clouds with self-attention and Gumbel subset sampling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3318–3327.

[57] G. Arvanitis, E. I. Zacharaki, L. Vása, and K. Moustakas, "Broad-to-narrow registration and identification of 3D objects in partially scanned and cluttered point clouds," *IEEE Trans. Multimedia*, vol. 24, pp. 2230–2245, Jun. 2022.

[58] K. Wang, L. Sheng, S. Gu, and D. Xu, "VPU: A video-based point cloud upsampling framework," *IEEE Trans. Image Process.*, vol. 31, pp. 4062–4075, 2022.

Mengyu Yang received the B.E. degree from Sun Yat-sen University, Guangzhou, China, in 2021, where she is currently pursuing the M.S. degree with the School of Computer Science and Engineering. Her current research interests include multimedia communication and edge computing.

Zhenxiao Luo received the B.E. degree from the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering and the Guangdong Key Laboratory of Big Data Analysis and Processing. His research interests include multimedia with machine learning and federated learning in edge computing.

Miao Hu (Member, IEEE) received the B.S. and Ph.D. degrees in communication engineering from Beijing Jiaotong University, Beijing, China, in 2011 and 2017, respectively. He was a Visiting Scholar with the Pennsylvania State University, State College, PA, USA, from 2014 to 2015. He is currently an Associate Professor with the School of Computer Science and Engineering and the Guangdong Key Laboratory of Big Data Analysis and Processing, Sun Yat-sen University, Guangzhou, China. His research interests include edge computing, federated learning, and multimedia system.

Min Chen (Fellow, IEEE) has been a Full Professor with the School of Computer Science and Engineering, South China University of Technology. He is also the Director of Embedded and Pervasive Computing Laboratory, Huazhong University of Science and Technology (HUST). He was an Assistant Professor with the School of Computer Science and Engineering, Seoul National University before he joined HUST. His Google Scholar Citations reached over 38 800 with an H-index of 93. His top paper was cited over 4090 times. He was selected as a Highly Cited Researcher from 2018 to 2022. He got IEEE Communications Society Fred W. Ellersick Prize in 2017, the IEEE Jack Neubauer Memorial Award in 2019, and IEEE ComSoc APB Oustanding Paper Award in 2022. He is the Chair of IEEE Globecom 2022 eHealth Symposium. He is the Founding Chair of IEEE Computer Society Special Technical Communities on Big Data. He is a Fellow of IET.

Di Wu (Senior Member, IEEE) received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 2000, the M.S. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2003, and the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, Hong Kong, in 2007. He was a Postdoctoral Researcher with the Department of Computer Science and Engineering, Polytechnic Institute of New York University, Brooklyn, NY, USA, from 2007 to 2009, advised by Prof. K. W. Ross. He is currently a Professor and the Associate Dean with the School of Computer Science and Engineering and the Guangdong Key Laboratory of Big Data Analysis and Processing, Sun Yat-sen University, Guangzhou, China. His research interests include edge/cloud computing, multimedia communication, Internet measurement, and network security. He was the recipient of the IEEE INFOCOM 2009 Best Paper Award and IEEE Jack Neubauer Memorial Award.