Anthony Chen, Shiwen Mao *Auburn University, Auburn, AL* **Zhu Li** *Department of Computer Science and Electrical Engineering, NSF I/UCRC Center for Big Learning, University of Missouri–Kansas City, MO* **Minrui Xu** *School of Computer Science and Engineering, Nanyang Technological University, Singapore* **Hongliang Zhang** *School of Electronics at Peking University* **Dusit Niyato** *School of Computer Science and Engineering, Nanyang Technological University, Singapore* **Zhu Han** *Electrical and Computer Engineering Department, Computer Science Department, University of Houston, TX*

Editors: Shiwen Mao and Michelle Gong

AN INTRODUCTION TO POINT CLOUD COMPRESSION STANDARDS

Illustration, istockphoto.com

The prevalent point cloud compression (PCC) standards of today are utilized to encode various types of point cloud data, allowing for reasonable bandwidth and storage usage. With increasing demand for high-fidelity three-dimensional (3D) models for a large variety of applications, including immersive visual communication, Augmented reality (AR) and Virtual Reality (VR), navigation, autonomous driving, and smart city, point clouds are seeing increasing usage and development to meet the increasing demands. However, with the advancements in 3D modelling and sensing, the amount of data required to accurately depict such representations and models is likewise ballooning to increasingly large proportions, leading to the development and standardization of the point cloud compression standards. In this article, we provide an overview of some topical and popular MPEG point cloud compression (PCC) standards. We discuss the development and applications of the Geometry-based PCC (G-PCC) and Video-based PCC (V-PCC) standards as they escalate in importance in an era of virtual reality and machine learning. Finally, we conclude our article describing the future research directions and applications of the PCC standards of today.

With the recent rapid development in communication and computation, Augmented Reality (AR), Virtual Reality (VR), and Mixed Reality (MR) are proceeding to make waves in the consumer technology market. With the avant-garde VR devices being capable of reproducing near-lifelike three-dimensional (3D) scenes and videos, it is of utmost importance to be able to accurately represent 3D objects, create various scenes and worlds within VR, and display 3D videos for the VR markets. Allowing users to navigate well-modeled spaces is currently one of the top priorities of developers of VR technology. Not to mention that the 3D modeled objects or scenes come with various caveats of having numerous changing attributes as users interact with their world, such as changing positions, colors, or sizes. To fill such a niche, the utility of point clouds has been rising as a vital option to generate such high-fidelity 3D representations and a key option in the 3D modeling scene. These point clouds have the ability to model various 3D objects or scenes, whether they are dynamic, static, or even part of a video. However, the usage of point clouds comes at a cost. Due to the sizeable nature of these point clouds, each with various attributes attached to them, point clouds cost a large amount of memory or transmission bandwidth, making them unwieldy to implement. These point clouds can grow to be millions of points within a single frame, which, when running at 30 fps, can lead to bandwidth usage on the order of several Gbps if uncompressed [1]. In order to meet the requirements of state-of-the-art VR devices, these point clouds may need to generate even higher resolution, attach more attributes, or create even larger scenes, resulting in exponential increases in both storage and transmission size. In order to abate these restrictions, compression technology has been applied to these point clouds to make them more easily transmittable and take up less space.

Currently, the Moving Pictures Experts Group (MPEG) has been developing various standards for compression of immersive media. The standards developed by MPEG have been applied over the past few decades across all kinds of devices with popular compression standards such as MPEG-2, AVC, and HEVC [7]. Along with these standards, MPEG has also been developing standards for point cloud compression (PCC) to aid in its development. These standardization activities are centered on the improvements on the codecs developed by the MPEG group, creating the original PCC standard, which included LIDAR PCC (L-PCC) for dynamically acquired data, Surface PCC (S-PCC) for static point cloud data, and finally, Video-based PCC (V-PCC) for dynamic contents. However, post 2017, the categories were officially compressed to geometry-based PCC (G-PCC) combining Lidar PCC and Surface PCC due to their similarities, while Video-based PCC remained the same [3]. These categories generated corresponding test models named TMC13 and TMC2, respectively, with the G-PCC combining categories 1 and 3 of test

TABLE 1. Point Cloud Categories

Categories	Point Cloud Technology	Data Type	Test Model	Example Applications
1	LIDAR	Static	TMC13	Static Objects
2	Video	Dynamic	TMC2	3D Videos
3	Mesh Models	Dynamically Acquired	TMC13	Mobile Mapping

TABLE 2. G-PCC versus V-PCC

Standard	PCC Data Type	Test Model	Deployability	Compression Dimension	ISO	Coding
G-PCC	Static or dynamically acquired	TMC13	Slow market deployability	3D	In development	Arithmetic Coder
V-PCC	Video	TMC2	Fast market deployability	2D	Published	Traditional Video Coder

FIGURE 1. The general PCC system structure.

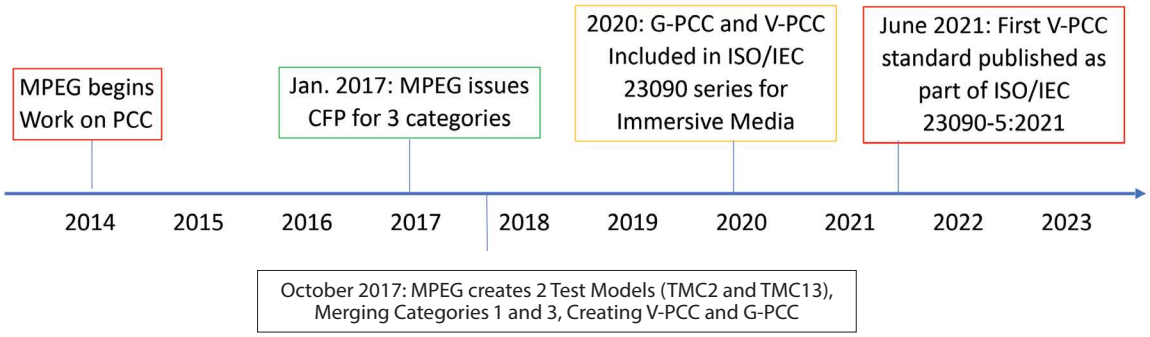


FIGURE 2. Timeline of major PCC standardization activities.

models (static and dynamically acquired) and V-PCC consisting of only TMC2. These two standards were released as part of ISO/IEC 23090-5 as well as 23090-9 as part of Immersive Media (MPEG-I) in 2020 [1,2,3]. The main features of each of these categories are summarized in Table 1.

In this article, we first introduce the concept of point cloud compression in "Overview of PCC." We then introduce the development of Geometry-based Point Cloud Compression as well as their standardization activities in "Geometry Based Point Cloud Compression." Next, we introduce and analyze the growth of Video based point cloud compression in "Video Based Point Cloud Compression." Finally, we conclude this article with a discussion of the various applications and future trends of rising PCC standards in "Applications/Future Trends."

OVERVIEW OF PCC

Point Clouds are collections of points in a 3D space, with each point being assigned a collection of attributes. From the standards currently being developed by MPEG, each point has been assigned three types of attributes, including a 3D coordinate (x, y, and z), the reflectance, and the RGB (red, green, blue) attributes associated with each point [4]. Among these attributes, the 3D coordinates of each point are usually characterized by floating point values, however these values can be quantized into integers through the usage of voxelization. Voxelization can be seen as a process of breaking down a 3D space into equal-sized cubes, or voxels, and each point within the cube is mapped to the center of such voxel. This process is then repeated d times, creating d levels of detail (LoD) of a 1×1×1 cubic root voxel.

To represent scenes and objects, these point clouds are generally acquired through the usage of cameras or sensors, and then processed in a computer. However, these point clouds have wide ranges of properties depending on their applications. To this end, the MPEG standards have categorized them as static point clouds with high details and large numbers of points, dynamic point clouds with fewer points but including time, and dynamically acquired point clouds, with large numbers of points and more attributes.

Although these point clouds can each be categorized into different data types and have vastly different properties, their overarching structure remains the same. The PCC general architecture draws much similarity to video coding, consisting of an encoder to compress the point cloud and make it easier to transmit or store, as well as a decoder to decompress and revert the point cloud to its original state, as shown in Figure 2. These point clouds can be compressed either with lossy coding or lossless coding, depending on the need for further compression or for preserving image/video quality. In order to measure the quality of PCC, the standard typically tests for image distortion, such as point-to-point distortion (D1) or point-to-plane distortion (D2), and through the use of RD (rate distortion) curves to demonstrate compression performance on various test cases [4].

The standardization of PCC originally began in 2013, when the MPEG group first decided to utilize point clouds in immersive applications. However, these standards were made for computer animated content and were thus not suitable for point cloud usage in real-time systems. This led to the MPEG group issuing a Call for Proposals in early 2017 [9], categorizing the state-of-the-art PCC technologies into LIDAR, Surface, and Video PCC. Test models were developed later in 2017, merging the two categories LIDAR and Surface PCC into Geometry-based PCC and keeping Video PCC. Later in 2020, S-PCC and L-PCC were officially merged into G-PCC as part of the immersive media standard. Finally, in 2021, the V-PCC standard was published as part of ISO 23090-5 as part of the immersive media standard. The G-PCC standard is still under development and has not been published yet. Development for these two codecs has been steadily progressing, with new versions of the codecs releasing every few months since their first version. The 2nd edition G-PCC codec, which was developed in mid-2022, achieved improvements to the coding schemes and transformations used within the codec. A general timeline of the major standardization activities can be seen in Figure 2. A comparison of G-PCC and V-PCC is provided in Table 2.

GEOMETRY BASED POINT CLOUD COMPRESSION

The G-PCC standard currently deals with all the point cloud compression data other than the video PCC format. It was originally developed about 2017-18 but was only officially introduced in 2020. Due to the similarities between the encoders and decoders of the L-PCC and S-PCC, these two were combined into the G-PCC. G-PCC now consists of all point cloud types ranging from point clouds for dynamically acquired data, e.g., for autonomous navigation, to detailed point clouds of static objects. The main feature of G-PCC is that it is coded directly in 3D without any changes in dimension like the V-PCC. Although the G-PCC does not allow for the usage of previous encoder technology, such as the video codecs, it has a lot of potential to be exploited, as G-PCC is open to many future developments in its geometric or attribute coding schemes. The G-PCC codec structure is presented in Figure 3, which, while not showing all the modules of the TMC13, provides a general overview of the codec. With each new version release of the G-PCC, advances have been made to refine various portions of the geometry or attribute coding or introduce new techniques to make the coding more efficient. Each portion of the encoding and decoding procedure is detailed in the following [2].

Geometry Coding: Geometry coding involves the procedure of processing and compressing the positions of the points within the point cloud. Because the positions begin as floating-point numbers within an original/world coordinate system, these positions must be pre-processed to be usable. This is done through coordinate transformation and voxelization, i.e., assigning each point to a voxel. Afterwards, the geometry of the structure is analyzed through the use of octree coding [3] or surface approximation utilizing trisoup [4]. Octree coding can be thought of as occupancy mapping of voxels, whereas occupied voxels are represented by 1s and unoccupied voxels are represented by 0s, and this is repeated recursively per subcube. These nodes are then further compressed through the usage of entropy coding. On the other hand, trisoup represents an object's surface as a series of triangular meshes (therefore creating a triangular soup) and is an optional coding option usually utilized in point clouds with dense surfaces. Recent advances in sparse convolutional engines [11] have also enabled very effective learning-based geometry coding, as represented by PU-Dense [12], Sparse Conv Point Cloud Compression (PCGC) [14], and learning based dynamic point cloud compensation and compression [15].

Attribute Coding: Regarding attributes in the MPEG standard, the standard only consists of RGB (color) and reflectance. Attribute coding represents the compression of these two attributes for each point within the point cloud. The first step of this process is an optional transformation, where users are given the option to transform colors in the point cloud from RGB to YCbCr (luminance, the blue-difference, and red-difference chroma components). Afterwards, the attributes are transferred to the reconstructed point cloud geometry (compressed, reconstructed, then decompressed points) and are either transformed using the Predicting Method, the Regional Adaptive Hierarchical Transform (RAHT), or Lifting Method. The predicting and lifting methods both generate and utilize a Level of Detail (LoD) representation whereas the Predicting Method encodes attributes based on a prediction of the LoD order, while the Lifting Method is built on top of the Predicting Method and utilizes an update operator as well as an adaptive quantization strategy. On the other hand, RAHT encodes attributes by spatially transforming them based on the octree hierarchy and then quantize them. An alternative implementation of RAHT was introduced by [5] as the Fixed-point implementation and is also being utilized along with a transform domain prediction technique to improve coding efficiency proposed by [6]. The type of transformation method used is typically selected based on the category of point cloud data (with RAHT typically being used in Category 1 and prediction/lifting typically being used in Category 3). Finally, after attribute transformation, the attribute coefficients are then quantized and encoded arithmetically like in the geometry coder. In recent years, learning based attributes coding schemes have shown promising results as shown by Deep-PAC [16] and octree based MLP predictive coding [17].

Decoding: Decoding of the geometry and positions is essentially performed in inverse of the encoder, where the geometry is arithmetically decoded, the octree or surface approximation is synthesized, the geometry is then reconstructed from the synthesis, and the coordinate transformation is inverted, resulting in the original positions. Attributes become decoded similarly through arithmetic decoding, then inversely

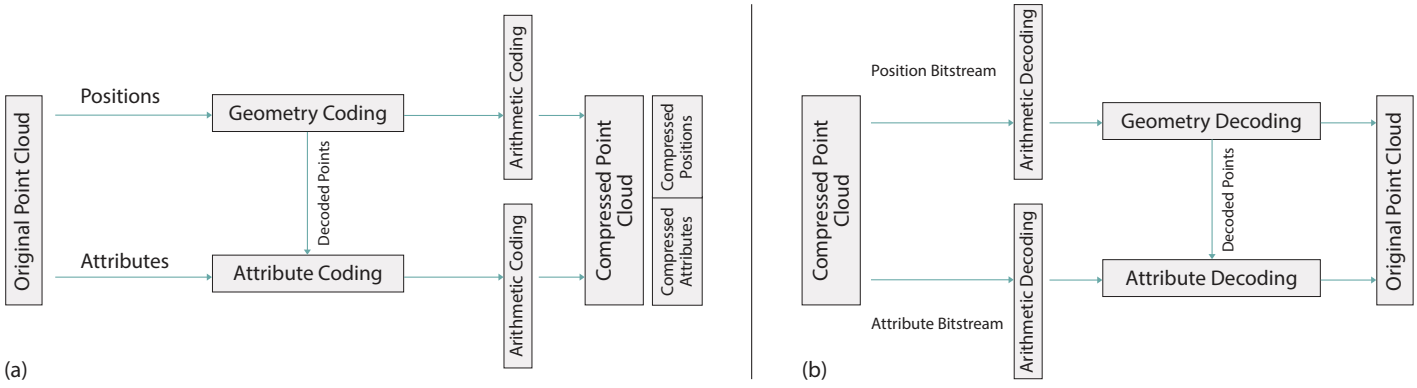


FIGURE 3. The structure of the G-PCC codec: (a) encoder, (b) decoder.

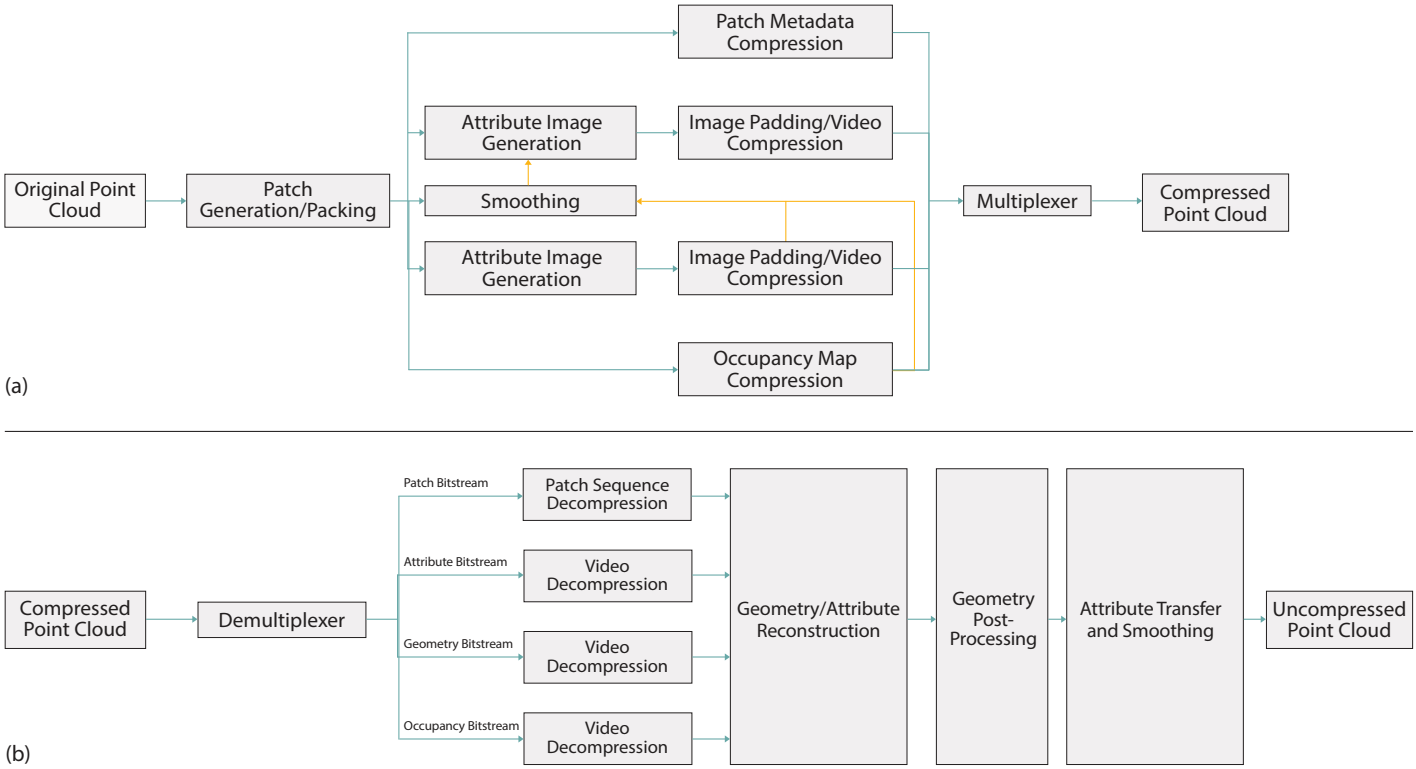


FIGURE 4. The structure of the V-PCC codec: (a) encoder, (b) decoder.

quantized, and inversely transformed; if the color was transformed to YCbCr, it can be transformed to RGB next, resulting in the original attributes. The uncompressed point cloud can be obtained through the usage of the decoded attributes and positions.

VIDEO BASED POINT CLOUD COMPRESSION

The V-PCC is based on the Visual Volumetric Video-based Coding (V3C). The V-PCC, on the other hand, has faster deployment in the market than G-PCC and makes use of video-coding properties of point cloud videos. V-PCC essentially splits videos into a geometric video sequence and a texture video sequence utilizing existing video codecs, such as MPEG 4, AVC, and HEVC. Unlike S-PCC, L-PCC, and G-PCC, patch generation is a big part of V-PCC encoding. Although the V-PCC encodes the attributes and positions separately, the codec performs its encoding and decoding in the 2D plane rather than 3D as in G-PCC, to allow for usage of existing video codecs. To do so, the encoder first generates and packs the original point cloud into a set of 3D patches and independently projects each patch to the 2D plane, where each patch takes up a unique location in a W×H grid. Afterwards, the geometry and attribute images are generated, padded with images, and compressed utilizing general video codecs. The reconstructed geometry may also be smoothed as part of a post-processing step. In addition to these two typical compressed bit streams, the V-PCC also generates an occupancy map and auxiliary patch information before multiplexing all of the bitstreams, as shown in Figure 4. Afterwards, the decoding will be much simpler, with de-multiplexing into decompression of each compressed bitstream, geometry and attribute reconstruction, geometry reconstruction steps, and ending with attribute transfer and smoothing, finally resulting in the uncompressed point cloud, as demonstrated in Figure 5. These V-PCC codecs are typically evaluated on common test cases, such as the Longdress dataset, RedandBlack dataset, and Soldier dataset [4]. The architecture of the codec is further elaborated in the following [1].

Patch Generation and Packing: Usage of patches is the key to utilizing video codecs for V-PCC, where 3D objects are mapped to series of 2D projections known as patches. The main objective of patch generation and packing is the generation and processing of 2D projections with low distortions and without loss to geometry or attributes. V-PCC accomplishes this by applying a heuristic segmentation process, creating a set of smooth patches. However, because the 3D patches can be mapped onto the same location in 2D, these patches are stored on different maps: a close map and a far map depending on depth. The patches are then packed, using a simple iterative strategy to insert patches onto a user-specified W×H grid and minimizing unused space. These patches are then used in image generation, geometry smoothing and reconstruction, occupancy map compression, and auxiliary patch information compression.

Image Generation/Padding: Image generation aims to generate the geometry and texture of the projections as images. The image generation process exploits the 2D projection process from the patch packing process to store the geometry and attributes as images. On the other hand, the padding process fills in the blank spaces

between patches to generate a smooth image. This is accomplished by checking blocks of pixels for empty space and copying previous rows or columns of pixels as replacement.

Occupancy Map Coding: The occupancy map is used to indicate whether a space in the image is filled or empty. This is a binary image where 1s represent at least 1 "occupied" pixel in the corresponding B×B block in the image, while 0s represent empty space in the B×B blocks. This occupancy map is usually lossless but is able to be coded lossy. To reduce excessive bit rate cost, the map is usually coded at 4×4 pixel resolution, and further improved utilizing deep learning-based occupancy map super resolution [18] to improve reconstruction quality.

Auxiliary Patch Coding: The auxiliary patch coding aims to compress the extra metadata required to reconstruct the 3D point cloud from the 2D images. This metadata includes the 3D location, 2D bounding box, and projection plane index for each patch as well as the patch index of each T×T block. This metadata is compressed and coded separately from the other bitstreams.

Smoothing and Reconstruction: The smoothing process is a post-processing procedure in the encoder that aims to remedy any potential discontinuities at patch boundaries. One approach to smoothing includes taking points at the boundaries of patches and moving the points to the centroid of its closest neighbors. On the other hand, the reconstruction process utilizes the occupancy map and auxiliary metadata to calculate the original 3D positions of the points to reconstruct the geometry and attributes of the original point cloud. Deep learning-based solutions for V-PCC deblocking have shown to achieve considerable performance gains, as demonstrated by sparse convolutional engine-based compression artifacts removal [13], as well as geometry [19] and attribute [20] deblocking.

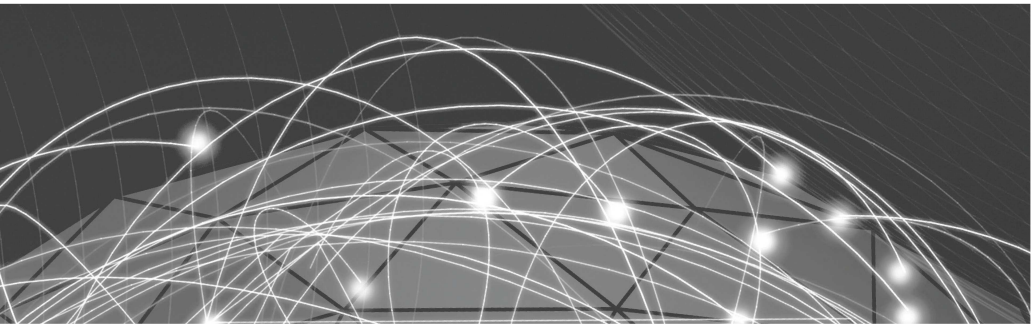
APPLICATIONS/FUTURE TRENDS

Point cloud compression has many applications, including cultural heritage, autonomous driving, VR/AR/XR, and metaverse. Regarding cultural heritage, many historical objects are stored virtually through the usage of high precision static point clouds. These objects are scanned using high precision 3D sensors and turned into point clouds so they can be shared and preserved in a digital format in places like historical museums and buildings. With the many advances being made in autonomous driving and modern vehicles becoming almost fully autonomous, these vehicles are equipped with a variety of sensors to acquire information on their surrounding environment, such as LIDAR sensors. These LIDAR sensors can continuously acquire the 3D environment surrounding such vehicles for creation of a time-varying point cloud with G-PCC usage to improve transmission and storage. Finally, in recent years, the VR/AR/MR field has been making large strides in innovation, generating larger and larger words (and the metaverse) with even more accurate virtual representations. This has led to a rise in immersive 3D videos as well as improved modelling explosively improving image quality, utilizing point clouds for high precision 3D content simulation and visualization. To enhance this immersive video experience, V-PCC has been applied to effectively compress and recreate these time-varying 3D models and videos in real time.

Currently, with the establishment of the V-PCC in its first edition ISO, development on the V-PCC has started slowing down. However, due to the nature of the V-PCC, as more video codecs are developed and become more efficient through machine learning and hardware acceleration, the V-PCC will become more efficient as a result. On the other hand, G-PCC is still seeing constant version updates by MPEG and its respective ISO standard, 23090-9, is still under development and should be published in the near future.

Due to the uniqueness and novelty of the standard category, G-PCC still has many improvements and potential yet to be capitalized on. One such potential improvement is the usage of machine learning in G-PCC, utilizing neural networks to improve prediction in voxel occupancy prediction or utilizing deep learning frameworks to improve compression performance. Other solutions have proposed fractional super-resolution [6] utilizing look-up tables to improve voxelized point cloud precision. These machine learning-based approaches tend to outperform the traditional G-PCC codec in terms of Rate-Distortion (RD) performance at the cost of CPU or GPU runtimes and complexity. With the advent of Deep Learning-based PCC, development in this area has approached improvements on PCC in many different ways, such as joint encoding or separate encoding of geometry and attributes, use of lossy or lossless compression, encoding based on voxels or points, etc. [10]. With the aid of neural networks, such as CNNs or RNNs, many PCC codecs have been optimized to significantly reduce distortion and improve RD performance. The application of deep learning has shown to improve PCC in comparison to MPEG's standard codecs at the cost of computational complexity [8].

IN THIS ARTICLE, WE PROVIDE AN OVERVIEW OF SOME TOPICAL AND POPULAR MPEG POINT CLOUD COMPRESSION (PCC) STANDARDS



Illustration, istockphoto.com

CONCLUSION

This article provided an overview of the current MPEG point cloud compression standards. The applications, a brief history and development of G-PCC and V-PCC were discussed, detailing the progression of recent PCC standards. With the recent interests in producing realistic 3D models and the requirement for efficient bandwidth usage as well as data storage, point cloud compression has been vital in the development of reliable 3D visualization and modeling services. PCC has proven to be a solid choice to effectively render in 3D for immersive media. ■

Anthony Chen received his MS in electrical and computer engineering from Clemson University, Clemson, S.C., in 2022. Currently, he is pursuing his PhD in the Department of Electrical and Computer Engineering at Auburn University, Auburn, AL. His research interests include wireless communications, deep learning, and optimization.

Shiwen Mao received his PhD in electrical and computer engineering from Polytechnic University, Brooklyn, N.Y., in 2004. Currently, he is a professor and Earle C. Williams Eminent Scholar at Auburn University, Auburn, AL. His research interests include wireless networks, multimedia communications, and smart grid. He is a recipient of many awards, including the 2023 SEC Faculty Achievement Award, and several service awards and best journal/conference paper awards from the IEEE. He is a Fellow of the IEEE and a Life Member of the ACM.

Zhu Li received his PhD in electrical and computer engineering from Northwestern University in 2004. From 2012 to 2015, he was the Senior Staff Researcher/Senior Manager with Samsung Research America's Multimedia Standards Research. He is currently a professor with the Department of Computer Science and Electrical Engineering, University of Missouri, and the Director of the NSF I/UCRC Center for Big Learning, UMKC. His research interests include point cloud and light field compression, graph signal processing, and deep learning.

Minrui Xu received his BS degree from Sun Yat-Sen University, Guangzhou, China, in 2021. He is currently working towards his PhD in the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests mainly focus on metaverse, mobile edge computing, deep reinforcement learning, and mechanism design.

Hongliang Zhang received his PhD at the School of Electrical Engineering and Computer Science at Peking University in 2019. He is currently an assistant professor in School of Electronics at Peking University. His current research interests include reconfigurable intelligent surfaces, aerial access networks, optimization theory, and game theory. He is a recipient of 2021 IEEE Comsoc Heinrich Hertz Award for Best Communications Letters and 2021 IEEE ComSoc Asia-Pacific Outstanding Paper Award.

Dusit Niyato is a professor in the School of Computer Science and Engineering, at Nanyang Technological University, Singapore. He received his BEng from King Mongkuts Institute of Technology Ladkrabang (KMITL), Thailand in 1999 and PhD in Electrical and Computer Engineering from the University of Manitoba, Canada in 2008. His research interests are in the areas of sustainability, edge intelligence, decentralized machine learning, and incentive mechanism design.

Zhu Han received his PhD in electrical and computer engineering from the University of Maryland, College Park in 2003. Currently, he is a John and Rebecca Moores Professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, TX. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. He has received many awards, including the IEEE Leonard G. Abraham Prize in the field of Communications Systems in 2016. He is an IEEE fellow, an AAAS fellow, and an ACM Distinguished Member.

REFERENCES

[1] "V-PCC Codec Description," document ISO/IEC JTC 1/SC 29/WG 11 MPEG, N 19526, 3D Graphics, Sept. 2020.

[2] "G-PCC Codec Description," document ISO/IEC JTC 1/SC 29/WG 7 MPEG, N00271, 3D Graphics, Apr. 2020.

[3] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai. April 2020. An overview of ongoing point cloud compression standardization activities: Video-based (V-PCC) and geometry-based (G-PCC) *APSIPA Transactions on Signal and Information Processing*, vol.9, e13, 1–17.

[4] S. Schwarz et al. March 2019. 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.

[5] E. S. Jang et al. May 2019. Video-based point-cloud-compression standard in MPEG: Evidence collection to committee draft [Standards in a Nutshell]. *IEEE Signal Processing Magazine*, vol.36, no.3, 118–123.

[6] Performance Analysis of Currently AI-based Available Solutions for PCC," document ISO/IEC JTC 1/SC 29/WG 7, N 00374m 3D Graphics, July 2022.

[7] T. Zhang and S. Mao. Dec. 2018. An overview of emerging video coding standards. *ACM GetMobile: Mobile Computing and Communications*, vol.22, no.4, 13–20.

[8] A.F.R. Guarda, N.M.M. Rodrigues, and F. Pereira. Feb. 21. Adaptive deep learning-based point cloud geometry coding. *IEEE Journal on Selected Topics in Signal Processing*, vol.15, no.2, 415–430.

[9] "Call for Proposals for Point Cloud Compression v2," document ISO/IEC JTC 1/SC 29/WG 11 MPEG, N16763, 3D Graphics, Apr. 2017.

[10] R. Hooda, W. D. Pan, and T. M. Syed. Mar./Apr.2022. A survey on 3D point cloud compression using machine learning approaches. *Proc. SoutheastCon 2022*, Mobile, AL, 522–529.

[11] C. Choy, J. Gwak, and S. Savarese. June 2019. 4D spatio-temporal convnets: Minkowski convolutional neural networks. *Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, 3075–3084.

[12] A. Akhtar, Z. Li, G. V. Auwera, L. Li, and J. Chen. June 2022. PU-Dense: Sparse tensor-based point cloud geometry upsampling. *IEEE Transactions on Image Processing*, vol.31, no.6, 4133–4148.

[13] A. Akhtar, W. Gao, L. Li, Z. Li, W. Jia, and S. Liu. Jan. 2021. Video-based point cloud compression artifact removal. *IEEE Transactions on Multimedia*, vol.24, no.1, 2866–2876.

[14] J. Wang, D. Ding, Z. Li, X. Feng, C. Cao, and Z. Ma. Sparse tensor-based multiscale representation for point cloud geometry compression. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, in press. DOI 10.1109/TPAMI.2022.3225816.

[15] T. Fan, L. Gao, Y. Xu, Z. Li, and D. Wang. July 2022. D-DPCC: Deep dynamic point cloud compression via 3D motion prediction. *Proc. Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)*, Vienna, Austria, 898–904.

[16] X. Sheng, L. Li, D. Liu, Z. Xiong, Z. Li, and F. Wu. June 2022. Deep-PCAC: An end-to-end deep lossy compression framework for point cloud attributes. *IEEE Transactions on Multimedia*, vol.24, no.6, 2617 –2632.

[17] L. Hou, L. Gao, Y. Xu, Z. Li, X. Xu, and S. Liu. Sept. 2022. Learning-based intra-prediction for point cloud attribute transform coding. *Proc. IEEE Workshop on Multimedia Signal Processing (MMSP)*, Shanghai, China. 1–6.

[18] W. Jia, L. Li, Z. Li, and S. Liu. May 2021. Convolutional neural network-based occupancy map accuracy improvement for video-based point cloud compression, *IEEE Transactions on Multimedia*, vol.24, no.5, 2352–2365.

[19] W. Jia, L. Li, Z. Li, and S. Liu. Aug. 2021. Deep Learning geometry compression artifacts removal for video-based point cloud compression, *Springer International Journal of Computer Vision*, vol.129, no.11, 2947–2964.

[20] P. Wang, S. Wang and Z. Li. Mar. 2023. Occupancy map guided attributes deblocking for video-based point cloud compression, *Proc. 2023 IEEE Data Compression Conference (DCC)*, Snowbird, UT.