REAL-TIME DECODING AND AR PLAYBACK OF THE EMERGING MPEG VIDEO-BASED POINT CLOUD COMPRESSION STANDARD

S. Schwarz, M. Pesonen

Nokia Technologies, Finland

ABSTRACT

This paper presents the world's first implementation and release of the upcoming MPEG standard for video-based point cloud compression (V-PCC) on today's mobile hardware. As V-PCC offloads most of the computational burden on existing video coding solutions, real-time decoding can be achieved on essentially every single media device on the market. Furthermore, as the infrastructure for 2D video distribution is well established, distribution and storage of V-PCC content can already be achieved on today's networks.

INTRODUCTION

Due to the increased popularity of augmented and virtual reality experiences, the interest in capturing the real world in multiple dimensions and in presenting it to users in an immersive fashion has never been higher.

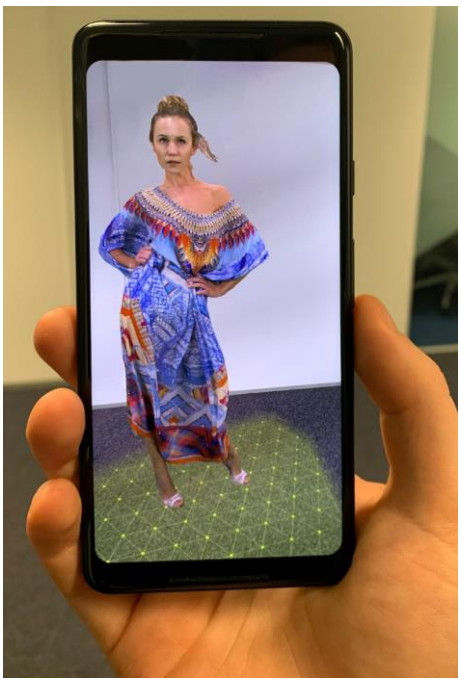

Figure 1 – Real-time decoding and AR playback of dynamic point cloud data.

Volumetric visual data represents dynamic 3D scenes and allows a user to freely navigate within. Unfortunately, such representations require a large amount of data and uncompressed transmission is not feasible over today's networks. Therefore, the Moving Picture Expert Group – MPEG (ISO/IEC JTC1/SC29/WG11), as one of the main standardization groups dealing with multimedia, recently started an ambitious roadmap towards immersive media compression. One of the elements of this roadmap is a standard for compressing dynamic 3D point clouds. This emerging standard, ISO/IEC 23090-5 (1), will rely heavily on the use of already available 2D video coding technology. Thus, claiming superior compression efficiency and accelerated time-to-market (2).

Following this claim, this paper presents the world's first implementation and source code release of the upcoming MPEG standard for video-based point cloud compression (V-PCC) on current mobile hardware, as shown in Figure 1. Because V-PCC offloads a lot of the computational burden on existing video coding solutions, hardware video decoders, available in

essentially every single media device on the market, can be utilised for real-time decoding. Furthermore, as infrastructure for 2D video distribution is well established, such as ISOBMFF and DASH, these existing solutions can be easily used to support distribution and storage of V-PCC content. In this paper, we will address topics such as V-PCC compression efficiency compared to state-of-the-art, decoding and rendering capabilities on current mobile clients, as well as approaches for V-PCC content distribution over existing 2D video infrastructure.

The remainder of the paper is structured as follows: First we provide a background overview on volumetric video compression. Then, we will present the upcoming MPEG standard for video-based point cloud compression (V-PCC) in its latest form, before going into details on the decoder implementation for real-time augmented reality (AR) playback, including performance evaluation on current mobile phone technology. The paper will conclude with an outlook on ongoing standardisation activities, such as V-PCC file encapsulation and streaming for distribution and storage.

COMPRESSION OF DYNAMIC 3D POINT CLOUD DATA

With the fast-rising number of 3D sensing technologies, dynamic 3D video is getting more important and practical for representing 3D contents and environments. For a long time, representing the world has been achieved using 2D video. Nowadays, there are plenty of devices which can capture and represent our world in a more immersible 3D fashion. For this task, dynamic 3D point cloud data is already used widely in environments such as virtual and augmented reality, 3D video streaming in mixed reality and mobile mapping.

Such dynamic 3D point cloud data consists of sets of points in 3D space where each point has its own spatial location along with a corresponding vector of attributes, such as texture, surface normals, reflectance, etc. And point cloud data produced by some high-resolution sensors, e.g. Microsoft Kinect, may contain several millions of points. Thus, processing and transmitting such data is challenging.

In recent years many efforts have been devoted to promoting the efficiency of compression method algorithms for such 3D point cloud data. One of the first and main classes of point cloud compressors is progressive point cloud coding. In this approach, which is mostly based on kd-tree, a coarse representation of the complete point cloud is encoded and then several refinement layers are built to achieve efficient compression. A 3D space is split in half recursively using a kd-tree structure and points in each half are encoded. For each subdivision, empty cells are not subdivided anymore (3). Applying the same approach using an octree structure achieved improved results due to considering connectivity information (4).

Regarding compression of dynamic point cloud sequences, Thanou et al. proposed a first work dealing with dynamic voxelized point clouds. In their work a time-varying point cloud codec is introduced, using sets of graphs and wavelet-based features to find matches between points to predict an octree structure for adjacent frames (5). Kammerl et al. also addressed simplified geometry prediction for a sequence of octree structures (6). Unfortunately, 3D motion estimation is one of the challenging issues in point cloud

compression, due to the lack of information in point-to-point correspondences in a sequence of frames. A more recent work on motion estimation coding can be found in (7). And in (8) a progressive hybrid compression framework is introduced, focusing on mixed-reality and 3D tele-immersive systems. The approach combines Kammerl's octree-based approach from (6) with a common image coding framework for attribute compression. This solution was selected as reference technology for the MPEG Call for Proposal (CfP) for Point Cloud Compression (9).

ISO/IEC 23090-5: A STANDARD FOR VIDEO-BASED POINT CLOUD COMPRESSION

In April 2017, ISO/IEC JTC1/SC29/WG11 (MPEG) issued a CfP for point cloud compression technology (9). The responses were evaluated in October 2017 and two standard activities for point cloud compression have been formed: ISO/IEC 23090 Part 5: Video-based point cloud compression (V-PCC), addressing media-centric dynamic point cloud compression, based on current 2D video technology and ready for fast deployment, and ISO/IEC 23090 Part 9: Geometry-based point cloud compression (G-PCC), addressing general point cloud compression with a slightly longer timeframe until deployment. An overview on the CfP evaluation and the chosen standardisation paths is given in (2). Since then, MPEG is working hard towards deployment of these two standards. The Committee Draft (CD) for V-PCC was submitted for balloting in January 2019 and is expected to be finalised by the end of the year. The CD for G-PCC was submitted in May 2019.

As this paper is addressing the implementation of V-PCC, this section provides a summary of the technology. The V-PCC approach for compressing dynamic point cloud data utilizes existing 2D video compression and hardware technology to achieve efficient compression. This solution is based on projecting patches of 3D data, i.e. point clouds, onto 2D image planes and compressing these planes by any available 2D video codec.

V-PCC encoding process

The main philosophy behind V-PCC is to leverage existing video codecs for compressing the geometry and texture information of dynamic point clouds. This is essentially achieved by converting the point cloud into a set of different video sequences. In particular, three video sequences, one that captures the geometry information, one that captures the texture information of the point cloud data, and another that describes the occupancy in 3D space, are generated and compressed using existing video codecs, such as MPEG-4 AVC, HEVC, AV1, or similar.

Additional metadata, which are needed for interpreting the three video sequences, i.e. auxiliary patch information, are also generated and compressed separately. The video generated bitstreams and the metadata are then multiplexed together so as to generate the final V-PCC bitstream. It should be noted that the metadata information represents a relatively small amount of the overall bitstream. The bulk of the information is handled by the video codec. Figure 2 provides an overview of the V-PCC compression (top) and decompression (bottom) processes (2).

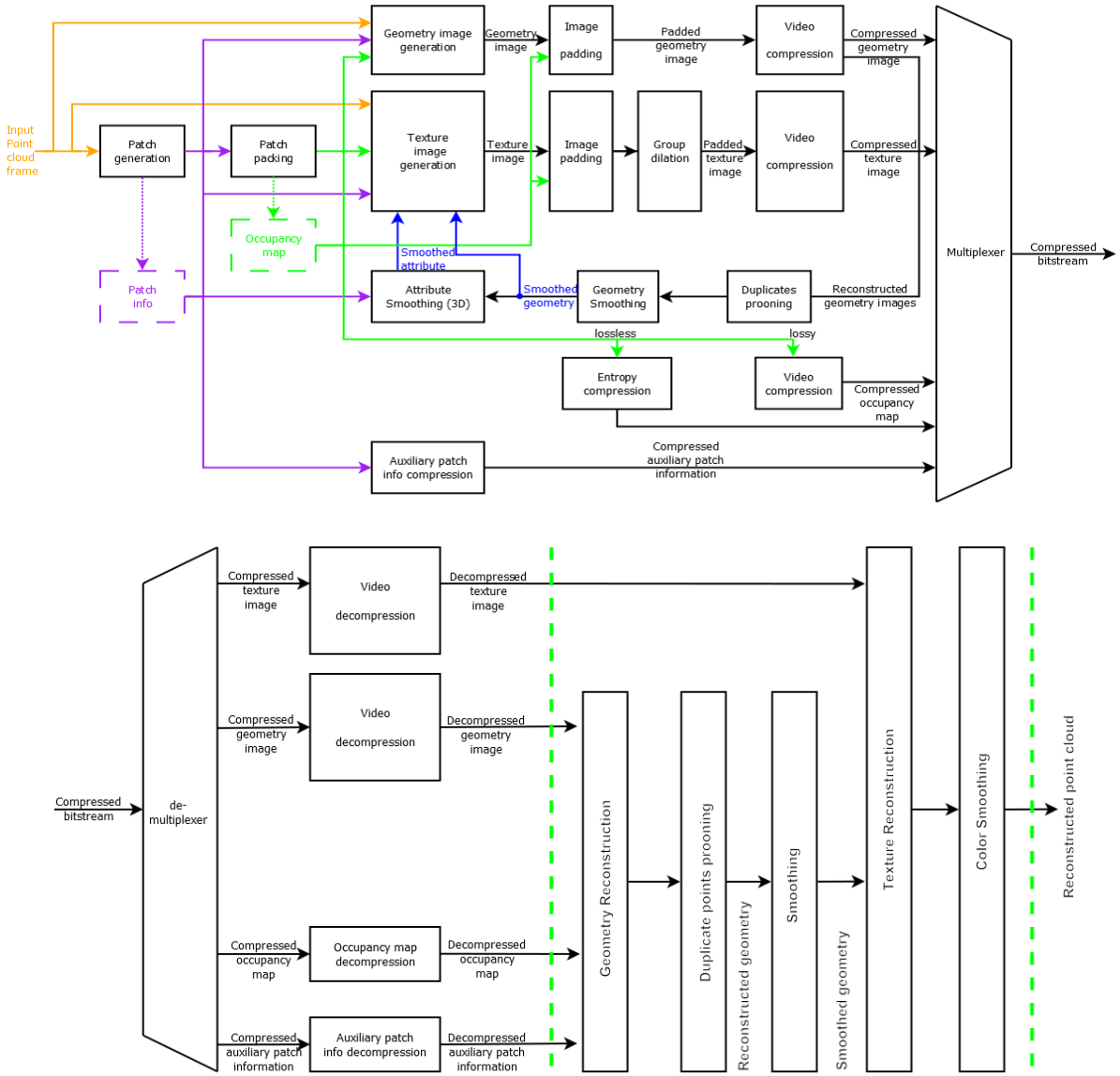

Figure 2 – V-PCC encoding (top) and decoding (bottom) process block diagrams (2).

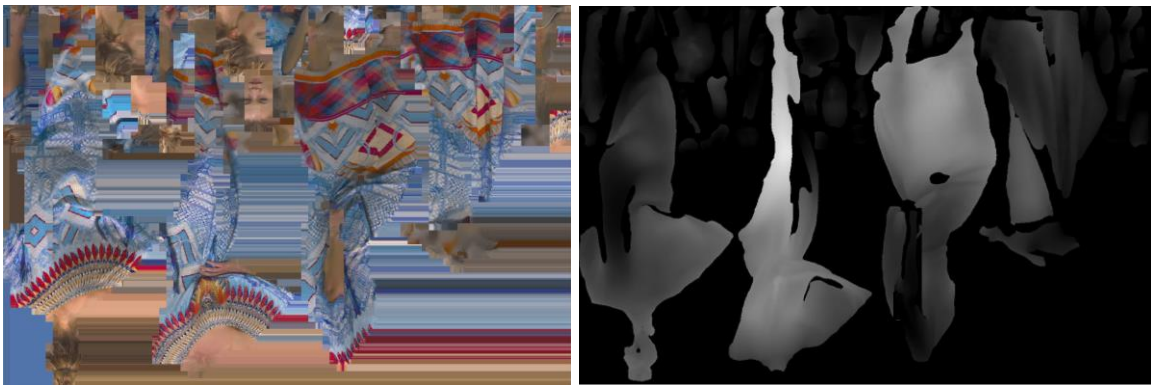

Figure 3 – Example results of the V-PCC patch generation after packing and padding.

Figure 4 – Original, anchor (8), and V-PCC decoded point cloud at 3.5 MBit/s.

Employing traditional 2D video codecs to encode dynamic 3D point clouds requires mapping the input point cloud to a regular 2D grid. The objective here is to find a temporally-coherent low-distortion injective mapping that would assign each point of the 3D point cloud to a cell of the 2D grid.

Maximizing the temporal coherency and minimizing the distance/angle distortions enables the video encoder to take full advantage of the temporal and spatial correlations of the point cloud geometry and attributes signals. An injective mapping guarantees that all the input points are captured by the geometry and attributes images and could be reconstructed without loss. Simply projecting the point cloud on the faces of a cube or on the sphere does not guarantee lossless reconstruction due to occlusions and generates significant distortions. Thus, input point clouds are decomposed into a minimum number of patches, 2D projections of the 3D data onto 2D surfaces, based on their surface normal information.

The image packing process maps the extracted patches onto a 2D grid, while trying to minimize the unused space. Image padding is applied to the texture and geometry images, to fill the empty space between patches to generate a piecewise smooth image better suited for video coding. The results of this process are shown in Figure 3.

For the decoder to be able to reconstruct the 3D point cloud from the 2D images, auxiliary patch metadata is encoded in the V-PCC bitstream, indicating the projection plane, 3D location and 2D image bounding box for each patch.

V-PCC decoding process

At the decoder, the received V-PCC bitstream is demultiplexed into the separate video bitstreams. Based on the decoded auxiliary patch information, the pixels in each 2D patch are remapped into 3D space based on their occupancy information, i.e. only points with a valid occupancy are mapped into 3D space. After the 3D reconstruction, geometry smoothing may be applied to alleviate potential discontinuities that may arise at the patch boundaries due to compression artefacts in the video coding process. The result is a reconstructed 3D point cloud, as shown in Figure 4 (right) for sequence *RedAndBlack*.

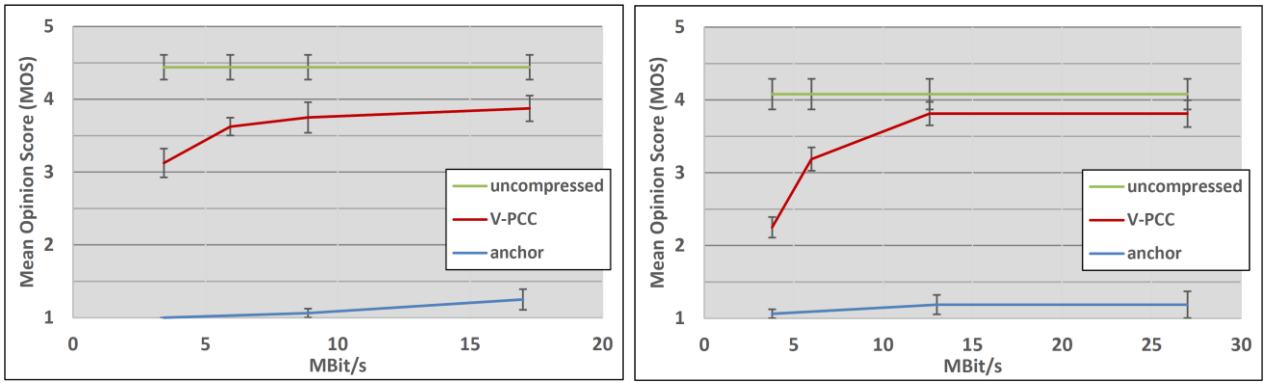

Figure 5 – Subjective evaluation results for sequences *RedandBlack* and *Longdress* (2).

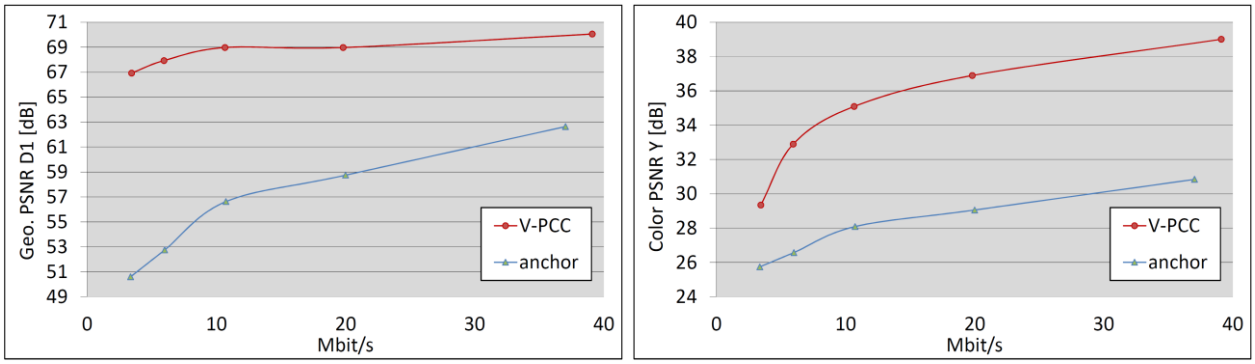

Figure 6 – Objective geometry and texture RD-curves for sequence *Soldier* (2).

V-PCC performance

The performance of V-PCC was originally addressed against the latest state-of-the-art given in (8) as anchor. As part of the MPEG CfP (9) evaluation, extensive objective and subjective assessments were carried out. An overview of the results is available in (2). As a brief summary, Figures 4 to 6 provide a subjective example of the V-PCC encoding performance, subjective evaluation results and objective rate-distortion curves at the time of the CfP evaluation, respectively.

The benefits of V-PCC over the anchor in terms of visual quality and coding performance are clearly visible. Even at the lowest target point, reasonable reconstruction quality is achieved. Depending on the sequence, compression factors between 1:100 to 1:500 are feasible with V-PCC.

V-PCC IMPLEMENTATION FOR REAL-TIME AR PLAYBACK

By employing standard 2D video coding technology, V-PCC benefits from several decades of video coding research and can rely on readily available hardware 2D video coding solutions and transmission infrastructure. Thus, billions of devices are already capable of decoding and displaying V-PCC content in real-time.

V-PCC enables exciting new user experiences, especially when combined with augmented reality, as shown in Figure 1. However, implementing V-PCC for current mobile devices

DEVICE	GPU / CHIPSET	FPS
iPhone 6S	Apple A9 GPU	15.4
iPhone 8	Apple A11	30.3
iPhone X	Apple A11	30.3
iPhone XS	Apple A12 GPU	30.3
Samsung Galaxy S10	Mali-G76 / Exynos 9820	25.0
Huawei Mate Pro 20	Mali-G76 / Kirin 980	23.3
Google Pixel	Adreno 530 / Snapdragon 821	29.4
Google Pixel 2 XL	Adreno 540 / Snapdragon 835	30.3
Nokia 8 Sirocco	Adreno 540 / Snapdragon 835	30.3
Google Pixel 3	Adreno 630 / Snapdragon 845	25.6
OnePlus (A6013)	Adreno 630 / Snapdragon 845	25.6

Table 1 – V-PCC decoding performance overview.

may be challenging. This section highlights some of the implementation difficulties experienced on Android and iOS platforms.

The source code for the V-PCC decoder and AR rendering application is now available for research purposes at (10).

V-PCC decoding performance

In typical 2D video player applications, only one video stream is decoded and displayed. Some stereo and VR stereo 360 video players handle and synchronise two video tracks, one track for each eye. However, V-PCC decoding requires the synchronisation of three video decoder instances. Unfortunately, current Android and iOS video decoders do not support adequate synchronisation of video streams. Video bitstreams are decoded on a “best effort" basis, depending on the available processing resources. The latest Android specification states in particular:

> *“[…] the contents of the texture object specified when the* `SurfaceTexture` *[video stream] was created are updated to contain the most recent image from the image stream. This may cause some frames of the stream to be skipped."*

This possible frame-skipping is a serious problem for V-PCC, as all three video frames are needed for 3D reconstruction. In typical 2D video display there is not much need for the application to know if a frame was skipped and displayed twice in a row. Especially, as the human eye adapts to such an event as it happens only rarely. For V-PCC decoding this is however a serious error, as it will destroy the complete 3D reconstruction. To avoid this issue and guarantee a high-quality V-PCC playback, sophisticated frame buffering is essential.

Furthermore, `SurfaceTexture` is currently used as target for decoded frames on Android. However, we discovered that `SurfaceTexture` holds multiple image planes and there is no direct access to the output image planes. This is not very optimal for V-PCC decoding. A better way would be a mode where direct access to the output buffers is available. We hope that upcoming Vulkan MediaCodec API could allow this kind of access. Video decoding on iOS has shown to be more reliable. Table I provides an overview of V-PCC decoding performance on current mobile hardware, e.g. decoding and synchronising three individual video decoder instances.

Release year	Device	GPU	Mpoints/frame @ 60 fps
2014	Samsung Galaxy Note 4	Adreno 420	1.04
2016	Samsung Galaxy S7 Edge	Mali-T880	0.43
2016	Google Pixel	Adreno 530	1.60
2017	Google Pixel 2	Adreno 540	1.81
2018/Q4	OnePlus 6T	Adreno 630	1.90
2018/Q4	Huawei Mate Pro 20	Mali-G76	1.32

Table 2 – V-PCC AR point rendering capabilities.

V-PCC AR rendering performance

Decoding the separate video streams is just one part of the V-PCC decoding pipeline. In order to fully evaluate how well a V-PCC standard can be implemented on current generation mobile devices, the V-PCC test model was modified and ported to Android and iOS platforms. A very minimalistic point cloud reconstruction process was selected, without any further postprocessing such as 3D smoothing. A GPU accelerated version of the 3D point reconstruction was implemented based on OpenGL ES 3.0. Table 2 summarises the achieved rendering capabilities on various mobile phones. The rendering performance is consistent between devices with the same GPU (see Table 1).

The 3D point reconstruction is calculated inside a vertex shader to save vertex memory bandwidth. Output of the shader is a uniformly distributed 1-pixel sized points, filling the display of the mobile device. In order to measure peak performance, the video dimensions (width, height) were increased to the point until the decoding performance drops below 57 fps. Typically, mobile devices can only sustain maximum performance for a short period of time. In order to measure more reliable benchmark results, the rendering benchmark is kept running for ten minutes and the overall average is reported as how many million points can be rendered in real-time (60 fps).

Typical V-PCC content currently consists of around 1 Million points per frame. Looking at the results from Tables 1 and 2, it can be seen that even 3-year-old hardware is already capable of decoding and rendering V-PCC bitstreams.

CONCLUSIONS & OUTLOOK

As ISO/IEC 23090-5 is closing in on to its publication as an international standard, this first V-PCC implementation is an important asset to prove its relevancy to the public. The V-PCC AR playback application source code has been made available to the public at (10) and will be further developed as the V-PCC standard has reaches codec stability, expected in Autumn 2019. This paper serves as a reference for this code release.

In the meantime, activities on efficient storage and streaming of V-PCC data have started. Again, these are benefiting from the underlying 2D video coding structure and utilising existing 2D video coding infrastructure solutions. For example, it is feasible to store each V-PCC video track as a separate track in the ISO base media file format (ISOBMFF). The auxiliary patch information is stored as timed metadata track and combined with the video

tracks as a V-PCC GroupListBox. This box provides the list of all tracks of the V-PCC content. Thus, a flexible V-PCC content configuration supporting a variety of client capabilities, e.g. multiple versions of encoded data components, can be stored in a file. Such content configurations can also be stored in an MPEG-DASH manifest (MPD) for dynamic adaptive streaming of V-PCC data over current video delivery infrastructure.

V-PCC is expected to be delivered as an international standard ISO/IEC 23090-5 in early 2020, with the accompanying technologies for storage and streaming shortly after. The publication of this real-time V-PCC AR playback application intends to promote the standard and highlight its importance for next-generation immersive media applications. We expect a follow-up application demonstrating real-time V-PCC AR streaming before the finalisation of the respective standards.

REFERENCES

1. ISO/IEC JTC 1/SC 29/WG 11, "Study Text of ISO/IEC CD 23090-5 Video-based Point Cloud Compression", N18180, January 2019.
2. S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuca, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan, A. Tabatabai, A. M. Tourapis, and V. Zakharchenko, "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, March 2019.
3. P. Alliez and M. Desbrun, "Progressive compression for lossless transmission of triangle meshes," in Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH ACM, 2001.
4. R. Schnabel and R. Klein, "Octree-based point-cloud compression," in Proceedings of the 3rd Eurographics / IEEE VGTC Conference on Point-Based Graphics, 2006.
5. D. Thanou, P. A. Chou, and P. Frossard, "Graph-based compression of dynamic 3D point cloud sequences," IEEE Transactions on Image Processing, vol. 25(4), pp. 1765–1778, 2016.
6. J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach, "Real-time compression of point cloud streams," in 2012 IEEE International Conference on Robotics and Automation, 2012.
7. R. L. de Queiroz and P. A. Chou, "Motion-compensated compression of dynamic voxelized point clouds," IEEE Transactions on Image Processing, vol. 26(8), pp. 3886–3895, 2017.
8. R. Mekuria, K. Blom, and P. Cesar, "Design, implementation, and evaluation of a point cloud codec for tele-immersive video," IEEE Transactions on Circuits and Systems for Video Technology, vol. 27(4), pp. 828–842, 2017.
9. ISO/IEC JTC 1/SC 29/WG 11, "Call for proposals for point cloud compression V2," N16763, April 2017.
10. Nokia Technologies, "Video Point Cloud Coding (V-PCC) AR Demo," https://github.com/nokiatech/vpcc