DELIVERING OBJECT-BASED IMMERSIVE MEDIA EXPERIENCES IN SPORTS

Fai Yeung[1], Basel Salahieh[1], Kimberly Loza[1], Sankar Jayaram[1], Jill Boyce[1]
Intel Corporation[1], Santa Clara, CA, 95054, USA.

***Abstract** – Immersive media technology in sports enables fans to experience interactive, personalized content. A fan can experience the action in six degrees of freedom (6DoF), through the eyes of a player or from any desired perspective. Intel Sports makes deploying these immersive media experiences a reality by transforming the captured content from the cameras installed in the stadium into preferred volumetric formats used for compressing and streaming the video feed, as well as for decoding and rendering desired viewports to fans' many devices. Object-based immersive coding enables innovative use cases where the streaming bandwidth can be better allocated to the objects of interest. The Moving Picture Experts Group (MPEG) is developing immersive codecs for streaming immersive video and point clouds. In this paper, we explain how to implement object-based coding in MPEG metadata for immersive video (MIV) and video-based point-cloud coding (V-PCC) along with the enabled experiences.*

Keywords – Immersive media, MPEG-I, point-cloud, six degrees of freedom (6DoF), volumetric video.

1. INTRODUCTION

Advancements in digital media technology are paving the way to innovations in delivering compelling immersive experiences with new media formats. The Moving Picture Experts Group (MPEG) is one of the main standardization groups driving the industry effort to develop a suite of standards to support immersive media access and delivery. The coded representation of immersive media (MPEG-I) is a set of emerging immersive media industry standards dedicated to immersive media formats such as panoramic 360-degree video, volumetric point clouds, and immersive video. Intel is working with other leaders in the immersive media industry to invest in the end-to-end immersive ecosystem to drive the worldwide adoption of immersive media. This includes everything from the production of immersive content to the delivery of innovative visual experiences and immersive services.

Part of Intel's immersive efforts is its active contribution to current state-of-the-art and emerging standards. One of its key collaborative efforts is object-based point cloud and immersive coding and their related supplemental messages adopted into MPEG-I industry standards. This provides a huge leap forward in how volumetric 6DoF video use cases are delivered.

The remainder of the paper is structured as follows: First we cover immersive media for sports by referencing the current immersive media platform architecture and workflow from Intel Sports in section 2. Then in section 3, we share the technical background for MPEG immersive coding (MIV) and video-based point-cloud coding (V-PCC) which are immersive media standards for compressing and streaming volumetric content. Section 4 covers in detail the implementation of the new object-based coding in MPEG-I V-PCC and MIV, as well as providing applications and services for the object-based approach. Section 5 then gives a conclusion of this paper.

2. IMMERSIVE MEDIA FOR SPORTS

Enabling sports fans to choose where and how they want to consume the next generation of sports content, whether they are 3DoF, 6DoF with limited motion range, or 6DoF immersive media experiences, makes sports an ideal use case for MPEG-I. In order to provide context on current industry practices for delivering immersive media experiences, we provide an overview of Intel Sports immersive work as an example.

2.1 Intel Sports immersive media platform

Intel Sports partners with leagues, teams, and broadcasters to give fans immersive media experiences allowing them to see any perspective of the play, personalizing the game experience according to each fan's preferences. The Intel Sports immersive media platform is comprised of two Intel proprietary capture systems: Intel® True View that outputs volumetric video and Intel® True VR for panoramic stereoscopic video.

The Intel Sports immersive media platform is designed to support a large number of streaming platforms and devices through a common video processing and distribution pipeline that is hosted in the cloud. The immersive media platform covers components of the volumetric and panoramic capture systems, as well as the immersive media processing and experiences pipeline.

2.1.1 Intel® True View for volumetric video

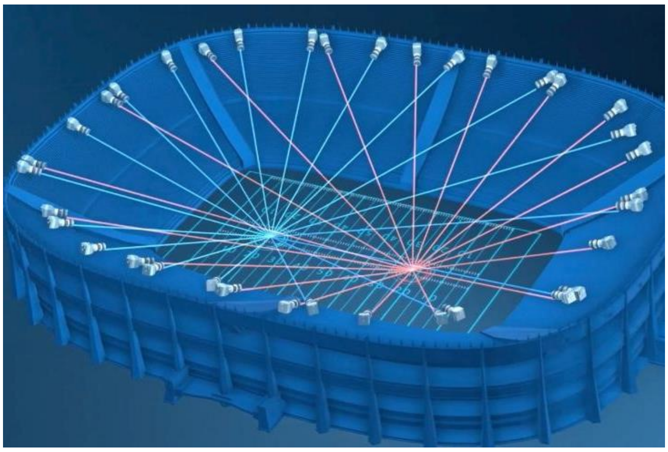

Fig. 1 – In-stadium Intel® True View cameras

Intel® True View is comprised of a large (~40) camera array that is built into the perimeter of the stadium as shown in Fig. 1. The high-definition 5K cameras are angled to capture the entire field of play. During an event the video captured at the stadium generates terabytes of volumetric data, in the form of three-dimensional pixels (voxels). The voxels contain three-dimensional location information and relative attributes (e.g., texture information) that are needed for point-cloud formation. The camera array is connected by fiber to dedicated on-site Intel servers. The data is stored, synchronized, analyzed, and processed in the cloud. The terabytes of volumetric data are rendered into high-fidelity video in the form of virtual camera videos.

2.1.2 Intel® True VR for panaromic video

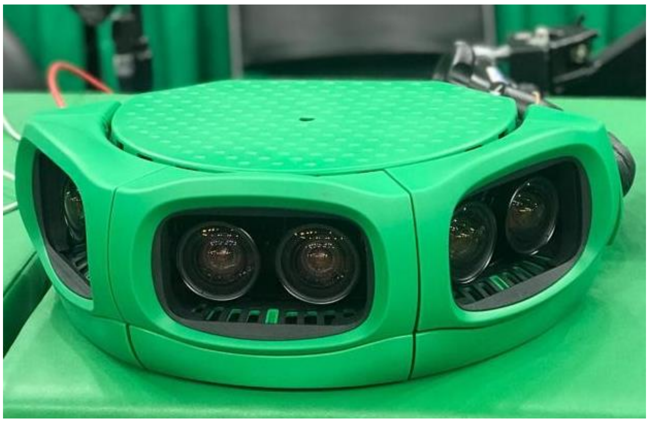

Fig. 2 – Intel® True VR camera pod

The Intel® True VR capture system captures video from multiple stereoscopic camera pods. The camera pods are Intel proprietary, paired-lens stereoscopic camera pods, as shown in Fig. 2, that are set up in the stadium for the live event. An event can have as many as eight camera pods, each pod including as many as 6 to 12 cameras that are stitched together. The stitched camera pod views are then turned into panoramic video that is sent to the common immersive media processing and distribution pipeline.

2.1.3 Immersive media processing and experiences pipeline

The immersive media platform architecture and workflow diagram shown in Fig. 3 showcases the Intel Sports immersive media platform that is powered by Intel® True VR and Intel® True View.

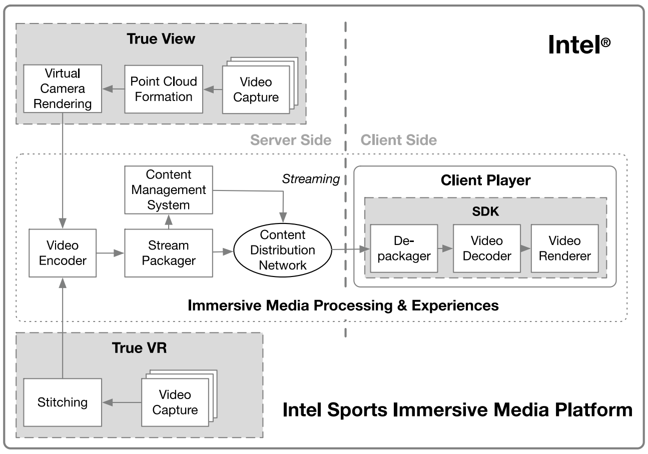

Fig. 3 – Immersive media platform architecture and workflow

Intel® True VR and Intel® True View share a common video processing and distribution pipeline that is hosted in the cloud.

The video processing component of the pipeline takes the uncompressed virtual camera videos from the Intel® True View capture and the panoramic video from the Intel® True VR capture then encodes the video sources.

The settings for stream types, codecs, compression quality, picture resolutions, and bitrates are highly dependent on the targeted end-user experiences and the expected network conditions. The system is flexible to support current industry standard codecs such as AVC, HEVC, and M-JPEG, and is flexible to utilize upcoming immersive media standards in order to maintain support of a wide range of platforms and devices.

The stream packager takes the encoded videos and converts the memory bits into consumable bitstreams. The content distribution network (CDN) enables the distribution component of the pipeline to stream the content to client players.

On the client side, the stream de-packager takes and reads from the consumable bitstreams that feed the decoder. The client player's video decoder then decompresses the video bitstream. Finally, the video renderer takes decoded video sequences and renders them based on user preference, into what is seen on the device.

3. MPEG IMMERSIVE MEDIA STANDARDS

MPEG is developing immersive codecs to compress and stream volumetric content. The emerging MPEG immersive media standards support point clouds and related data types, as well as immersive video formats. A brief description of the immersive codecs is given here to establish the technical background for the object-based implementation described in a later section.

3.1 Video-based point-cloud coding (V-PCC)

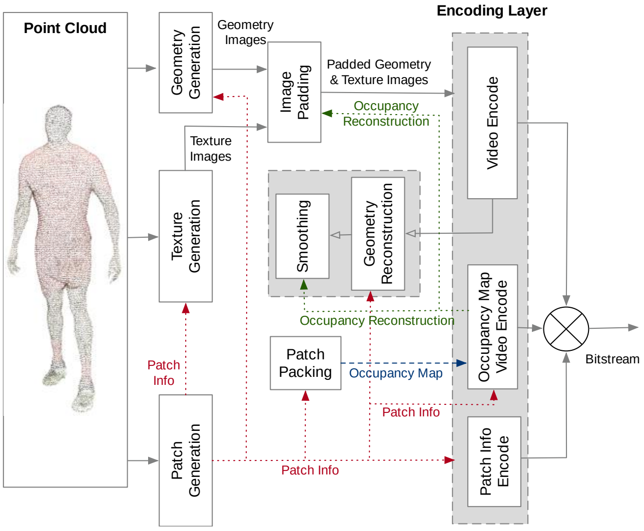

Fig. 4 – V-PCC encoding process

The V-PCC [1, 2] encoder takes dynamic point-cloud data as an input (per frame), projects it onto orthogonal planes (forming a cube) resulting into texture (i.e., RGB components) and geometry (i.e., depth between projection plane and point cloud) images. Then it performs segmentation to extract rectangular regions known as patches (of nearly similar depths) from each projection plane.

These patches are packed into a tiled canvas (also known as atlases in immersive video) along with the occupancy map that indicates parts of the canvas that shall be used (i.e., occupied regions within the patches packed in the canvas).

Also, patch information metadata is generated to indicate how patches are mapped between the projection planes and the canvas. Afterwards, existing video encoders such as AVC and HEVC are used to exploit the spatial and temporal redundancy of geometry and texture components of the canvas. The video encoded streams are combined with the video encoded occupancy maps and the encoded patch information metadata into a single bitstream. The encoding process in V-PCC compression is summarized in Fig. 4.

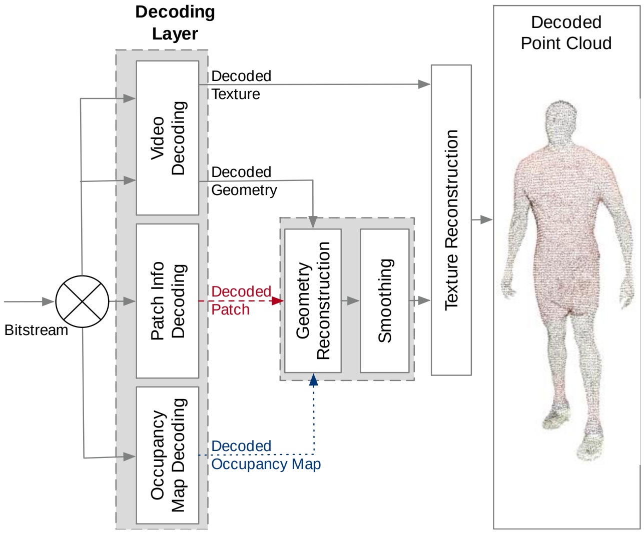

Fig. 5 – V-PCC decoding process

At the decoding stage, the V-PCC bitstream is demultiplexed and the components (texture, geometry, occupancy maps, patch information) bitstreams are decoded reversing the applied coding operation.

Then a geometric reconstruction is carried out from the decoded geometry content using the occupancy maps and patch information followed by a smoothing procedure to clean the reconstruction. The smoothed reconstructed geometry information is then used to map the texture video to generate the decoded dynamic point cloud. The V-PCC decoding process is illustrated in Fig. 5.

3.2 MPEG immersive video (MIV)

The MIV standard [3] enables the viewer to dynamically move with 6DoF, adjusting position (x, y, z) and orientation (yaw, pitch, roll) within a limited range, i.e., as supported by a head mounted display or 2-D monitor with positional inputs as examples.

The MIV [4] encoder takes texture and depth videos from multiple source views, each at a particular position and orientation, as an input, optimizes them by identifying a few as basic views, and prunes the non-basic ones by projecting them one by one against the basic views (and the previously pruned views) to extract the non-redundant occluded regions.

The aggregator then accumulates the pruning results over an intra-period (i.e., preset collection of frames) to account for motion (which helps in efficiently encoding the content). By the end of the intra-period, a clustering is applied to extract the rectangular patches, which in turn are packed into atlases (composed of texture and depth components) with content updated per frame across the processed intra-period.

The occupancy maps (indicating the valid regions within the patches packed in atlases) are embedded within the lower range of the depth component of the atlases during the depth occupancy coding stage rather than signaling them separately as in the V-PCC case. The atlases are finally encoded using the existing HEVC video codec.

The associated camera parameters list (illustrating how views are placed and oriented in space) and atlas parameters list (indicating how patches are mapped between the atlases and the views) are carried as metadata within the bitstream.

The encoding process of MIV compression is summarized in Fig. 6.

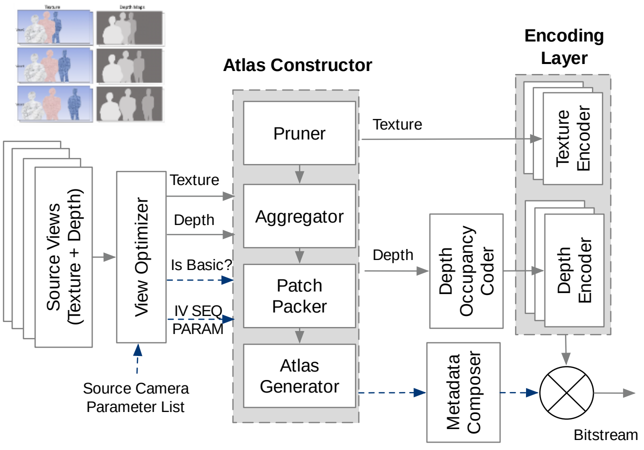

Fig. 6 – MIV encoding process

At the decoding stage, video decoding is applied to retrieve the atlases, the metadata is parsed, and the block to patch maps (also known as patch ID maps and of the same size as the atlases indicating the patch ID the associated pixel within the atlas belongs to which helps resolve overlapped patches) are generated. The MIV decoder does not specify the reference renderer but supplies it with the required metadata and decoded streams.

The intended output of the reference renderer is a perspective viewport of the texture, selected based upon a viewer's position and orientation, generated using the outputs of the immersive media decoder. The MIV decoding process is illustrated in Fig. 7.

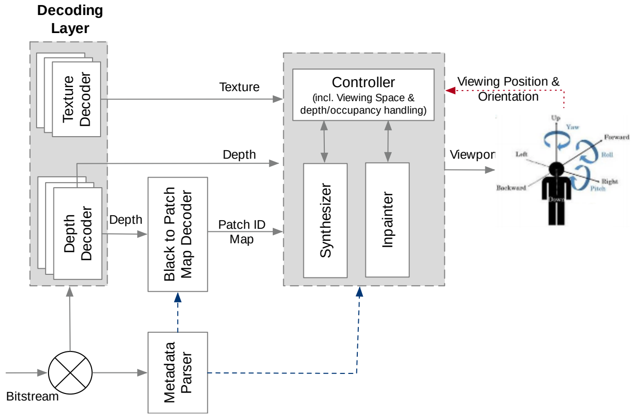

Fig. 7 – MIV decoding process

4. DELIVERING OBJECT-BASED IMMERSIVE MEDIA EXPERIENCE

The implementation of object indexing input provides a solution for delivering object-based features for immersive media experiences.

4.1 Objects indexing input

The object based coding solution requires the ability to relate points and pixels in the scene to their objects. For point-cloud representation [5], we annotate each input point with an object ID, as part of point-cloud object attributes, shown in Fig. 8. The object ID is set to uniquely identify per point-cloud object in a scene within a finite time period.

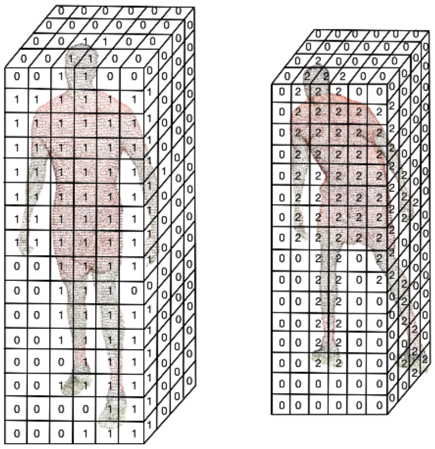

Fig. 8 – Object IDs annotation for point-cloud objects

For immersive multi-view videos [6], pixels from different views that belong to the same object are assigned the exact object ID in a form of maps.

Object maps are of the same resolution as the texture and depth maps but their bit depth depends on the number of objects that require indexing in the scene. Fig. 9 shows the components of immersive content made available at the MIV encoder input.

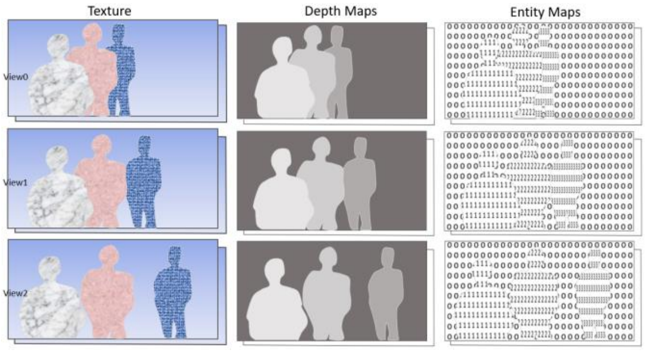

Fig. 9 – Immersive data composed of texture views, depth maps, and object maps (showing 3 views for simplicity)

Object IDs can be generated by using machine learning or a conventional classifier, or a segmentation algorithm running across all points in the point cloud or across all views in the immersive content to identify different objects and assign the exact object ID to various points belonging to the same object.

Alternatively, objects can be captured separately and then populated in the same scene making it simple to tag the points or pixels of each object with the related object ID.

4.2 Implementation in immersive standards

With object maps and object attributes being available at the input, the object based encoder aims to extract patches where each includes content from a single object. Thus the patches can be tagged by the associated object ID whether added as part of the patch metadata or sent within a supplemental enhanced information (SEI) message.

In the V-PCC case [5], the point cloud is segmented and projected (with all its attributes including the object ID) onto the surrounding cube faces forming geometry and texture views along with the object maps. For the MIV case [6], the view optimizer labels the source views (and possibly novel views) as basic or not, and the object maps are carried through.

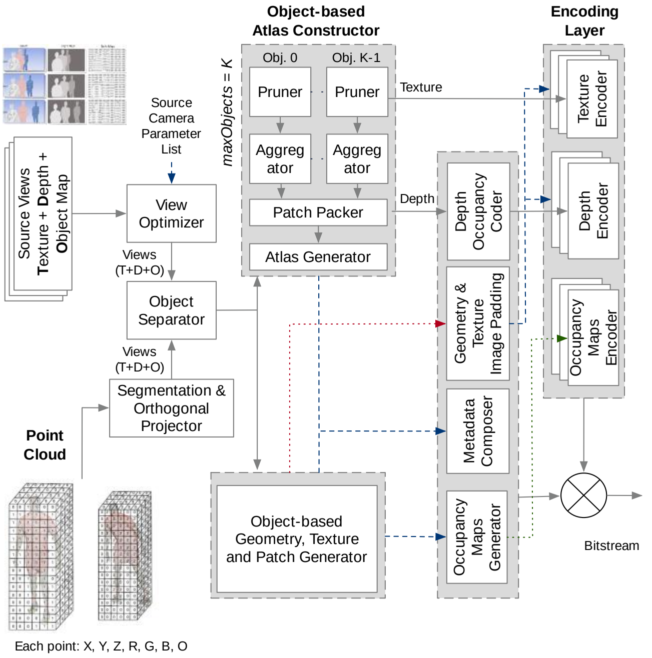

Fig. 10 – Object-based MIV & V-PCC encoding process

A summary of the object-based encoding process for MIV and V-PCC is illustrated in Fig. 10.

Object separators are used to turn the views (texture and geometry/depth) into multiple layers based on the associated object maps where each layer only has regions belonging to a single object. Then the geometry-texture-patch generation in V-PCC (explained in section 3.1) and pruner-aggregator-clustering in MIV (explained in section 3.2) are applied on the layers belonging to one object at a time.

This results in patches where each patch has content from a single object; although, they may be packed together in the atlases/canvases and encoded as previously done. Note that in case of limited bandwidth or a need to highlight certain regions of action, the encoder may choose to drop (or blur) patches of non-related objects from the atlases or dedicate higher resolution for patches of objects of interest. This is only made feasible by adopting the object-based immersive coding solution.

The decoding process of MIV and V-PCC remains the same. The only difference is that the renderer now can make use of the object ID per patch to render only the objects of interest or replace others by a synthetic content enabling innovative use cases.

In addition, a 3D bounding box (may include an object label) per object can be signaled in a supplemental enhanced information (SEI) message [7]. This allows efficient identification, localization, labeling, tracking, and object-based processing.

4.3 Object-based immersive media platform

Fig. 11 illustrates additional modules required for the object-based MIV and V-PCC coding features on our immersive media platform. On the encoding (i.e., server) side, both depth information and object segmentation information are computed from the point-cloud sequence.

An MIV encoder combines the multiple virtual cameras and the depth and object information to form coded bitstreams for immersive video. Similarly, the point cloud with points' attributes (texture, geometry, object ID) are passed to the object-based V-PCC encoder for processing. An optional video encoder can also be used to encode a few virtual cameras (could be 360 videos) in separate channels to support backward compatibility in case consumers' devices do not support V-PCC or MIV decoders.

The stream packager combines the encoded bitstreams together and adds further metadata information to indicate various assets in the scene. Then the output multiplexed bitstream is handled by the content distribution network.

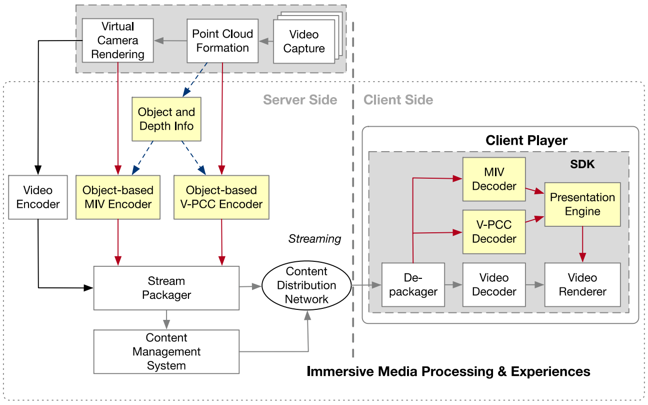

Fig. 11 – Immersive media platform supporting object-based V-PCC and MIV coders

At the client side, the process is reversed and the bitstream is demultiplexed by the depackager so substreams can be handled by the relevant decoders (regular video decoder, MIV decoder, V-PCC decoder). Then the rendering engine makes use of all the decoded representations to deliver the desired viewport/volumetric content.

Supporting various representations (e.g., 360 virtual video, immersive video, point clouds) of the same objects in the streaming solution gives the renderer better capabilities to reconstruct the scene elements adaptively with user motion. For instance, far objects can be rendered from the virtual 360 video since they do not react to the viewer's motion.

Middle-distant objects can be rendered from the immersive video content, since they can support the motion parallax within a certain viewing space defined by the capturing system. Close objects may be rendered using the point clouds in case the viewer is circulating around to experience them from all sides. The renderer can also combine various representations of the same object to produce a more complete and higher quality rendering results. In addition, augmenting new objects of either representation can also be handled by the immersive media platform to enrich the visual immersive experience.

Furthermore, since MIV and V-PCC substreams are essentially the same (e.g., geometry and attribute components of the streamed atlases/canvas plus common metadata parts), it is possible to take an MIV substream and process it by V-PCC decoder to reconstruct it in 3D (although parts of it may be incomplete depending on the camera arrangment capturing the MIV content) or take a V-PCC substream and process it by MIV decoder to render a viewport interactively with the viewer's position and orientation.

5. APPLICATIONS AND SERVICES

Introducing object ID per patch enables novel use cases for volumetric video encoding, decoding, and rendering. It also helps in meeting the MPEG requirements for immersive media access and delivery [8].

5.1 Priority objects rendering

With object IDs available at the decoder side, the renderer can select which objects to output first (e.g., it may start rendering the front objects first) while background/static objects (can be inferred from objects' labels within the associated SEI message) can be carried from the last rendered intra-frame (to save compute and bandwidth). This helps to speed up processing at the decoding side to meet real-time requirements.

5.2 Objects filtering

The user may choose to render only the objects of interest while blur or remove other objects. Also, the encoder may choose to pack patches only from those objects of interest and transmit them rather than sending them all in the event of limited bandwidth, or objects may dedicate more bits to these patches to deliver objects of interest in higher resolution.

5.3 Background rendering

Background is a special static object that can be rendered by itself from the related patches or synthesized from virtual/pre-rendered content. In the case of rendering from patches, there might be regions in the background that are not visible in any of the input source views due to occlusions. Such hole regions can be filled using inpainting techniques. Another approach is to capture the scene ahead of time without any objects and stream a single image metadata once per intra-period so it can be used for rendering the background content and populate the scene with objects of interest afterwards. A synthetic background can be inserted as well, and objects can be augmented within.

5.4 Object-based scalability

Point-cloud objects and a separate background provide object-based scalability for adaptive streaming over different network conditions. Patches belonging to unimportant point-cloud objects, e.g., point-cloud objects too far away from a viewport, can be entirely dropped or encoded at lower visual quality. It is the job for an encoder to decide the relative importance of point-cloud objects using contextual information available to it.

5.5 Personalized 6DoF user experience

Viewers of object-based volumetric video can filter out uninteresting or unimportant objects and keep only relevant or interesting objects based on the bounding box attributes of point-cloud objects, even though all data / objects have been streamed to the client side. The decoder simply does not render patches if a viewer filters out the object these patches belong to. This allows a personalized experience for viewers to choose only content that matters to them, e.g., show me only the red team players or offense players.

5.6 Object of interest

The same personalization idea can be extended to content creation: each object can be compressed in different visual quality. If one wants to focus on a specific point-cloud object, one can latch on to the stream with the object-of-interest encoded in higher visual quality.

6. CONCLUSION

At the time of this paper's publication, the immersive media standardization effort is still ongoing. The context of the paper presents an object-based point-cloud signaling solution that provides a simple way to meet multiple MPEG-I requirements for emerging immersive media standards V-PCC and MIV. This approach supports the future interoperability needs to have a standard-based signaling mechanism for uniquely identifiable objects in sports.

The requirements contributions for MPEG-I include the signaling and object ID for each patch in V-PCC and adding signaling per patch of an object ID as part of MIV. These contributions were adopted by MPEG, and modified to be included in an SEI message instead of in the patch data syntax structure. The object-based applications proposal to MPEG was adopted as part of MIV, which adds the signaling ability per patch of an object ID.

These contributions address the needs of Intel Sports to optimize point-cloud compression and object identification for creating immersive media experiences in sports. The object-based approach also provides a very low impact solution for when the feature is not utilized. In addition, the object-based point-cloud signaling can be used to create innovative visual experiences outside of sports by providing a simple solution for identifying points of interest that ultimately create higher quality volumetric content.

To conclude, MPEG-I supports delivering object-based immersive media experiences for sports, and the approach can be applied towards other use cases.

REFERENCES

[1] "Draft DIS of ISO/IEC 23090-5 Video-based Point Cloud Compression", 11 October 2019, N18888, Geneva, CH.

[2] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivoku´ca, 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 Emmerging and Selected Topics in Circuits and Systems, Vol. 9, No. 1, Mar. 2019.

[3] J. Boyce, R. Doré, V. Kumar Malamal Vadakital (Eds.), "Working Draft 4 of Immersive Media (Video)", ISO/IEC JTC1/SC29/WG11 MPEG/N19001, Jan. 2020, Brussels, Belgium.

[4] B. Salahieh, B. Kroon, J. Jung, M. Domański, "Test Model 4 for Immersive Video", ISO/IEC JTC1/SC29/WG11 MPEG/N19002, Jan. 2020, Brussels, Belgium.

[5] B. Salahieh, J. Boyce, F. Yeung, "Object-Based Applications for Video Point Cloud Compression", ISO/IEC JTC1/SC29/WG11 MPEG/m50950, Oct. 2019, Geneve, CH.

[6] B. Salahieh, J. Boyce, F. Yeung, "Object-Based Applications for Immersive Video Coding", ISO/IEC JTC1/SC29/WG11 MPEG/m50949, Oct. 2019, Geneve, CH.

[7] J. Boyce, B. Salahieh, F. Yeung, "SEI Messages for MIV and V-PCC", ISO/IEC JTC1/SC29/WG11 MPEG/m49957, Oct. 2019, Geneve, CH.

[8] "Requirements for Immersive Media Access and Delivery", July 2019, N18654, Gothenburg, SE.