XR Carousel: A Visualization Tool For Volumetric Video

Sylvie Dijkstra-Soudarissanane
TNO
The Hague, The Netherlands
sylvie.dijkstra@tno.nl

Simon Gunkel
TNO
The Hague, The Netherlands
simon.gunkel@tno.nl

Alexandre Gabriel
TNO
The Hague, The Netherlands
alexandre.gabriel@tno.nl

Leonor Fermoselle
TNO
The Hague, The Netherlands
leonor.fermoselle@tno.nl

Frank ter Haar
TNO
The Hague, The Netherlands
frank.terhaar@tno.nl

Omar A. Niamut
TNO
The Hague, The Netherlands
omar.niamut@tno.nl

Figure 1: Carousel demonstrator allowing for subjective testing by showing the differences between the selected encoding parameters

ABSTRACT
Recent years have seen a new uptake in immersive media and eXtended Reality (XR). And due to a global pandemic, computer-mediated communication over video conferencing tools became a new normal of everyday remote collaboration and virtual meetings. Social XR leverages XR technologies for remote communication and collaboration. But in order for XR to facilitate a high level of (social) presence and thus high-quality mediated social contact between users, we need high-quality 3D representation of users. One approach to providing detailed 3D user representations as new immersive media is to use point clouds or meshes, but these representation formats come with complexity on compression bitrate and processing time. In the example of virtual meetings, compression has to fulfill stringent requirements such as low latency and high quality. As the compression techniques for 3D immersive media steadily advance, it is important to be able to easily compare different compression techniques on their technical and visual merits in an easy way. The proposed demonstrator in this paper is a visualization tool that helps assessing the visual quality of a 3D representation employing various coding schemes. The complete end-to-end rendering/encoding chain can be easily assessed, allowing for subjective testing by showing the differences between the selected encoding parameters. The tool presented in this demo paper offers an improved and easy visual process for the comparison of encoders of immersive media.

CCS CONCEPTS
• Human-centered computing → User interface design; Social content sharing; Information visualization.

KEYWORDS
Compression, depth coding, visualization, point clouds, 3D representation

ACM Reference Format:
Sylvie Dijkstra-Soudarissanane, Simon Gunkel, Alexandre Gabriel, Leonor Fermoselle, Frank ter Haar, and Omar A. Niamut. 2021. XR Carousel: A Visualization Tool For Volumetric Video. In 12th ACM Multimedia Systems Conference (MMSys ’21) (MMSys 21), September 28-October 1, 2021, Istanbul, Turkey. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3458305.3478436

1 INTRODUCTION
Transporting one’s self to any place on Earth is a compelling idea; providing expertise or skills at a distance can be useful for industry (co-working, inspection, maintenance), enabling remote education and training, and supporting the inclusion of citizens with accessibility barriers. By using virtual meetings, it is possible to
reduce commutes, lowering our ecological footprint, and even alleviate physical distancing measures caused by a pandemic such as COVID-19. There is a strong need to make communication and remote collaboration as transparent as possible, where the interface should appear to be imperceptible and almost nonexistent to the user. Providing a shared and collaborative 6 degrees-of-freedom experience by using photorealistic and volumetric human representations in a format that can be easily captured, compressed, and transported to current and upcoming VR devices is an important step in making this a reality. 3D point clouds offer a natural representation of a scene as volumetric media, however, due to the complexity of the data and its significant size, the direct usage of 3D data becomes difficult in a VR communication system that needs to comply to stringent requirements such as high throughput, low latency, reliable communication, and high visual quality.

Photorealistic representation of users can prove to be quite challenging, both at the capture level and the processing of the data [10]. For most use cases, a single camera capture and data stream is not sufficient. Assuming 2 users are facing each other in a collaborative XR experience, high level of detail and full coverage of users are achieved with a parallax of a multicamera setting and at a minimum representation of \(180^\circ\), to render the body parts facing each other: at least 2 cameras placed around the user are required [5]. The complexity can increase further if full body reconstruction is needed, which in turn requires at least a 4 cameras setup [3]. While processing and combining 3D captures from RGBD sensors such as the Azure Kinect, the resulting raw format often contains artefacts, as highlighted in Figure 2 with the red boxes.

To compress such user representations, recent initiatives focus on new immersive media representation formats. Some prominent examples are video-based point cloud coding (VPCC) [12], 3D meshes [4] and colour-plus-depth (RGBD) [8]. Thus with multiple sets of encoders in development, the evaluation and comparison of different strategies and solutions is crucial. Furthermore, any evaluation should focus both on objective or technical aspects as well as on subjective (i.e. quality of experience and visual perception) testing. Subjective experiments often prove to be challenging and require extensive efforts to be reliable and reproducible. Thus, new tools are needed to support subjective experiment facilitators and simplify the technical process.

With this demonstrator as depicted in Figure 1, we present a simple tool and processes to simplify the testing of 3D volumetric encoding strategies and allow direct comparison of different settings of one encoder, in-between encoders, and between the raw captured volumetric data. This demonstrator brings a focus on 3D photorealistic human representations and XR communication scenarios.

2 COMPRESSION TECHNIQUES

Most XR communication scenarios incorporate volumetric capture, where persons are captured by multiple cameras. The camera output is typically merged together to create a unified 3D representation of the persons, which is then processed further to enable delivery over a network. We use a modern RGBD sensor, the Azure Kinect, as an example.

![Figure 2: Example of capture artefacts (in red boxes), such as the hole near the ear, blurry hands or the arms rendered with coloring problems that often result from joining the dual cameras captures in one common point cloud.](image)

The Azure Kinect camera captures an 8 bit 720p RGB image as well as a 16 bit depth 720p image, which at 15fps results in a raw throughput of:

\[
8 \times 3 \times (1280 \times 720) + 16 \times (1280 \times 720) \times 15 \text{fps} = 552,960,000 \text{Mbps}
\]

It becomes clear that encoding plays an essential role in making remote immersive experiences possible [13], as at this bitrate the vast majority of internet connections would not be able to handle the content coming from even one camera.

In this paper, an evaluation tool is presented, where different codecs are evaluated. To this end, three modern volumetric video codecs will be evaluated with respect to encoding latency and visual quality performance indicators, as well as giving insight to the resulting visual artefacts. In this demo paper, bitrates will not be directly compared between the different codecs because each of the encoders is different in its implementation, maturity and configuration parameters and it is therefore very difficult to match bitrates with all encoders. Since each codec has different raw input formats, our main focus was placed in preparing a tool that allows the evaluation of subjective differences between the codecs as opposed to objective metrics such as PSNR.

In setting up this demo, as shown in Figure 3, a video (VP9 [7]), point cloud (VPCC [1]) and mesh (Draco [6]) encoder were selected for testing. The video encoder was chosen since the original content is captured as video by the Azure Kinect, while point cloud and meshes are more in line with how the content is eventually rendered at the user side. The captured raw input format was converted to a point cloud and a mesh so that it can be encoded by the 2 non-video encoders.

It should be noted that despite the raw capture being a video, the input for the video encoder still requires conversion. That is, the depth has been converted from 16 bit to 12 bit to be able to use VP9’s 12 bit encoder, as there is currently no open source 16 bit VP9 encoder, and the RGB data is converted to YUV420 format. To form a 3D rendering with the left and right camera, the camera alignment data is used and the depth and RGB values are projected into a 3D space. Both the point cloud and mesh use the raw video data and this simple conversion method to obtain the raw input that is then fed into the VPCC and Draco encoder. For visualizing
the content Unity was used. The evaluation tool itself is further explained in the following section.

3 CAROUSEL DEMONSTRATOR

The main objective of this demonstrator, named “XR Carousel”, is to allow subjective testing of different encoding parameters and codecs. The demonstrator is built in the Unity game engine and allows for cycling through multiple decodings. It is possible to cycle through the different outputs and compare the visual quality at different bitrates with the arrow keys and between different codecs with the spacebar. The test content consists of 400 frames at 15fps and loops with a human subject sitting down on a chair, going through a range of motions and moving their arms - all encodings have encoded this dataset.

The output has been placed in a photorealistic virtual environment (The Great Drawing Room [9]). The user can experience the demo both on the screen of the computer as well as through a VR HMD and has complete freedom explore and view the scene from multiple angles.

Independently of the decoded data type, the XR Carousel demonstrator always represents the data in a colored 3D point cloud format, through a particle system game object. The reason for this choice lies in the fact that this data representation is sufficiently general to represent all different types of decoded output data, as point clouds can also be obtained from RGB and depth images or meshes, through the extrinsic and intrinsic parameters from capture and preprocessing steps, or the underlying color-vertex structure, respectively.

The complexity associated to the point cloud computation from the decoded data (conversion for rendering in Figure 3), is similar to the complexity required to convert the raw captured video from the Azure Kinect sensors into the multiple formats (conversion to data structure in Figure 3).

This demonstrator creates a simple platform for subjective quality comparison of different encoders and VR experience assessment. A video showcasing the demo can be found here: https://tnomedialab.github.io/go/mmsys21-carrousel/.

4 RESULTS

All encodings currently implemented in the demonstrator were run on the same server with an 8 core 2.1GHz Intel Processor, 16GB of RAM and running Ubuntu 20.04.1 LTS. Listed in Tables 1 and 2 are the performance results in terms of the bitrates (Mbps), time (seconds) and fps of each of the encodings. Qualitative assessment on the maturity of the various codecs was also part of this demonstrator development, however, the results of that assessment are not presented here as they are out of the scope of this paper.

Bitrates have not been matched since both VPCC and Draco do not have configuration parameters to do so; VPCC and Draco have been encoded with fixed quality encoding. Figure 4 shows some of the visual artefacts identified for each encoder.

Google’s VP9 is the most mature of the encoder options tested with the fastest encoding time and integration with ffmpeg. When encoding, the depth and RGB values were encoded separately with a fixed bitrate of 2 Mbps for each RGB capture (left and right cameras) since the intent was to visualize depth coding artefacts while video coding of color and its resulting artefacts are quite well known. The bitrates for depth transmission vary between each of the encodings, with roughly 4 Mbps of each encoding reserved for the left and right RGB encodings. There are a total of 4 encodings running in parallel for each quality, 2 RGB and 2 depth. The longest time in seconds has been used as the encoding time. Results are showing higher visual quality at higher depth bitrates, demonstrating that video encoding is a viable solution to encode depth, while at lower bitrates many visible artefacts are visible. At the lowest bitrate, there is evidently more noise between the arms and the body than in the raw capture, and details in the face and forehead also suffer when the depth is encoded at a lower bitrate.

Table 1: Performance results of VP9 in bitrates (Mbps), time (seconds) and fps

| VP9 bitrate | VP9 time | VP9 fps |
|-------------|----------|---------|
| 8,01        | 72       | 5,6     |
| 10,10       | 72       | 5,6     |
| 16,11       | 76       | 5,3     |
| 25,28       | 91       | 4,4     |
| 38,46       | 87       | 4,6     |

Table 2: Performance results of the VPCC-TM2 in bitrates (Mbps), time (seconds) and fps

| VPCC bitrate | VPCC time | VPCC fps |
|--------------|-----------|----------|
| 0,28         | 117673    | 3,4E-03  |
| 0,44         | 127085    | 3,1E-03  |
| 0,79         | 133661    | 3,0E-03  |
| 1,59         | 143301    | 2,8E-03  |
| 3,23         | 159028    | 2,5E-03  |

The VPCC-TMC2 software from MPEG1 has unstable behavior, with frequent crashes occurring in longer encodings; as a workaround, the content was encoded in GOPs of 32 frames at a time, and the separate encoding times and bitrates summed up. The VPCC software itself is far from having real-time capabilities, although the use of the experimental HEVC code from MPEG–HM16 with the 3D extensions as the auxiliary video encoder (required by VPCC) may help explain the long encoding times. While it is possible to couple with other video encoders, HM16 was deemed sufficient for the purposes of this demonstrator. The implementation does not seem to be geared towards a live encoding scenario as of now. We conclude this from the configuration files available that typically have large GOPs and the overall computational complexity of the encoder. On the other hand, the bitrates were the lowest of all tests done with substantially less visible artefacts. Some of the noticeable artefacts at lower bitrates include overall blurriness around the torso and blurring of facial details.

Draco has been released by Google3 and compresses each frame independently. The code is targeted towards online use, where only

---

1https://github.com/MPEGGroup/mpeg-pcc-tmc2
2https://vcgit.hhi.fraunhofer.de/jct-vc/HM
3https://github.com/google/draco
We have developed a tool enabling the visualization of volumetric content for facilitating the comparison between different encoding settings. Example content was included, captured by two RGBD sensors (Azure Kinect) and encoded in 3 different formats: video (VP9), point cloud (VPCC) and mesh (Draco) encoder.

From the 3 codecs that we tested, video-based encoding was the fastest and the only encoder of the set that we think can currently meet the real-time requirements needed for remote communication, although some configuration parameters may need to be adjusted. Point clouds encoded with VPCC resulted in data being transmitted at substantially lower bitrates and have comparable or even better subjective visual quality when viewed in the carousel visualization tool. The Draco mesh encoder resulted in data requiring high bitrate and high encoding time: it became clear that Draco mesh encoding is not a viable option for real-time communication. Despite the high bitrates, there are still visible artefacts around the area with more details such as hair or elements on the T-shirt.

## 5 CONCLUSIONS

We have developed a tool enabling the visualization of volumetric content for facilitating the comparison between different encoding settings. Example content was included, captured by two RGBD sensors (Azure Kinect) and encoded in 3 different formats: video (VP9), point cloud (VPCC) and mesh (Draco) encoder. From the 3 codecs that we tested, video-based encoding was the fastest and the only encoder of the set that we think can currently meet the real-time requirements needed for remote communication, although some configuration parameters may need to be adjusted. Point clouds encoded with VPCC resulted in data being transmitted at substantially lower bitrates and have comparable or even better visual quality when viewed in the carousel visualization tool. The Draco mesh encoder resulted in data requiring high bitrate and high encoding time: it became clear that Draco mesh encoding is not a tool designed for virtual meeting applications. Given that we see more focus on real-time codecs for 3D volumetric representations, e.g. geometry-based point cloud codecs in [2, 11], we expect to broaden our set of codecs under evaluation now that the evaluation tool has been developed.

This demonstration setup proves to be useful for assessing subjective visual quality when dealing with varying input/output from currently available encoders. Thanks to our tool, more in-depth subjective testing can be easily achieved.

## 6 ACKNOWLEDGMENTS

The work in this paper is funded by the TNO Early Research Programme ‘Social eXtended Reality’.

## REFERENCES

[1] MPEG 3DG. 2021. MPEG VPCC TMC2 specifications Overview. https://github.com/MPEGGroup/mpeg-pcc-tmc2 Accessed: 2021-05-31.

[2] Chao Cao, Marius Preda, and Pablo Cesar. 2017. Design, Implementation, and Evaluation of a Point Cloud Codec for Tele-Immersive Video. IEEE Transactions on Multimedia 58, 10 (2020), 49–55. https://doi.org/10.1109/MMCOM.001.2000242

[3] Anargyros Chatzitofis, Leonidas Saroglou, Prodromos Boutis, Petros Drakoulis, Nikolaos Zoulis, Shishir Subramaniam, Bart Kevelham, Caecilia Charbonnier, Pablo Cesar, Dimitrios Zarpalas, et al. 2020. HUMAN4D: A Human-Centric Multimodal Dataset for Motions and Immersive Media. IEEE Access 8 (2020), 176241–176262.

[4] Kyriaki Christaki, Emmanouil Christakis, Petros Drakoulis, Alexandros Doumanoglou, Nikolaos Zoulis, Dimitrios Zarpalas, and Petros Daras. 2019. Subjective visual quality assessment of immersive 3D media compressed by open-source static 3D mesh codecs. In International Conference on Multimedia Modeling. Springer, 80–91.

[5] Sylvie Dijkstra-Soudarissanane, Karim El Assal, Simon Gunke, Frank ter Haar, Rick Hindrikis, Jan Willem Kleinrouweiler, and Omar Niamut. 2019. Multi-sensor capture and network processing for virtual reality conferencing. In Proceedings of the 10th ACM Multimedia Systems Conference. 316–319.

[6] Google Draco. 2021. Draco specifications Overview. https://google.github.io/draco/ Accessed: 2021-05-31.

[7] Google. 2021. VP9 Overview. https://developers.google.com/media/vp9 Accessed: 2021-05-31.

[8] Simon Gunke, Rick Hindrikis, Karim El Assal, Hans Stokking, Sylvie Dijkstra-Soudarissanane, Frank ter Haar, and Omar Niamut. 2021. VRComm: An end-to-end web system for real-time photorealistic socialVR communication. In Proceedings of the 12th ACM Multimedia Systems Conference.

[9] Sketchfab Halfway Museum. 2021. The Great Drawing Room. https://sketchfab.com/3d-models/the-great-drawing-room-feb9ad17e60424188ce5759b88e3b2e5d? License: https://creativecommons.org/licenses/by/4.0/legalcode Accessed: 2021-05-31.

[10] Jeroen van der Hooff, Maria Torres Vega, Tim Wauters, Christian Timmerer, Ali C. Begen, Filip De Turck, and Raimund Schatz. 2020. From Capturing to Rendering: Volumetric Media Delivery with Six Degrees of Freedom. IEEE Communications Magazine 58, 10 (2020), 49–55. https://doi.org/10.1109/MCOM.001.2000242

[11] Rafaël Melkuria, Kees Blom, and Pablo Cesar. 2017. Design, Implementation, and Evaluation of a Point Cloud Codec for Tele-Immersive Video. IEEE Transactions on Multimedia 58, 10 (2020), 49–55. https://doi.org/10.1109/MMCOM.001.2000242

Table 3: Performance results of Draco in bitrates (Mbps), time (seconds) and fps

| Draco bitrate | Draco time | Draco fps |
|---------------|------------|----------|
| 49.05         | 811        | 0.49     |
| 69.43         | 813        | 0.49     |
| 89.62         | 842        | 0.48     |
| 109.00        | 916        | 0.44     |
| 128.18        | 971        | 0.41     |

The work in this paper is funded by the TNO Early Research Programme ‘Social eXtended Reality’.
(a) VP9 video encoding. Visual artefacts are visible around the arms at low bitrates as well as the facial area even at higher bitrates can sometimes be hard to see.

(b) VPCC encoding. In general quality is quite high but the facial area can sometimes be hard to see.

(c) Draco encoding. Highly detailed areas such as the t-shirt logo can be a bit blurry.

Figure 4: Visual artefacts testing 3 different codecs