PC-MCU: Point Cloud Multipoint Control Unit for Multi-user Holoconferencing Systems

Gianluca Cernigliaro, Marc Martos
Mario Montagud, Amir Ansari, Sergi Fernandez
{gianluca.cernigliaro,marc.martos,mario.montagud,amir.ansari,sergi.fernandez}@i2cat.net
i2CAT Foundation
Barcelona, Spain

ABSTRACT

This paper introduces the Point Cloud Multipoint Control Unit (PC-MCU): a key component for multi-user holoconferencing systems, where remote participants are represented as Point Clouds. The presented solution redefines the idea of MCU, broadly used to optimize connections and communications between users in traditional videoconferencing, and introduces a set of key features for the optimization of holoconferencing services where multiple users can be remotely connected. The PC-MCU is a virtualized cloud-based component, that aims at reducing the end-user client computational resources and bandwidth usage, providing the following key features: fusion of volumetric videos, Level of Detail (LoD) adjustment and non visible data removal. The results obtained for a scenario with two remote users, show how the introduction of the PC-MCU provides significant benefits in terms of computational resources and bandwidth savings, thus alleviating the requirements at the client side in holoconferencing services when compared to a baseline condition without using it. These improvements open the door to further research on this area to enable scalable and adaptive holoconferencing services using lightweight devices.

CCS CONCEPTS

• **Computer systems organization → Real-time operating systems**; **Real-time systems**; *Real-time operating systems*; • **Networks → Cloud computing**; **Network services**; *Cloud computing*.

KEYWORDS

VR, AR, XR, Point Clouds, volumetric video, holoportation, holoconferencing, MCU, Social VR

ACM Reference Format:
Gianluca Cernigliaro, Marc Martos and Mario Montagud, Amir Ansari, Sergi Fernandez. 2020. PC-MCU: Point Cloud Multipoint Control Unit for Multi-user Holoconferencing Systems. In *Proceedings of NOSSDAV '20: Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'20).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3386290.3396936

1 INTRODUCTION

Nowadays people are connected remotely on a daily basis, thanks to technologies as videotelephony and videoconferencing. Strong examples are services like Skype and Google Hangouts, commonly used also in a variety of environments, transmitting every day large volumes of data. Considering the needs of such systems in terms of data handling and scalability, and given the limited resources at the client side, Multi-point Control Units (MCUs) [19] rapidly became core components in video-communication systems, managing sessions and communications, and performing additional advanced features, as layout and quality adaptability.

Yet, the continuous evolution of video technologies focuses on other aspects, apart from those related to networking, like for instance 3D video paradigms and their constant improvements in terms of quality, sense of realism and immersion. The research community and industry have then started focusing on the development of novel systems to represent, compress and transmit 3D, or volumetric video, giving a strong importance to natural content. As proof of relevance, this topic has recently attracted the attention of the Moving Pictures Experts Group (MPEG), initiating the very first Point Clouds compression standardization process [17]. This evolution has also pushed the development of other key aspects, as real-time capturing and delivering of volumetric video. All the aforementioned advances, together, have opened the door to the holoportation concept, enabling a real-time rendering of volumetric videos captured from remote locations. One of the very first holoportation systems has been developed by Microsoft, as shown in Figure 1 [11].

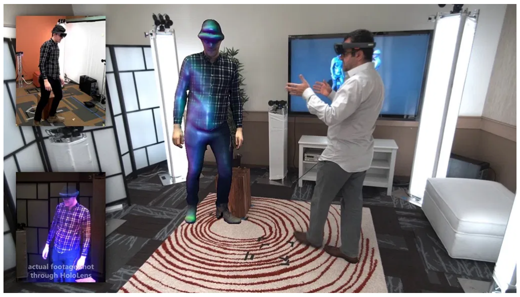


Figure 1: Microsoft Real-time holoportation system [11].

However, the quality of the volumetric video represented in holoconferencing is still not comparable to the one reached by traditional 2D videoconferencing systems. The technology behind holoportation systems is indeed at an early stage and, given the high volumes of data involved in the representation of Point Clouds,

NOSSDAV'20, June 10–11, 2020, Istanbul, Turkey

ACM ISBN 978-1-4503-7945-8/20/06…$15.00
https://doi.org/10.1145/3386290.3396936

real time processing is a demanding task and not suitable to all client devices. These issues have been addressed by the Network Based Media Processing (NBMP) task force within MPEG [14], with the suggestion of moving the overloading parts of the processing of immersive applications content to the cloud.

To face the most challenging issues of a real time 3D holoconferencing system, we present the Point Cloud Multipoint Control Unit (PC-MCU): a solution for multi-user holoportation systems, in charge of handling the volumetric video content, represented as Point Clouds, applicable to any Virtual/Augmented/Mixed Reality (VR/AR/MR), or more generically, eXtended Reality (XR) services. As the MCU for 2D videoconferencing, the PC-MCU is a cloud application able to receive multiple Point Cloud streams and, considering the emerging needs of a multi-user 3D holoconferencing system, performs the next key novel features:

- **Multiple Volumetric Video de/coding**, being compatible to the most common Point Cloud compression strategies.
- **Reception and delivery** of multiple MPEG Dynamic Adaptive Streaming over HTTP (DASH) streams.
- **Level of Detail (LOD) adjustment**: the Point Cloud representation of the participants can be down-sampled, providing the most appropriate resolution based on the relative users' position, activity and underlying context.
- **Removal of non visible volumetric video**: the non visible parts of the volumetric environment can be removed and never received.
- **Fusion of volumetric videos**: the incoming videos are decoded, processed accordingly, and converted into as a fused volumetric video for the scene, thus providing a single stream to the client devices.

This set of features creates specific volumetric video streams, optimized for every user depending on their position and viewpoint, producing a reduction of resources usage, as RAM, CPU, GPU, and of bandwidth.

The PC-MCU and its features have been evaluated in a simple real scenario, with two virtual remote users, and by performing several actions to determine the obtained benefits. In general, the obtained results prove a significant reduction in terms of resources consumption metrics (RAM, CPU, GPU) and bandwidth when introducing the PC-MCU. These promising results open to door to further research on this timely and relevant topic.

The paper is structured as follows: Section 2 breafly reviews the state of the art related to real-time holoconferencing systems, Section 3 presents the designed PC-MCU and its key novel features. The obtained results are presented in Section 4 and, finally, the paper ends with the conclusions and the future research ideas in Section 5.

2 STATE OF THE ART

The strategy of reducing the computational load of a videoconferencing client machine, by moving to the cloud the heaviest operations, has been widely addressed for communications based on traditional 2D video. The MCU concept, already introduced in the '90s [19], has been indeed deeply analyzed and considered in standardization works, like the ITU recommendation H.323, that defines the protocols to provide audio-visual communication sessions [18].

The community has also devoted efforts on adapting the MCU concept to newly developed video distributions systems (e.g. [1], [7]), and converting it into an essential component of virtualization architectures, like the widely known Cisco Unified Computing System (UCS) [5].

The idea of virtualizing part of the processing load for immersive applications, has been suggested by the Global System for Mobile Communications Association (GSMA) [4], providing a high level architecture with shared computational resources in the cloud for VR/AR services, independently from the content to be delivered.

The research work oriented to the optimization of volumetric video transmission systems can exploit the previous work on traditional video, but also the new strategies generically defined for other immersive formats, like 360 degrees videos. A relevant example of this possibility is given by techniques as the viewport-aware adaptive streaming for 360 degrees videos. One example is the solution by Ozcinar *et al.* [12], where the content, is divided into tiles, and higher quality tiles are provided to the region associated to the users' viewport. The adaptation of such strategy has been explored by Park *et al.* [13], who proposed an adaptive streaming strategy for Point Cloud based volumetric video, with the difference that the tiling, previously applied as a 2D segmentation of the 360 degrees, follows now a volumetric approach, dividing the three-dimensional environment into cubic sections. That streaming strategy considers both the bit-rate and user's viewpoint to decide the tiles to deliver and with which quality.

Other peer-to-peer oriented strategies, for the dynamic streaming of Point Clouds, have been proposed by Hosseini and Timmerer [6], including a resolution scalability strategy, giving the chance to select the best trade-off between quality and bandwidth/resolution. Qian et al. [15] proposed a volumetric video streaming system for commodity mobile phones, where they propose heuristics to decide on representations and to use edge computing to reduce the computational load on mobile clients.

The usage of an MCU for videoconferencing in three-dimensional environments has been explored by Dijkstra et al. [2], where MCU features, normally used in 2D video, are applied to users representations transmitted as multi-view plus depth. This way, it is possible to take advantage of the bi-dimensional nature of such content to minimize the load at the client side.

Compared to the existing state-of-the-art solutions in this field, the presented PC-MCU provides the next key advantages and/or novel aspects:

- The MCU concept is, for the first time, applied to fully three-dimensional, volumetric video represented as Point Clouds.
- Actions as resolution selection and viewport aware streaming are now included in a cloud-based managing system for an efficient distribution of volumetric video.
- The availability of a virtualized cloud-based PC-MCU, and its functionalities, allows every user to send a constant, and full resolution, volumetric representation, delegating the optimization to the PC-MCU.

The next section describes the architecture, components and aforementioned features of the proposed PC-MCU, which aims at being a standard-compliant enabler for multi-user volumetric conferencing services.

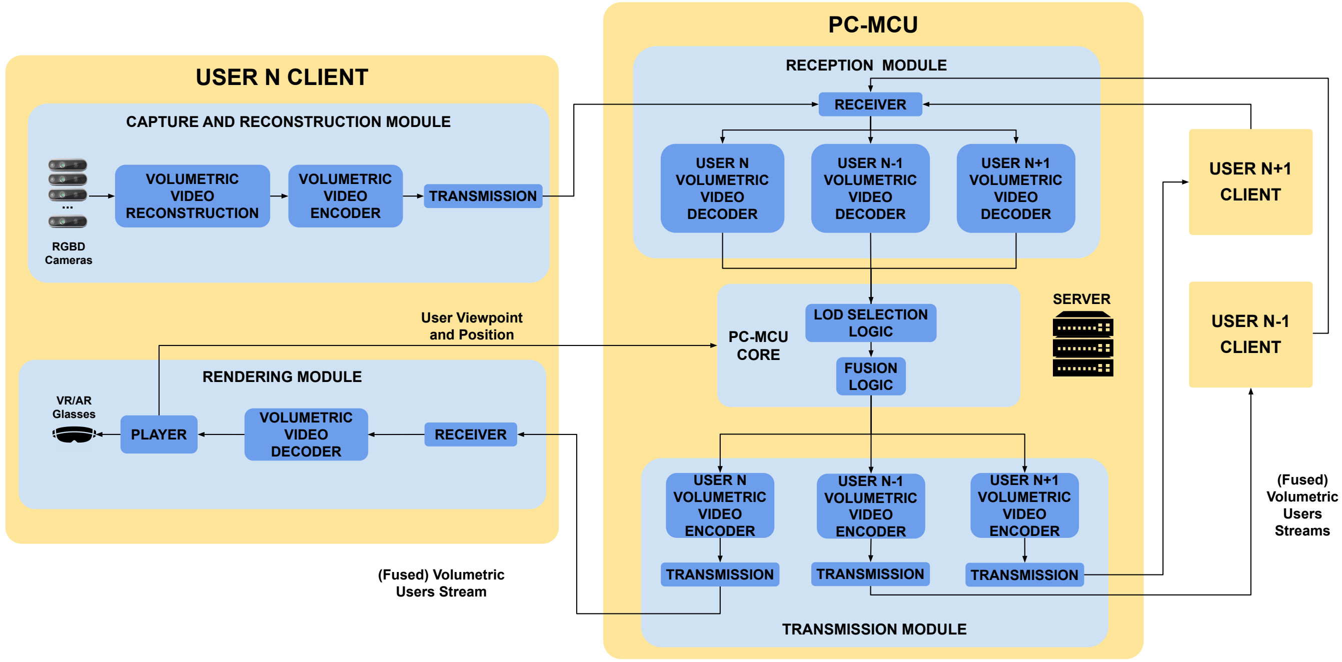

USER N CLIENT
CAPTURE AND RECONSTRUCTION MODULE
RGBD Cameras
VOLUMETRIC VIDEO RECONSTRUCTION
VOLUMETRIC VIDEO ENCODER
TRANSMISSION
User Viewpoint and Position
RENDERING MODULE
VR/AR Glasses
PLAYER
VOLUMETRIC VIDEO DECODER
RECEIVER
(Fused) Volumetric Users Stream
PC-MCU
RECEPTION MODULE
RECEIVER
USER N VOLUMETRIC VIDEO DECODER
USER N-1 VOLUMETRIC VIDEO DECODER
USER N+1 VOLUMETRIC VIDEO DECODER
PC-MCU CORE
LOD SELECTION LOGIC
FUSION LOGIC
SERVER
USER N VOLUMETRIC VIDEO ENCODER
USER N-1 VOLUMETRIC VIDEO ENCODER
USER N+1 VOLUMETRIC VIDEO ENCODER
TRANSMISSION
TRANSMISSION
TRANSMISSION
TRANSMISSION MODULE
USER N+1 CLIENT
USER N-1 CLIENT
(Fused) Volumetric Users Streams

Figure 2: PC-MCU Overall Architecture.

3 POINT CLOUD MULTIPOINT CONTROL UNIT (PC-MCU)

This section presents the design of the proposed solution, focusing on the description of: i) how the PC-MCU is integrated, as a component, in a multi-user system (Section 3.1), specifying inputs and outputs and how it interacts with the other elements of the network; ii) the specific functionality provided and their effects on the holoconferencing experience (Section 3.2).

3.1 Architecture and sub-components

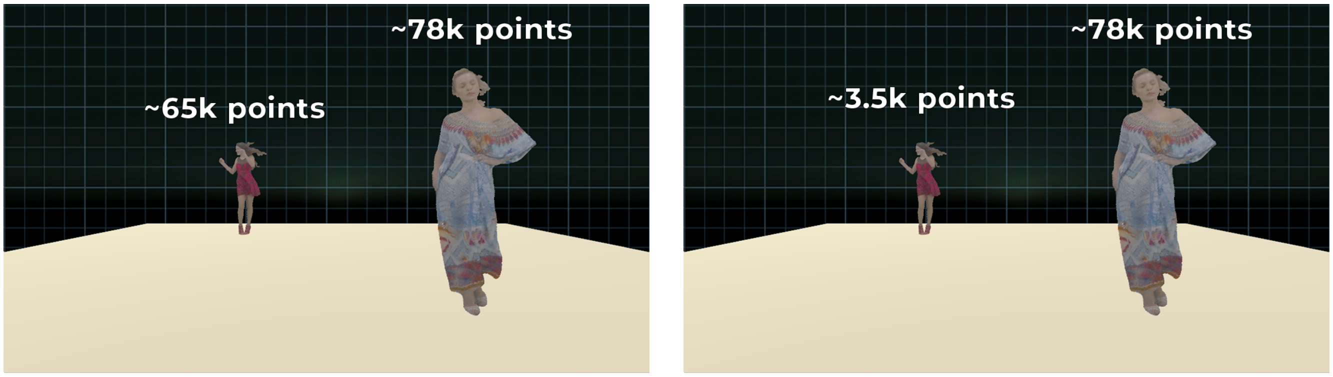
We consider a holoconferencing network where a number of N users are connected to the PC-MCU. A graphical description of the overall architecture of the system is shown in Figure 2. Each user's client includes i) a *Capture and Reconstruction module*, in charge of creating, compressing and transmitting the volumetric video, represented as Point Clouds, captured by a number of RGBD sensors and ii) a *Rendering module* able to receive, decode and represent the volumetric videos of the other users in a VR/AR display. Apart from the volumetric video, the user's client is in charge of transmitting the Viewpoint and Position of the user. The PC-MCU will then handle such information to optimize the content specifically for each user.

The PC-MCU interfaces to the user's clients are the *Reception* and *Transmission* modules. The first one is in charge of receiving and decoding the videos from the users, before providing them to the *Core Sub-system*. The second one compresses and transmits the specific content to each user. The *Core Sub-system* includes the main functionalities available in the PC-MCU.

3.2 PC-MCU Functionalities

The PC-MCU *Core Sub-system* implements key features that allow a significant reduction of the client computational load and bandwidth consumption. These features are explained in the following sub-sections.

3.2.1 Volumetric Video De/Coding. The *Reception* and *Transmission* modules represent the interfaces between the PC-MCU and the network, for the reception and transmission of volumetric videos from/to each user client. The *Reception Module* receives one Point Cloud per user, and delivers the incoming streams to a number of decoder instances equal to the number of participants. The uncompressed data from the decoders are provided to the *Core Sub-system*, which generates a data stream optimized for each client, by applying the functionalities described next. After the optimization, the *Core Sub-system* provides the streams to the *Transmission Module*, with a dedicated encoding instance for each client, directly connected to its corresponding transmission component via MPEG DASH. The PC-MCU has been conceived with the goal of being compatible with the most popular volumetric video compression methods that can perform a real time, low latency, encoding and decoding of volumetric content. The version presented in this paper uses the MPEG anchor implementation developed by Mekuria et al. [9], with only Intra frames, due to its real-time oriented nature. In future releases of the PC-MCU, the outcomes from ongoing standardization contributions in this context [17] will be considered. However, note that the aforementioned details are provided for completeness as the specific encoding and transmission strategies for the DASH stream are out-of-scope of this paper.



(a) No PC-MCU: similar resolution Point Clouds (b) With PC-MCU: Further (left) Point Cloud downsampled

Figure 3: Example of LoD selection operated by the PC-MCU.

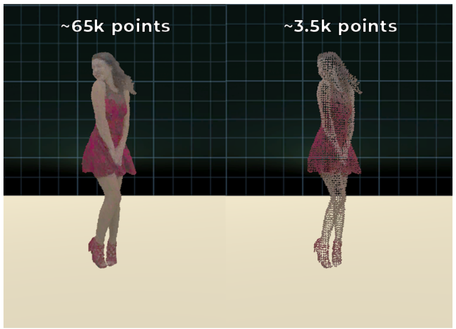


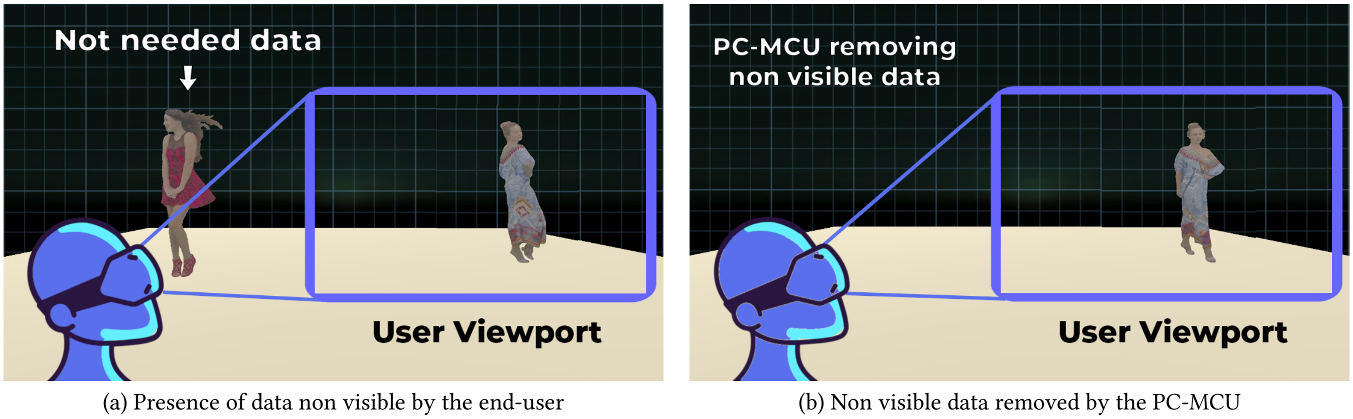
Figure 4: Comparison of the two resolutions applied to the Point Cloud affected by the LOD Selection.

3.2.2 Level of Detail (LoD) Selection. Traditional MCUs for videoconferencing are capable of controlling the resolution of the incoming videos, adapting it according to the bandwidth capacity and to the client device characteristics. In a volumetric video, the resolution can be adjusted by tuning a parameter called Level of Detail (LoD) [8], which controls the geometrical characteristics of the video, usually represented as vertexes of polygons or simple three-dimensional points, in x, y, z coordinates. The adjustment and selection of appropriate LoDs, is then a powerful tool to save bandwidth and computational requirements, exploiting the knowledge about the relative distance and positions of the elements in the 3D environment. When a user is observing the 3D space from a certain viewpoint, the other users will be perceived closer or further depending on their relative positions. When users are close, denser representations are needed and, as a consequence, high resolutions will be represented. When users are further, downgrading the LoD will reduce the amount of data to process , potentially without having a negative impacting on the perceived quality of the representation. The PC-MCU *Core Sub-system* includes a module called *LoD Selection Logic* that, after uncompressing the volumetric video, will apply a specific LoD downgrading level, depending on the users' positions.

The LoD selection strategy currently implemented is based on a simple direct reduction when a certain distance threshold is reached, and serves as an initial tool to quantify the possible gain in terms of resources usage. Additional methods to improve the Quality of Experience (QoE) are planned to be included in the next stages of the development. One interesting method is the one by Quach *et al.* [16], based on adjusting the size of the points. Anyway subjective studies need to be conducted to determine the most appropriate strategies.

Figure 3 shows an example of the PC-MCU *LoD Selection Logic*, where a user is receiving two Point Clouds placed in different relative positions. Figure 3.a shows how, without the PC-MCU, the two received streams have similar resolutions. Figure 3.b shows how the PC-MCU reduces the resolution of the further the Point Clouds, by analyzing the relative positions of the volumetric elements. Figure 4 shows a visual comparison of the two resolutions applied to the Point Cloud when affected or not by the *LOD Selection Logic*. Section 4 confirms how the application of such strategy results in a reduced consumption of bandwidth and computational resources.

3.2.3 Removal of non visible volumetric video. In 2D videoconferencing systems, all pixels from the videos are needed, because they are fully visible to all participants in a session. In 3D systems, users can freely navigate around the environment and look around. Not all the information may, then, be needed by all the users, or not along the whole session. In 3D rendering, the viewport represents the portion of the virtual environment that is visible from the viewpoint of a specific user. So, at a specific time, the users are able to visualize only certain elements and will not be able to see what that is located outside the viewport. The delivery of the whole data to all the participants would then result in an inefficient solution, having a significant impact on the consumption of bandwidth and computational resources, without enhancing the QoE. Therefore, the availability of a logic able to select the right information to be delivered, at the right time, becomes fundamental for volumetric holoconferencing systems. The presented PC-MCU implements a first version of this feature, by directly removing the data outside the user's viewport from the delivered stream for that specific user. The removal is performed by relying on the viewport information reported by each client (i.e. the xyz position and viewing angle). Other advanced (adaptive and hybrid) versions of this feature, like the ones introduced in Section 2, will be analyzed in future work. Figure 5 shows an example of how the PC-MCU considers the users viewport to remove the part of the video that would not be visualized. Section 4 will prove the benefits of this feature.



(a) Presence of data non visible by the end-user

(b) Non visible data removed by the PC-MCU

Figure 5: Example of non visible data removal operated by the PC-MCU.

3.2.4 Fusion. Traditional MCUs are typically in charge of merging the pixels of two or more videos from several users (e.g. in a mosaic view) for a videoconferencing session. If that is the case, every participant will receive and visualize a merged and composed version of the content as a single 2D video. In volumetric holoconferencing systems, the same layout and merging strategies (e.g. side-by-side) are non-applicable, because each volumetric video is an independent structure, with a geometry that has to be placed in the 3D space and will also depend on the user's position. However, the geometry of two or more volumetric videos can be fused considering a single coordinate system. The PC-MCU *Core Sub-system*, after selecting the appropriate LoD and removing the non-visible information, performs, for each user, a fusion of the representations of all visible participants and elements. This features provides key benefits, as each user client will just receive a single stream, without needing to execute N of the Point Cloud receiver and decoder modules, thanks to the optimization of such a process by the cloud-based PC-MCU.

4 EXPERIMENTAL RESULTS

The presented experimental analysis is based on the use of a first implementation of the proposed PC-MCU, compared against a system in which the volumetric videos are delivered in a peer-to-peer fashion. The goal is to assess the benefits of using the PC-MCU in a scenario with two volumetric videos compared to a baseline scenario without the use of the PC-MCU. This section starts by describing the experimental setup in Section 4.1, then it continues by detailing the evaluation methodology and metrics in Section 4.2 and concludes by presenting the obtained results in Section 4.3.

4.1 Experimental Setup

The setup used in this experiment considers a set of 10 simulated holoconferencing sessions where one of three end-users receives the Point Clouds of two other users remotely connected. The sequences considered in the tests are two among the ones available at the 8i Voxelized Full Bodies database [3]: *Red and Black* and *Longdress*. In order to facilitate a real time processing of the whole pipeline, the resolution of the two sequences have been downsampled to 65k points and 78k points, respectively.

In order recreate a typical holoconferencing system, a set of actions are forced to activate and assess the benefits of the PC-MCU functionalities (Section 3). The duration of a single session was 34 seconds. The actions are specifically performed as follows:

- **Step 1**: initial position defined to receive the two Point Clouds at their maximum resolution (65k and 78k points).
- **Step 2**: viewpoint panning, excluding *Red and Black* from the Viewport, to evaluate the computational load reduction when the *Non Visible Area Removal* function is active.
- **Step 3**: viewpoint panning, excluding *Longdress* from the Viewport, to evaluate the computational load reduction when the *Non Visible Area Removal* function is active.
- **Step 4**: user's viewpoint back to the initial position.
- **Step 5**: simulation of user moving away to evaluate the benefits when the *LoD selection* function is active.

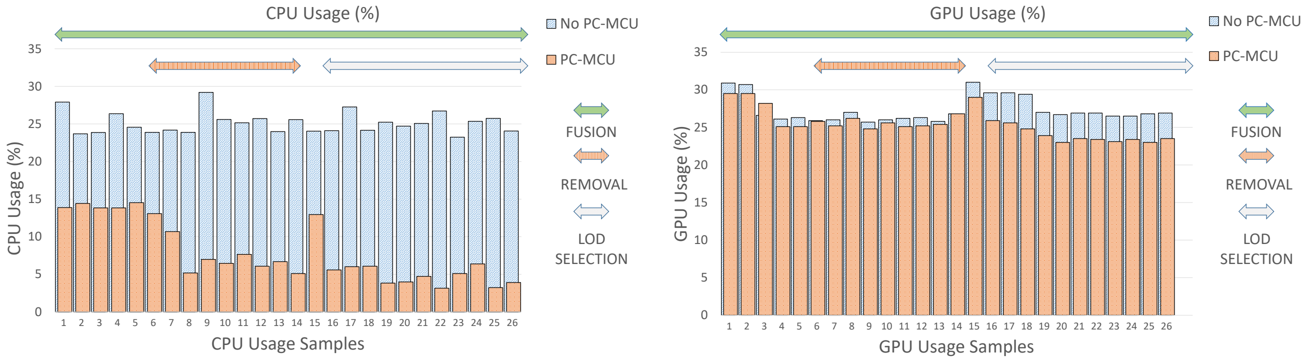
The PC-MCU *Fusion* function is always active. For the comparison, the same sequence of actions are simulated also when the volumetric videos are delivered in a peer-to-peer fashion, without the PC-MCU. All the features presented in Section 3 have been implemented according to a CPU oriented sequential programming model. At this stage, the implementation follows a sequential calls composition without any parallelization technique, nor GPU implementation. The GPU involved in this study is the one used at the end-user client machine, for the volumetric video rendering. The specifications of the machine used as client for the holoconferencing simulated sessions are the following:

- CPU: Intel Core i7-6700 @ 3.40GHz
- RAM: 16 GB @ 2133MHz
- GPU: NVIDIA Quadro K4200 4GB GDDR5
- NET: Realtek PCIe GbE Family Controller

4.2 Evaluation Methodology and Metrics

The tests consisted of launching 10 iterations with the PC-MCU plus 10 iterations without the PC-MCU, collecting samples of the following set of metrics, by using a tool from Montagud *et al.* [10]:

- **CPU** usage at the client machine (in %).
- **GPU** usage at the client machine (in %).
- **Memory** usage at the client machine (in MBs).
- **Bandwidth** consumption (in Mbps).



(a) CPU usage

(b) GPU usage

Figure 6: Percentage of CPU and GPU usage: comparison between holoconferencing sessions with and without the PC-MCU.

4.3 Results

In order to provide a thorough comparison, we have performed a simulation of 10 holoconferencing sessions without the PC-MCU and 10 sessions using it. During each session, the metrics mentioned above have been sampled along the experience. The final results show the average of each sample over the 10 iterations, showing the different benefits of the PC-MCU along an entire session.

Figure 6.a shows the reduction, in terms of percentage of CPU usage, when the PC-MCU is included in the holoconferencing session, when compared to the baseline condition (without the PC-MCU). In that figure, it is also specified the period intervals in which the PC-MCU features are active. When the PC-MCU is not used, the percentage of CPU usage do not suffer strong fluctuations along the duration of the 10 sessions, due to the constant incoming stream from the two Point Clouds. When the PC-MCU is used, it is possible to notice that, initially, when only the *Fusion* functionality is active, there is already a considerable gain in terms of CPU usage. The gain is considerable when, afterwards, the PC-MCU performs the *Non Visible Areas Removal* actions, first excluding the *Longdress* sequence, and then excluding *Rad and Black*. The CPU usage increases again when the 2 Point Clouds are newly available (see sample 15 in Figure 6.a) and then starts being furtherly reduced when the *LoD Selection* function is active.

Figure 6.b shows the evolution for the GPU usage. In this case, the benefits of the *Non Visible Areas Removal* function are less noteworthy than for the CPU usage, because the GPU is in charge of the rendering of the visible part of the 3D scenario; however, when the *LoD Selection* function is active, it is possible to notice the gain also for this metric. The GPU load is indeed reduced thanks to the lower amount of voxels needed to represent the Point Clouds.

The overall average results for the 10 iterations are summarized in Table 1, by also including additional ones regarding the RAM and bandwidth consumption. It is possible to observe how the introduction of the PC-MCU resulted in a reduction of 69% of the CPU usage, 7% of the GPU usage, 18% of memory usage and of 84% of bandwidth consumption. For completeness, the additional latency introduced by the PC-MCU has been also evaluated. In average, the PC-MCU adds a latency of 89 ms when the Point Clouds are both at their lowest resolution and an average of 160 ms when they are both visible at the maximum resolution. For the intermediate cases the latency is kept in between those values. A demo showing the virtual scenarios, the set of simulated actions, and the registered metrics can be watched here: https://youtu.be/qEENaFVeLrk.

Table 1: Average results in terms of resources consumption metrics when using and not the PC-MCU

-	No PC-MCU	PC-MCU	Δ Reduction
CPU (%)	25	7.8	68.7%
GPU (%)	27.2	25.4	6.6%
RAM (MBs)	445.5	366.3	17.85%
BW (Mbps)	92.3	14.9	83.9%

5 CONCLUSIONS AND FUTURE WORK

Videoconferencing is one of the current goals for the introduction of three-dimensional video paradigms. Indeed, the concept of holoportation is capturing the attention of the research community and industry alike. Considering the high demanding volumetric video processing, holoconferencing clients have to handle considerable volumes of information. In addition, the nature of the content, different from the 2D one, makes part of the information not visible for the end-user, or with too high resolution. To avoid not needed operations, and to optimize the heaviest ones, the proposed PC-MCU includes a set of features providing, to the end-user, the most adequate stream. The results have proven to provide, for a standard client machine, a significant reduction of bandwidth consumption and of the computational load, in terms of CPU, GPU and RAM usage. The benefits are reached with a reduced extra latency, mainly due to the sequential CPU based nature of the PC-MCU implementation. Future work will be targeted at using GPU based programming models and parallelization techniques to improve the performance in terms of latency, scalability and quality (i.e. by optimizing the media processing tasks). In addition, further research will be devoted at determining the most appropriate strategies for LoD selection and handling of the non-visible elements, based on different aspects, like distance, navigation patterns, number of volumetric media elements, etc. Finally, scalability and QoE tests will be conducted for each one of the provided features by the PC-MCU.

6 ACKNOWLEDGMENTS

This work has been funded by European Union's Horizon 2020 program, under agreement nº 762111 (VR-Together project).

REFERENCES

[1] M Reha Civanlar, Öznur Özkasap, and Tahir Çelebi. 2005. Peer-to-peer multipoint videoconferencing on the Internet. *Signal Processing: Image Communication* 20, 8 (2005), 743–754.

[2] Sylvie Dijkstra-Soudarissanane, Karim El Assal, Simon Gunkel, Frank ter Haar, Rick Hindriks, Jan Willem Kleinrouweler, 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.

[3] Eugene d'Eon, Bob Harrison, Taos Myers, and Phil A. Chou. 2017. 8i voxelized full bodies-a voxelized point cloud dataset. *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006* (2017).

[4] Global System for Mobile Communications Association. [n.d.]. Cloud AR/VR Whitepaper. https://www.gsma.com/futurenetworks/wiki/cloud-ar-vr-whitepaper/. Accessed: 2020-03-09.

[5] Silvano Gai, Tommi Salli, and Roger Andersson. 2010. *Cisco Unified Computing System (UCS)(Data Center): A Complete Reference Guide to the Cisco Data Center Virtualization Server Architecture*. Pearson Education.

[6] Mohammad Hosseini and Christian Timmerer. 2018. Dynamic adaptive point cloud streaming. In *Proceedings of the 23rd Packet Video Workshop*. 25–30.

[7] Junlin Li, Li-wei He, and Dinei Florêncio. 2007. Multi-party audio conferencing based on a simpler MCU and client-side echo cancellation. In *2007 IEEE International Conference on Multimedia and Expo*. IEEE, 84–87.

[8] David Luebke, Martin Reddy, Jonathan D Cohen, Amitabh Varshney, Benjamin Watson, and Robert Huebner. 2003. *Level of detail for 3D graphics*. Morgan Kaufmann.

[9] Rufael Mekuria, Kees Blom, and Pablo Cesar. 2016. Design, implementation, and evaluation of a point cloud codec for tele-immersive video. *IEEE Transactions on Circuits and Systems for Video Technology* 27, 4 (2016), 828–842.

[10] M. Montagud, J.A. De Rus, R. Fayos-Jordán, M. García-Pineda, and J. Segura-Garcia. 2020. Open-Source Software Tools for Measuring Resources Consumption and DASH Metrics. In *Proceedings of the 11th ACM Multimedia System Conference*.

[11] Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, et al. 2016. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 741–754.

[12] Cagri Ozcinar, Ana De Abreu, and Aljosa Smolic. 2017. Viewport-aware adaptive 360 video streaming using tiles for virtual reality. In *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2174–2178.

[13] Jounsup Park, Philip A Chou, and Jenq-Neng Hwang. 2019. Rate-utility optimized streaming of volumetric media for augmented reality. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9, 1 (2019), 149–162.

[14] Kyungmo Park. [n.d.]. Network Based Media Processing. ISO/IEC JTC1/SC29/WG11 N18270, January, 2019.

[15] Feng Qian, Bo Han, Jarrell Pair, and Vijay Gopalakrishnan. 2019. Toward practical volumetric video streaming on commodity smartphones. In *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*. 135–140.

[16] Maurice Quach, Giuseppe Valenzise, and Frederic Dufaux. 2019. Learning convolutional transforms for lossy point cloud geometry compression. In *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 4320–4324.

[17] Sebastian Schwarz, Marius Preda, Vittorio Baroncini, Madhukar Budagavi, Pablo Cesar, Philip A Chou, Robert A Cohen, Maja Krivokuća, Sebastien Lasserre, Zhu Li, et al. 2018. Emerging MPEG standards for point cloud compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9, 1 (2018), 133–148.

[18] Gary A Thom. 1996. H. 323: the multimedia communications standard for local area networks. *IEEE communications Magazine* 34, 12 (1996), 52–56.

[19] MH Willebeek-LeMair, Dilip D Kandlur, and Z-Y Shae. 1994. On multipoint control units for videoconferencing. In *Proceedings of 19th conference on local computer networks*. IEEE, 356–364.