|  |  |
| --- | --- |
| Agenda Item: | 10.9 |
| Source: | **Fraunhofer HHI** |
| Title: | **[5G\_STAR] pCR Proposed architecture for streaming volumetric video** |
| Document for: | **Discussion and agreement** |

1. Introduction

In the previous meeting 3GPP SA4 111-e, in S4-201423, a use case titled “Streaming of Volumetric video” was proposed. The proposed use case was agreed and integrated in the TR 26.998[1]. The delivery mechanism for the use case is characterized in two options; Streaming and Split-rendering. In this proposal, we study and extend existing split-rendering architectures present in TR 26.928[2] to design a system architecture which suits the use case for the split-rendering delivery mechanism. Implementations of such mechanisms can be found in [3] and [4].

1. XR view(s) split-rendering architecture for volumetric video streaming

## 2.1 Introduction

In TR26.928[2], Raster-based Split rendering (section 6.2.5) and Generalized XR rendering (section 6.2.6) relies on splitting the XR workload on a powerful XR Server (on the edge/cloud) and on the XR Device (with pose correction). The objective of the split architecture is facilitated by generating a rasterized-view which is encoded, delivered and decoded on the XR device. In the proposed architecture, the XR Server is capable of playing back a volumetric media sequence. The rasterized-view in sections 6.2.5 and 6.2.6 corresponds to the XR viewport. In the below extended architecture, the XR Server generates stereo XR view(s) for volumetric object(s) as shown in Figure 1 instead of the XR viewport. The rasterized XR view(s) for each volumetric object(s) are encoded and packed into a decodable 2D format. The 2D pre-rasterized encoded media along with associated XR metadata are delivered over a 5G network.

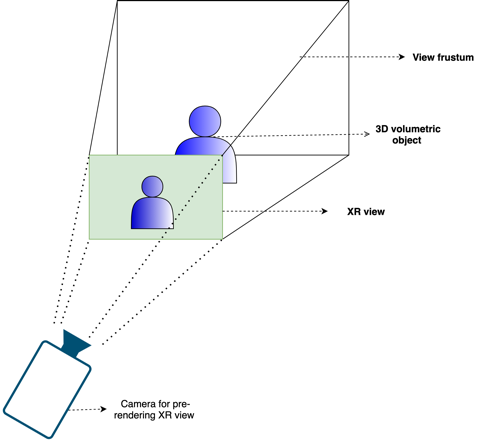


Figure 1. Pre-rendering a XR view for a volumetric video object on XR server

The XR device decodes the 2D pre-rasterized media. The 2D pre-rasterized media is integrated in the local XR scene which is generated by the XR device. The 2D pre-rasterized media(s) are anchored at positions specified by their respective XR metadata information in the XR scene. Upon a change in viewing position and/or orientation of the XR device, the pre-rendered 2D XR view is adjusted to align with the position and orientation of the XR viewer as shown in Figure 2. An position/orientation update (collected from different sensors) is sent in a regular period to the XR server to pre-render appropriate XR view(s) of the volumetric video objects.

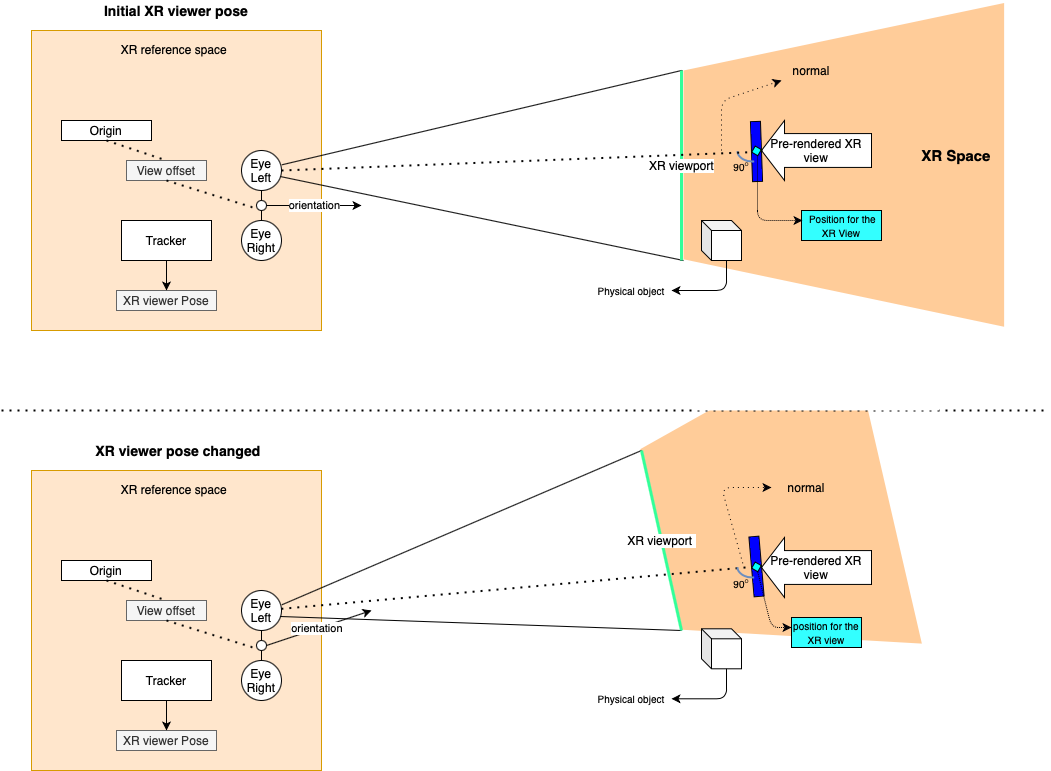


Figure 2. Positioning and adjustment to pre-rendered XR view in the XR space

## 2.2 Architecture

In Figure 3, an architecture is shown for which the XR server pre-renders an XR View for a volumetric video object. The pre-rendered XR view is encoded into a simpler format (i.e., 2D video) to be processed by the device (e.g., it may provide additional metadata that is delivered with the pre-rendered XR view). The device recovers the 2D media and does the compositing. At the final compositing stage, the pre-rendered XR View is positioned in the MR scene described by XR metadata. The device does a final render of the MR scene for glass-type devices and does adjustments based on local correction on the actual pose.

- XR graphics workload is split into rendering workload on a powerful XR server (on the Edge/cloud) and simpler XR processing on the XR device

- XR server is capable to play multiple volumetric video(s)

- The rasterised view of the volumetric video(s) are encoded in to a simple 2D format

- Associated spatial audio for the volumetric video(s) are also pre-rendered into a simple audio format

- The XR device is capable to generate a local semantic information of the surrounding using sensor data for spatial mapping and tracking

- The pre-rendered XR view(s) 2D media are decoded

- The decoded pre-rendered XR view(s) may be subjected to minimal pixel processing (e.g., background removal)

- The pre-rendered XR view(s) are positioned as described by the XR metadata during final compositing

- The pre-rendered XR view(s) may also be occluded by any real-world object physically present and overlapping the MR scene

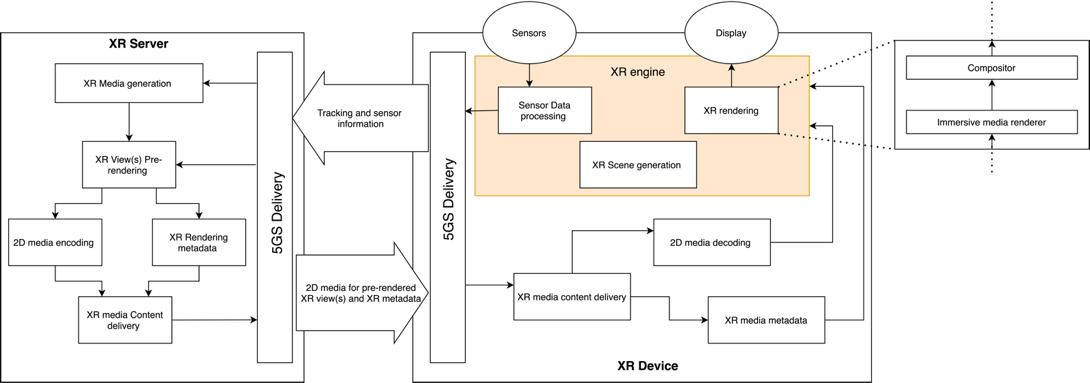


Figure 3. Split Rendering architecture for single/multiple pre-rendered XR view(s)

The architecture above applies to devices Type 1 and Type 3 defined section 4 in TR 26.998[1]. As seen in Figure 3, the XR rendering consists of the functional blocks such as “Immersive media renderer” and “Compositor”. Note that the architecture does not apply to devices Type2 as Type 2 devices are based on pre-rendered viewports, while the architecture above is based on XR views that are mapped to some 2D planes at certain positions in the scene that are not necessarily the viewport.

Such an approach needs careful considerations on the formats of projected media and their compression with media decoders. Also important is distribution of latencies to different components of the system. More details and breakdown of the architectures is necessary. The interfaces in the device however are aligned with the general structure defined above.

<NOTE: Mapping to 5G System, e.g. 5G Media Streaming to be provided>

## 2.3 Basic Procedures

The following call flow highlights the key steps:

1. An XR Device starts an XR application and initialises an XR scene

2) The XR Device connects to the network and send information to the XR server such as following

a) Sends static device information (supported decoders, viewport size, supported formats)

b) Requests to stream desired volumetric video objects and their respective pose in the XR Scene

3) Based on this information, XR server on the network sets up encoder and formats

4) Loop

a) XR Device collects XR pose (or a predicted XR pose)

b) XR Pose is sent to XR Server

c) The XR Server uses the pose to pre-render XR Views for multiple volumetric video object resulting in one or multiple rendering buffers, possibly with different update frequencies

d) The rendering buffers are encoded with 2D encoder(s) (potentially as a single 2D video)

e) The compressed media is sent to XR device along with additional XR metadata that describes the media

f) The XR device decompresses the multiple buffers and sends these to the XR rendering engine

g) The XR rendering engine takes the buffers, the rendering pose assigned to the buffers

h) The XR rendering engine composites a XR scene whereby, the buffered XR views are accurately positioned as described by the XR metadata.

i) The latest XR pose is used (added with local pose correction) to create the finally rendered XR viewport of the XR scene for the XR device

## 2.4 Media Exchange format and profiles

A pre-rendered XR view for volumetric video object(s) is rasterized according to the XR pose received on the XR server. The pre-rendered XR view(s) are encoded into simple 2D format. The pre-rendered XR view(s) may be packed into a single 2D video in case of multiple objects. The encoded media constitutes of XR frames for stereo view.

The format of the 2D media, e.g., resolution, is limited by media profiles.

< NOTE: More clear information about format for perfect use experience needs to be provided. The object inserted into the scene does not cover the whole viewport. Maybe a way forward is defining a pixel density per degree>

The final compositing of the XR viewport happens on the XR device taking into account different decoded pre-rendered XR views.

## 2.5 Delivery protocol and Quality of Services

Low latency delivery protocols (i.e. RTP) are essential to achieve a fully immersive experience so that an update on any change in XR pose is made available as soon as possible. In this case, typically less than round trip latency of 50ms is a requirement for a fully immersive experience. This constitutes the end-to-end latency between the user motion and the XR device rendering. The XR views must be correctly positioned. Usually, the spatial mapping information is tracked locally and the scene semantic are constructed to identify different spatially located real-physical objects.

< NOTE: The value of 50ms needs more information to be provided>

## 2.6 Potential Normative works

In the context of the proposed architecture, the following potential standardisation needs are identified:

* Edge computing discovery and capability discovery for XR view(s) split rendering
* A XR split rendering application framework
* More flexible 5QIs and QoS support in 5G System for generalized split rendering addressing differentiated end-to-end latency requirements in the range of 10ms up to potentially 50ms and with bitrate guarantees
* XR metadata (e.g., defining XR views – position of planes for the 2D video)
* Media profiles
* Content delivery protocols

1. Proposal

The proposal is to add Section 2 to the TR 26.998 as a potential system architecture solution for use case 18 in TR 26.998[1].

# Reference

[1] 3GPP Technical report 26.998, Support for 5G glass-type Mixed/Augmented reality devices

[2] 3GPP Technical report 26.928, Extended Reality in 5G

[3] Serhan Gül, Dimitri Podborski, Jangwoo Son, Gurdeep Singh Bhullar, Thomas Buchholz, Thomas Schierl, Cornelius Hellge, “Cloud Rendering-based Volumetric Video Streaming System for Mixed Reality Services” Proceedings of the 11th ACM Multimedia Systems Conference (MMSys'20), June 2020

[4] <https://www.hhi.fraunhofer.de/5GXR>