**3GPP TSG-SA WG4 Meeting #126 S4-231797**

**Chicago, US, 13 – 17 November 2023**

|  |
| --- |
| *CR-Form-v12.2* |
| **PSEUDO CHANGE REQUEST** |
|  |
|  | **26.966** | **pCR** |  | **rev** | **-** | **Current version:** | **0.1.0** |  |
|  |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME | **x** | Radio Access Network |  | Core Network | **x** |

|  |
| --- |
|  |
| ***Title:***  | Layered XR rendering |
|  |  |
| ***Source to WG:*** | Nokia  |
| ***Source to TSG:*** | S4 |
|  |  |
| ***Work item code:*** | FS\_HEVC\_Profiles |  | ***Date:*** | 04-11-2023 |
|  |  |  |  |  |
| ***Category:*** | B |  | ***Release:*** | Rel-18 |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(mirror corresponding to a change in an earlier release)****B*** *(addition of feature),* ***C*** *(functional modification of feature)****D*** *(editorial modification)*Detailed explanations of the above categories canbe found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | *Use one of the following releases:Rel-8 (Release 8)Rel-9 (Release 9)Rel-10 (Release 10)Rel-11 (Release 11)…Rel-16 (Release 16)Rel-17 (Release 17)Rel-18 (Release 18)Rel-19 (Release 19)* |
|  |  |
| ***Reason for change:*** | Introduction of a new scenario related to layered rendering. |
|  |  |
| ***Summary of change:*** | This CR proposes an additional scenario related to layered rendering. In many rendering cases, the scene to be rendered is composed by different elements, including UI. When the device is running pose-correction in the device, this may drop QoE as some parts of the rendered image are more or less sensitive to the pose correction. To maximize QoE, the layer information can be carried out to the end user device with side-metadata driving the pose-correction. This can be achieved through multi-layer encoding or through a single-layer approach complemented with depth/alpha channels. This CR add a new scenario and a solution for it. |
|  |  |
| ***Consequences if not approved:*** | Pose correction in the context of TR 26.966 will suffer from QoE drops.  |
|  |  |
| ***Clauses affected:*** | 5.4 (new), 6.0 and 6.8 (new). |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **x** |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  | **x** |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  | **x** |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** |  |

|  |
| --- |
| 1st Change |

5.4 Scenario #4: Pose correction optimisation

5.4.1 Overview

This scenario deals with a split-rendering case where the device is running a pose correction method (e.g., using ATW). While pose correction is a good solution to cope with the latency introduced by the roundtrip communication and the rendering, it can introduce visual artifacts if only 2D projected images are used. As an example, a rendered scene may be composed by multiple elements having different sensitivity to time warping. For instance, the user-interface (UI) does not need to be corrected as its position won’t change in the user’s Field of View (FoV). A 3D object near the user may benefit from a time warping as the pose correction would address parallax differences. The far away background similarly to the UI does not need warping as parallax fall off in the distance. This is illustrated in the Figure below.



Figure 1: Illustration of layering in rendering

To maintain the effectiveness of pose correction, a rendering engine may apply segmentation and generate multiple layers of projected texture images that may be handled differently based on their time-warping sensitivity. Those different texture layers could be encoded and processed separately in multiple video streams but could also be encoded in a single stream with additional depths and alpha channels.

To drive the pose-correction and maximize the QoE, additional information may provided to support segmentation into layers and to support the pose correction how the different texture layers should be handled by the pose correction engine. However, such optional metadata is currently not supported by OpenXR APIs.

Generally, the carriage of depth and alpha channels in the video bitstreams for proper scene and UI restitution allows to improve pose correction. New video codecs have the potential to address this scenario in a bandwidth efficient manner.

Editor’s NOTE: MeCAR is in the process to define a simple format for this purpose. It should be referenced.

5.4.2 Review of previous work

The same previous work as Scenario #1 can be considered for this use-case. Multi-layer extension includes mechanisms to carry additional channel such as alpha channel.

The carriage of depth or alpha auxiliary channels has not been addressed until now.

5.4.3 Evaluation criteria and methodology

1. Assessment/discussion of hardware impact: there are two possibilities for this:
	1. There is existing hardware product-grade support for the tool. In that case, refer to the example hardware.
	2. There is no existing hardware support. In this case, a discussion/description with justifications on the expected impact on hardware implementation is provided, or reference to existing demos etc.
2. Codec performance evaluation can be evaluated in two possible ways:
	1. For single layer case, the performance evaluation of impact on bandwidth will be determined by the overhead introduced by adding additional channels to the video (alpha, depth, …) compared to traditional approach. It is expected that the additional cost is negligible.
	2. For multi-stream case, the performance evaluation of impact on bandwidth will be determined by measuring the overhead introduced by multiple encodings compared to a single-layer approach. It is expected that the additional cost is low.

|  |
| --- |
| End of change |

|  |
| --- |
| 2nd Change |

# 6 Solutions

## 6.0 Mapping of Solutions to Scenarios

Table 6.0-1: Mapping of Solutions to Scenarios

|  |  |  |
| --- | --- | --- |
| Solution # | Solution Title | Scenario(s) |
| #1.1 | HEVC simulcast | #1.1 |
| #1.2 | Multiview HEVC coding | #1.1 |
| #2.1 | HEVC 4:2:0 coding | #2 |
| #2.2 | HEVC 4:2:2 coding | #2 |
| #2.3 | Native 4:4:4 coding - HEVC Main 4:4:4 profiles | #2 |
| #2.4 | Derived 4:4:4 coding - Layered use of HEVC 4:2:0 profiles | #2 |
| #3.1 | Scalable HEVC coding | #3 |
| #4.1 | MV-HEVC with auxiliary depth/alpha channels ~~and side metadata~~ | #4 |
|  |  |  |

|  |
| --- |
| End of change |

|  |
| --- |
| 3rd Change |

6.8 Solution #4.1: MV-HEVC with auxiliary depth/alpha channels ~~and side metadata~~

6.8.1 Introduction

This solution explores the use of auxiliary alpha or depth channels, complementary to an HEVC bitstream to enable rendering optimization based on the auxiliary alpha/depth channels~~.data, and driven by side metadata indicating how to interpret the received channels~~. This can be done in two ways: ~~attached to a single HEVC bitstream, but also attached to multiple HEVC bitstreams used to carry the different rendered layers.~~

* Solution 4.1-A: An MV-HEVC bitstream carrying a single video layer and alpha/depth video channels.
* Solution 4.1-B: Multiple MV-HEVC bitstreams, each carrying a texture layer and with alpha/depth channels.

6.8.2 High-level Description

This solution explores the usage of MV-HEVC to carry the alpha and depth information as auxiliary channels. The carriage of such data is described in clause 6.8.2.1.

Additional information on possible SEI messaging transmitted to drive pose-correction is also documented for information in clause 6.8.2.2 but is not supported at this stage by OpenXR APIs and thus is excluded from this evaluation.

6.8.2.1 Carriage of alpha and depth auxiliary channels with MV-HEVC

The usage of auxiliary pictures in HEVC is part of the multi-layer extensions. The carriage of auxiliary data such as depth or alpha channels is defined by the ScalabilityId signalled through the scalability\_mask\_flag in the Video Parameter Set (VPS). This is possible by configuring the scalability mask index to ‘3’, the value reserved for enabling “Auxiliary” as scalability dimension, as highlighted in yellow in Table 1.

Table 1: Mapping of ScalabilityId to scalability dimensions, as specified in HEVC (see Table F.1)

|  |  |  |
| --- | --- | --- |
| **Scalability mask index** | **Scalability dimension** | **ScalabilityId mapping** |
| 0 | Texture or depth | DepthLayerFlag |
| 1 | Multiview | ViewOrderIdx |
| 2 | Spatial/Quality scability | DependencyId |
| 3 | Auxiliary | AuxId |
| 4-15 | Reserved |  |

The selection of alpha/depth auxiliary pictures is then set by the AuxId which can be configured as defined in the Table below. Setting value ‘1’ would signal the auxiliary picture is an Alpha plane while ‘2’ would indicate a depth picture. Additional information about how to interpret and process those channels can be carried in SEI messages, through the Alpha channel and depth representation information SEI messages.

Table 2: Mapping of AuxId to the type of auxiliary pictures, as specified in HEVC (see Table F.2)

|  |  |  |  |
| --- | --- | --- | --- |
| **AuxId** | **Name of AuxId** | **Type of auxiliary pictures** | **SEI message describing interpretation of auxiliary pictures** |
| 1 | AUX\_ALPHA | Alpha plane | Alpha channel information |
| 2 | AUX\_DEPTH | Depth picture | Depth representation information |
| 3..127 |  | Reserved |  |
| 128..159 |  | Unspecified |  |
| 160..255 |  | Reserved |  |

6.8.2.2 Additional information on SEI messages

Additionally, alternative SEI messages can be carried to indicate how the picture texture should be rendered and processed in the device, based on information carried through the alpha or depth channel. In the case of pose-correction parameters, the GUI can be isolated from the rest of the picture through specific depth ranges, or alpha values. The strength or sensibility to the pose-correction can be also indicated for each depth or alpha range value.

A specific SEI message is needed to carry out this information, which can be done for example through a private ITU-T 35 message, or by defining a new one in MPEG. A message carrying the desired information is provided in the Table 3. The provided SEI handles all possible scenarios.

Table 3: Possible payload for pose-correction parameters SEI

|  |  |
| --- | --- |
| pose\_correction\_parameters( payloadSize ) { | **Descriptor** |
|  pcp\_metric | u(1) |
|  pcp\_n\_intervals | u(16) |
|  for ( i=0; i<pcp\_n\_intervals; i++){ |  |
|  pcp\_interval\_upper\_bound[i] | u(16) |
|  pcp\_interval\_correction\_sensitivity[i] | u(16) |
|  } | u(1) |
| } |  |

With the following semantic:

* **pcp\_metric** indicates what metric is used to extract different layers from video texture. 0 means alpha ranges are used, 1 means depth ranges are used.
* **pcp\_n\_intervals** indicates in how many intervals the layering is described for the selected metric.
* **pcp\_interval\_upper\_bound**[i] indicates the upper bound value of the i-th interval.
* **pcp\_interval\_correction\_sensitivity**[i] indicates the intensity of pose correction that should be applied on the i-th interval in the received frame. 0 means no pose correction should be applied, other values describes different degrees of pose-correction sensitivity.

The SEI messaging driving the pose-correction is depicted here for information but is currently not supported by OpenXR APIs and is then not included in the performance evaluation.

6.8.3 Evaluation

6.8.3.1 Assessment/discussion of hardware impact

~~This potential solution requires the device to perform two main tasks:~~

1. ~~Demuxing and decoding of auxiliary alpha/depth pictures, forwarding of the decoded data to the pose-correction engine.~~
2. ~~Demuxing and forwarding of the SEI messages to the pose-correction engine.~~

This potential solution requires the device to decode the auxiliary channels and forward them to the XR runtime. This task is expected to be ~~Those two operations are~~ straightforward and light in terms of processing. A software update on top of an existing 4:2:0 decoder is expected to be sufficient to enable the feature with minimal complexity and power consumption overhead. To assess this impact, a complexity evaluation needs to be included in the evaluation.

6.6.3.2 Codec performance evaluation

In this scenario, additional data is carried, through auxiliary pictures ~~and SEI messages to drive the rendering and processing done in the device~~. As two solutions ~~scenarios~~ are possible, the performance should be evaluated as follows:

* ~~Scenario~~ Solution 4.1-A: a MV-HEVC bitstream + alpha/depth channel ~~+ SEI messages~~
* ~~Scenario~~ Solution 4.1-B: multiple HEVC bitstreams ~~+ SEI messages~~

For the two scenarios, the performance is measured in terms of BD-RATE overhead, compared to the legacy single-layer approach. This performance measurement needs to be included in the evaluation.

Editor’s NOTE: MeCAR is in the process to define a simple format for this purpose. The usage of MV-HEVC with auxiliary pictures should be evaluated against this reference as well.

|  |
| --- |
| End of change |