**3GPP TSG-SA WG4 Meeting #131-bis-eS4-250536**

**Online, 11 - 17 April 2025**

**Source: Philips, Nokia, InterDigital**

**Title: [FS\_Beyond2D] Evaluation of multi-view plus depth with preliminary results**

**Agenda item: 9.7**

**Document for: Agreement**

**1. Introduction**

This document provides input on the evaluation of Scenario 3: Streaming of Multi-view plus depth Produced Content (TR 26.956 0.3.0, section 7.4), with some first results. Currently only one sequence was prepared. The aim is to add 2-4 more sequences for this scenario.

**2. Proposal**

The proposal is to add the provided information to a new section x.y of the PD.

**3. Attachments**

* Configuration files
* Encoder log files
* Objective results
* Pose trace videos (on a temporary FTP)

## x.y Evaluation of multi-view plus depth produced content

### x.y.1 Test sequences

The evaluation has been performed on the sequences listed in Table xy1.

[Ed.(BK): The aim is to add 2-4 more sequences for this scenario. Add a decription of each sequence.]

The Breakfast sequence is part of the MIV CTC [XY1] but has not been used for the development of ISO/IEC 23090-12:2023.

Table xy1: Test sequences for the evaluation of the scenario

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sequence** | **ID** | **Provider** | **Frames** | **Resolution** | **Bit depth** | **Color format** |
| Breakfast(Figure xy1) | D02 | InterDigital | 97 @ 30 Hz3.2 s | 1920 x 10805 x 3 views | texture: 10bdepth: 16b | texture: 4:2:0 BT.709depth: 4:2:0 full range linear |
| ... |  |  |  |  |  |  |
| ... |  |  |  |  |  |  |
| ... |  |  |  |  |  |  |

 

Figure xy1: Breakfast sequence (view 7, frame 0)

### x.y.2 Software

The software that has been used for the evaluation of the scenario is listed in Table xy2. All software has been built from source using Python 3.10, LLVM 18.1.8 with help of the install.py script of TMIV, as follows:

*# environment with python, clang and clang++ on the path*

git clone https://gitlab.com/mpeg-i-visual/tmiv.git

cd tmiv

python -m venv venv

. venv/bin/activate

python -m pip install --upgrade pip

pip install -r requirements.txt

scripts/install.py clang-release

Table xy2: Software used for the evaluation of the scenario

|  |  |  |
| --- | --- | --- |
| **Software** | **URL** | **Version** |
| Test model for MPEG immersive video (TMIV) | <https://gitlab.com/mpeg-i-visual/tmiv> | 24.0[[1]](#footnote-2) |
| HEVC test model (HM) | <https://vcgit.hhi.fraunhofer.de/jvet/HM> | 18.0 |
| Quality metrics for immersive video (QMIV) | <https://gitlab.com/mpeg-i-visual/qmiv> | 2.0 |

HM 18.0 and Kvazaar 2.3.1 have been compared in MPEG context for the coding of MIV video sub-bitstreams [XY2]. HM 18.0 was selected for this study because it has a better rate-distortion characteristic in general. However, because HM lacks support for delta QP maps, packed video support was disabled in TMIV.

### x.y.2 Encoding of bitstreams

Encoding of MIV bitstreams using TMIV and HM involves three steps:

1. Run the TMIV encoder to output a raw YUV video file for each video sub-bitstream, and a partial MIV bitstream with patch parameters and video parameters. The main work of the TMIV encoder is to prune pixels, patch patches, and generate atlas frames.
2. Run HM TAppEncoder to encode each YUV file using a fixed QP.
3. Run the TMIV multiplexer to combine the partial MIV bitstream and the coded video sub-bitstream into a full MIV bitstream (a V3C sample stream).

The texture QP's are chosen per sequence to cover a similar bit rate range. In the Common test conditions for MPEG immersive video [XY1] an equation (1) is specified to select a geometry QP (*q'*) based on a texture QP (*q*), to avoid per-sequence tuning, and this study uses the same equation. It has been verified using expert viewing. In general a lower QP is used for geometry than for texture because geometry errors are more visible.

All sequences have been encoded using the configurations in Table xy3. The configuration files are attached to this document.

- The ***Anchor*** *full views* (FV) condition serves as an anchor whereby each component of each view is encoded separately as a HEVC Main 10 bitstream. TMIV and MIV are only used for practical reasons (re-use of scripts and carrying metadata).

- The *MPEG* *MIV main ~~anchor~~* (A) condition is the MIV CTC anchor, defined in ISO/IEC JTC 1/SC 29/WG 04 N 0659. It results in two atlases, each with a texture and geometry component.

- The *Synthesize center view* (SCV) condition produces a single atlas with a texture and geometry component. It has an atlas with a single synthesizerd center view plus patches of the source views. The aim of this condition is to provide a level 2.5 result by lowering the pixel rate compared to the MPEG anchor.

[Ed.(BK): Explain better how and why these conditions were created. Make it clear that SCV is meant to target a lower level that is more practical for mobile devices, and explain how this was done.]

Table xy3: Encoder conditions

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Profile** | **Abbreviation** | **Directory name** |
| Full views | HEVC Main 10 | FV | config/full\_views |
| MIV main anchor | HEVC Main 10 MIV 2(FDIS 23090-12:—) | A | config/miv\_main\_anchor |
| Synthesize center view | HEVC Main 10 MIV Extended(23090-12:2023) | SCV | config/synthesize\_center\_view |

Encoding was performed by running the encode.py script of TMIV with appropriate parameters. It executes the TMIV Encoder, HM, and the TMIV Multiplexer with appropriate parameters. For example:

*TMIV\_DIR*/bin/encode.py -i *INPUT\_DIR* -o out -s D02 -n 32 \

 -r RP0 -f 0 -v HM -j 4 -t *TMIV\_DIR* \

 --config-dir share/config \

 -c config/synthesize\_center\_view/SCV\_1\_TMIV\_encode.json \

 -m config/synthesize\_center\_view/SCV\_3\_TMIV\_mux.json \

 -C share/config/hm/encoder\_randomaccess\_main10.cfg

The only substantial difference between the encoder conditions is the TMIV encoder configuration because the TMIV multiplexer configuration is trivial and the HM configuration is kept to the same random-access configuration for all conditions.

[Ed.(BK): Current results are based on 32 frames with 128 frame pose trace videos. The intent is to use at least 65 frames and at least 260 frame pose trace videos.]

The rate point RP0 is a result without coding of the video sub-bitstreams that can be used to determine how much quality is lost by the pixel pruning prior to video coding. Rates RP1 .. RP4 correspond to the following QP values in Table xy4.

Table xy4: QP values (texture, geometry) of Breakfast for all encoder conditions

|  |  |  |  |
| --- | --- | --- | --- |
| **Rate point** | **Full views** | **MIV main anchor** | **Synthesize center view** |
| RP1 | 25, 6 | 25, 6 | 25, 6 |
| RP2 | 30, 14 | 30, 14 | 30, 14 |
| RP3 | 35, 14 | 35, 14 | 35, 14 |
| RP4 | 43, 20 | 43, 20 | 43, 20 |

### x.y.3 Results

#### x.y.3.1 Example atlas frames

The full views (FV) condition encodes each component of each view separately, e.g. resulting in 30 separate 1920 x 1080 videos for the Breakfast sequence. Figure xy2 and Figure xy3 provide examples of atlas frames for the MIV main anchor (A) and synthesize center view (SCV) conditions. A comparison of pixel rates is provided in Tabe xy5. Note that the size of each atlas depends on the sequence and on the encoding condition. This is because TMIV calculates the atlas frame size based on a number of inputs.

   

Figure xy2: Video components of condition A with left to right: texture for atlas 0 and 1, geometry for atlas 0 and 1

 

Figure xy3: Video components of condition SCV with left texture and right geometry

#### x.y.3.2 Pixel rate

The pixel rates per video sub-bitstreams and the aggregate pixel rate are depicted in Table xy5.

[Ed.(BK): Provide level information for each condition and sequence. As discussed HEVC Main 10 Level 6.1 is the limit for high-end devices, but the client also needs to perform view synthesis so the practical level is probably lower. Explain that the aim of SCV is to have a lower HEVC level.]

[Ed.(BK): There is no need for anchor because these is a new representation. Use multiple conditions and explain how they relate to hardware requirements.]

[Ed.(BK): Use the same kind of geometry upsampling for FV as for SCV.]

Table xy5: Pixel rates for all sequences and conditions:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Condition** | **Sequence** | **Components** | **Sizes** | **Aggregate size**(# luma samples) | **Aggregate luma sample rate** |
| FV Anchor | D02 | 15 x texture15 x depth | 1920 x 10801920 x 1080 | 59.3 M | 1.74 G/s |
| A | D02 | 2 x texture2 x geometry | 1920 x 4608960 x 2304 | 21.1 M | 0.618 G/s |
| SCV | D02 | texturegeometry | 2880 x 24321440 x 1216 | 8.3 M | 0.245 G/s |

#### x.y.3.2 Rate-distortion characteristics

The aggregate bit rates are provided in Table xy6, and average IV-SSIM values are provided in Table xy7.

[Ed.(BK): Tune QP values to have more overlap between conditions. After QP tuning, provide per-sequence rate-distortion graphs to make it easier to interpret the objective results; and calculate BD-rates to compare the conditions.]

Table xy6: Aggregate bit rates for all sequences and conditions:

|  |  |  |
| --- | --- | --- |
| **Condition** | **Sequence** | **Aggregate bit rate [Mb/s]** |
|  |  | **RP1** | **RP2** | **RP3** | **RP4** |
| FV | D02 | 69.8 | 42.7 | 26.9 | 12.7 |
| A | D02 | 40.1 | 23.1 | 13.9 | 6.1 |
| SCV | D02 | 9.5 | 5.7 | 3.5 | 1.6 |

Table xy7: IV-SSIM values averaged over all source views, for all sequences and conditions:

|  |  |  |
| --- | --- | --- |
| **Condition** | **Sequence** | **Average IV-SSIM** |
|  |  | **RP0** | **RP1** | **RP2** | **RP3** | **RP4** |
| FV | D02 | 0.998 | 0.986 | 0.981 | 0.974 | 0.947 |
| A | D02 | 0.990 | 0.981 | 0.978 | 0.971 | 0.946 |
| SCV | D02 | 0.972 | 0.964 | 0.959 | 0.952 | 0.926 |

#### x.y.3.3 Pose trace videos

For each bitstream, that is for each sequence for each encoder condition and for each rate RP0 .. RP4, three pose trace videos have been rendered. A bitstream can be decoded and rendered using a command like this:

*TMIV\_DIR*/bin/TmivDecoder -j 1 -n 32 -N 128 -s D02 -r RP3 -P p01 \

 -c config/synthesize\_center\_view/SCV\_4\_TMIV\_decode.json \

 -p inputDirectory out -p outputDirectory out \

 -p configDirectory share/config

The decoder configurations differ only in path formats: there is no out-of-band information for RP1 .. RP4.

The pose trace videos are available for informal expert viewing at the following links:

<https://fileshare.ehv.campus.philips.com/private/20250422-RD58423C1CCB54A24970EE11A66D162DC>

sftp://anonymous@fileshare.ehv.campus.philips.com/private/20250422-RD58423C1CCB54A24970EE11A66D162DC/

This directory will be automatically removed after Tuesday, April 22, 2025.

### x.y.4 References

[XY1] Common test conditions for MPEG immersive video, ISO/IEC JTC 1/SC 29/WG 04/N 659, April 2025, url: <https://www.mpeg.org/wp-content/uploads/mpeg_meetings/150_OnLine/w25084.zip>, Online.

[XY2] Encoder guidelines for MPEG immersive video, ISO/IEC JTC 1/SC 29/WG 04/N 660, April 2025, url: <https://www.mpeg.org/wp-content/uploads/mpeg_meetings/150_OnLine/w25085.zip>, Online.

1. To be pulished soon. A zip-file can be provided to non-MPEG members on request. [↑](#footnote-ref-2)