ETSI TR 126 999 V18.0.0 (2024-05)

TECHNICAL REPORT

LTE;
5G;
Virtual Reality (VR) streaming interoperability and characterization
(3GPP TR 26.999 version 18.0.0 Release 18)

Reference

RTR/TSGS-0426999vi00

Keywords

5G,LTE

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from:
https://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

If you find a security vulnerability in the present document, please report it through our Coordinated Vulnerability Disclosure Program:
https://www.etsi.org/standards/coordinated-vulnerability-disclosure

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Legal Notice

This Technical Report (TR) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under https://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Intellectual Property Rights 2

Legal Notice 2

Modal verbs terminology 2

Foreword 5

1 Scope 7

2 References 7

3 Definitions of terms, symbols and abbreviations. 8
3.1 Abbreviations 8

4 Source content material 8
4.1 Introduction 8
4.2 Orange test sequence 8
4.2.1 The VR Experience 8
4.2.2 Video capture 9
4.2.3 Audio configuration 10
4.2.4 Final product 11
4.3 InterDigital test sequence 12
4.3.1 The VR Experience 12
4.3.2 Video Capture 14
4.3.3 Final Product 15

5 Test results 15
5.1 Introduction 15
5.2 ABR Streaming of Tiled Video 15
5.2.1 Design 16
5.2.2 Representative Viewport V 17
5.2.3 Full sphere bit rate estimation 17
5.2.4 ABR Algorithms 17
5.2.5 Implementation 18
5.2.6 Experiments 18
5.2.6.1 Head Motion 19
5.2.6.2 Network conditions 19
5.2.6.3 Metrics 20
5.2.7 Results 20
5.2.7.1 Stable Network Conditions 20
5.2.7.1.1 Stall Events 20
5.2.7.1.2 Quality Variation 20
5.2.7.1.3 Throughput 22
5.2.7.2 Variable Network Conditions 22
5.2.7.2.1 Stall Events 22
5.2.7.2.2 Adaptability 22
5.3 Streaming of Tiled Video using Viewport Margins 23
5.3.1 Head Motion Aware (HMA) Margins 23
5.3.2 Experimental Setup 24
5.3.3 Results 24

Annex A: Process steps for video 26

Annex B: Test Vectors 29
B.1 Introduction 29
B.2 Uploading and Hosting Test Vectors 29

Annex C: MPEG OMAF 2nd Edition Nokia public source code 31
C.1 Introduction 31
C.2 New features in Nokia OMAF public release 31

Annex D (informative): Change history 33

History 34

Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

Y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

shall indicates a mandatory requirement to do something

shall not indicates an interdiction (prohibition) to do something

The constructions ”shall“ and “shall not” are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions “must“ and “must not” are not used as substitute for ”shall“ and “shall not”. Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

Should indicates a recommendation to do something

should not indicates a recommendation not to do something

may indicates permission to do something

need not indicates permission not to do something

The construction “may not” is ambiguous and is not used in normative elements. The unambiguous constructions “might not” or “shall not” are used instead, depending upon the meaning intended.

Can indicates that something is possible

cannot indicates that something is impossible

The construction “can“ and “cannot” are not substitute for “may“ and “need Not”.

Will indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

will not indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

might indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

might not indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

is (or any other verb in the indicative mood) indicates a statement of fact

is not (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions “is“ and “is not” do not indicate requires.

1 Scope

The present document provides reference test material and test results for improved usability of technologies in [2].

The specification [2] includes several VR media profiles for video and a single media profile for audio with different configuration options. The specification focuses primarily on interoperability requirements for VR360 applications, but does not address performance characterization of the solutions. In order for content providers and the rest of the ecosystem to be able to select and configure the technologies defined in [2] and to generate content for streaming applications, collecting such information would be most valuable.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21“905: “Vocabulary for 3GPP Specifica”tions”.

[2] 3GPP TS 26.118: “Virtual Reality (VR) profiles for streaming applications”.

[3] 3GPP TS 26.260: “Objective test methodologies for the evaluation of immersive audio systems”.

[4] Nokia OMAF implementation, https://github.com/nokiatech/omaf.

[5] Chenghao Liu, Imed Bouazizi, Miska M. Hannuksela, and Moncef Gabbouj. 2012. Rate adaptation for dynamic adaptive streaming over HTTP in content distribution network. *Sig. Proc.: Image Comm. 27*, 4 (2012), 288–311. https://doi.org/10.1016/j. image.2011.10.001

[6] Kevin Spiteri, Rahul Urgaonkar, and Ramesh K. Sitaraman. 2016. BOLA: Nearoptimal bitrate adaptation for online videos. In *35th Annual IEEE International Conference on Computer Communications, INFOCOM 2016, San Francisco, CA, USA, April 10-14, 2016*. IEEE, 1–9. https://doi.org/10.1109/INFOCOM.2016.7524428

[7] S. Ahsan, A. Hourunranta, Igor D.D. Curcio, E.B. Aksu, FriSBE: Adaptive Streaming of Immersive Tiled Video, ACM Packet Video Workshop, 8 June 2020, Istanbul, Turkey.

[8] I. D. D. Curcio, H. Toukomaa, and D. Naik. 2017. 360-Degree Video Streaming and its Subjective Quality. In *SMPTE Annual Technical Conference and Exhibition*. https://doi.org/10.5594/M001758.

[9] I. D. D. Curcio and S. Ahsan. 2020. Viewport Margins for 360-Degree Immersive Video. In *IEEE Multimedia Signal Processing Workshop*. https://doi.org/10.1109/MMSP48831.2020.9287078.

[10] Mehmet N. Akcay, Burak Kara, Saba Ahsan, Ali C. Begen, Igor D.D. Curcio and Emre Aksu, Head-Motion-Aware Viewport Margins for Improving User Experience in Immersive Video, *ACM Multimedia Asia*, 1-3 December 2021, Gold Coast, Australia.

[11] N00072, “Text of ISO/IEC FDIS 23090-2 2nd edition OMAF”, MPEG 132, January 2021

[12] ISO/IEC 23090-2:2021, “Information technology — Coded representation of immersive media — Part 2: Omnidirectional media format”, https://www.iso.org/standard/79881.html

3 Definitions of terms, symbols and abbreviations.

3.1 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

ABR	Adaptive Bit Rate
BOLA	Buffer Occupancy Lyapunov Algorithm
DASH	Dynamic Adaptive Streaming over HTTP
FriSBE	Full Sphere Bit rate Estimation
FS	Full Sphere
GOP	Group Of Pictures
HMA	Head Motion Aware
HMD	Head Mounted Display
MCTS	Motion Constrained Tile Sets
MTHQD	Motion To High Quality Delay
OMAF	Omnidirectional MediA Format
QR	Quality Ranking
VDS	Viewport Dependent Streaming
VR	Virtual Reality

4 Source content material

4.1 Introduction

This clause documents relevant source content material for VR Streaming interoperability.

4.2 Orange test sequence

4.2.1 The VR Experience

The VR 360 sequence is intended to be experienced through a VR headset. The VR spectator is immersed into the stage of a TV news. The real presenter welcomes the spectator and let him on his own 2 minutes before going live. The scenario has been defined to let the VR spectator feel the increasing pressure 2 minutes before the live broadcast as well as perceive the coordination of the technical team both on stage and in the control room in order to make such a well-known program possible.

Figure 1 illustrates the environment of the sequence.

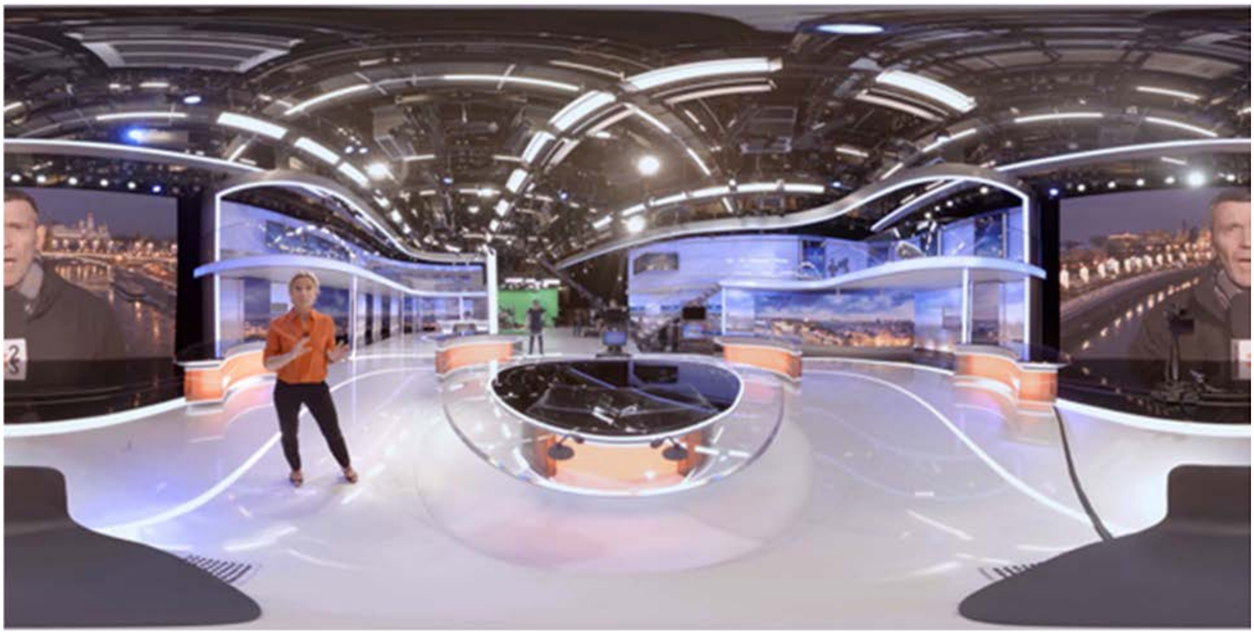


Figure 1: Screenshot of the candidate test sequence

4.2.2 Video capture

Video was shot with a rig of 24 cameras (4 cameras to the top, 4 cameras at the bottom and 16 in a horizonal crown configuration). Each of the cameras had a 2.7K resolution with a 120° angle. Unlike the 8 cameras facing the top and the bottom, the 16 horizontal cameras worked in couples in order to create a stereoscopic effect. The recording was done on SD cards, thus generating 24 files to be synchronized altogether for each shot. Figure 2 below illustrates the camera rig configuration.

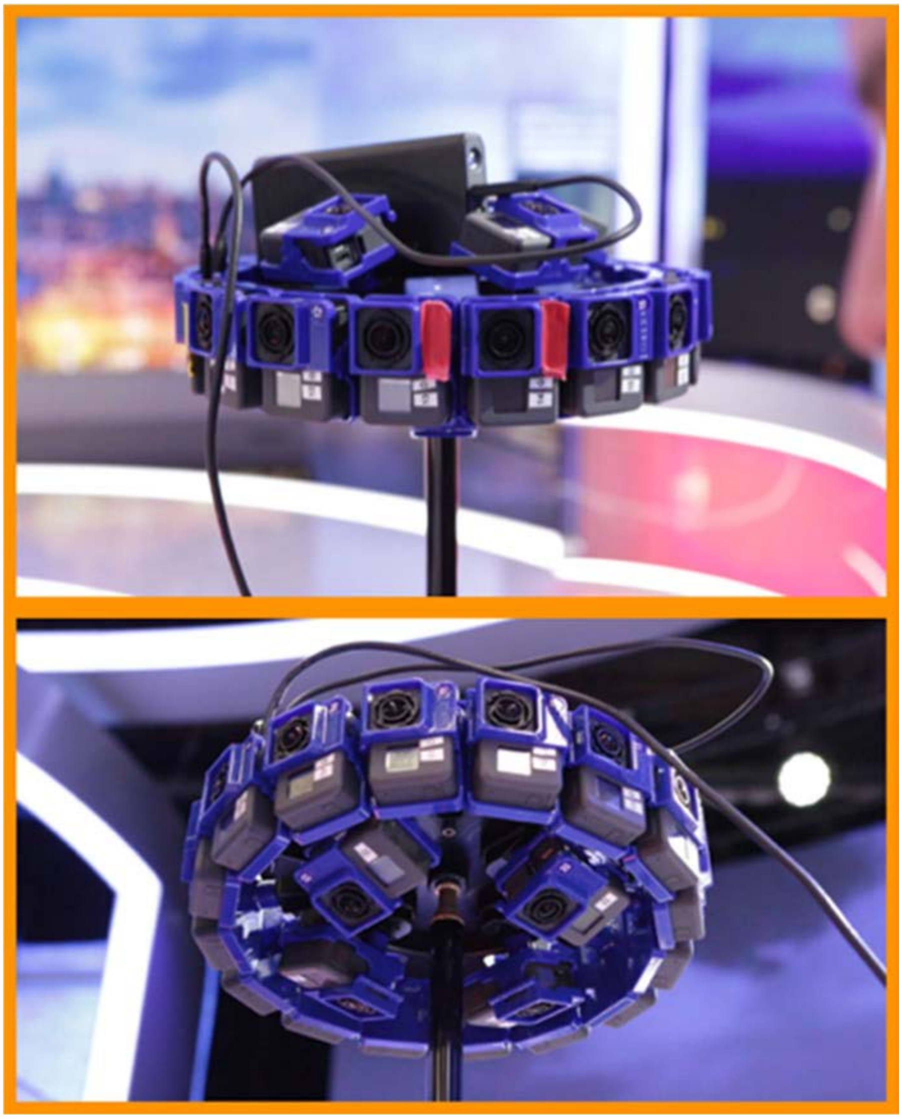


Figure 2: Video shooting configuration

4.2.3 Audio configuration

A 3D microphone made of 32 sensors was placed over the rig. It was linked via RJ45 to an interface delivering through FireWire towards a computer for recording. The audio source in HOA format allowed the recording of ambient sounds perfectly localized in 3D. A few lapel microphones were used as well as one audio signal to simulate the earpiece of the presenter to receive the control room information.

Figure 3 shows the audio capture system.

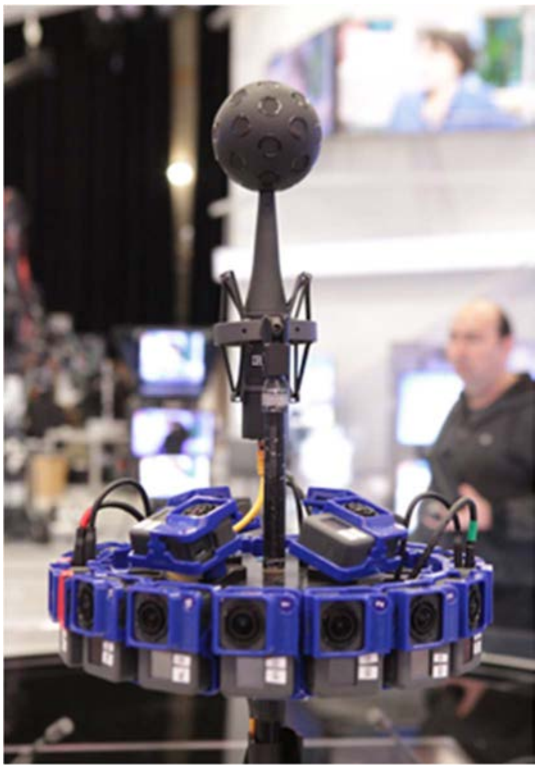


Figure 3: Audio capture configuration

4.2.4 Final product

The resulting version of the sequence is made of:

- Video
  1) 8k resolution
  2) 50 frames per second
  3) Equirectangular projection
  4) 8 bits per pixel, RGB
  5) BT.709 color space
  6) around 2min 10sec. duration
  7) Available in mono and stereo.

- Audio:
  1) French
  2) HOA 3rd order
  3) one stereo track for head lock.

4.3 InterDigital test sequence

4.3.1 The VR Experience

The VR 360 sequences are intended to be experienced through a VR headset. The experience is that of a biker or street walker strolling through a number of attractions and local community in San Diego, California, USA, including Gaslamp Quarter neighbourhood, the harbour, a park, an old trolley, and a local residential community. Figures 4 to 7 provide screenshots from the four full-8K sequences in raw video format.

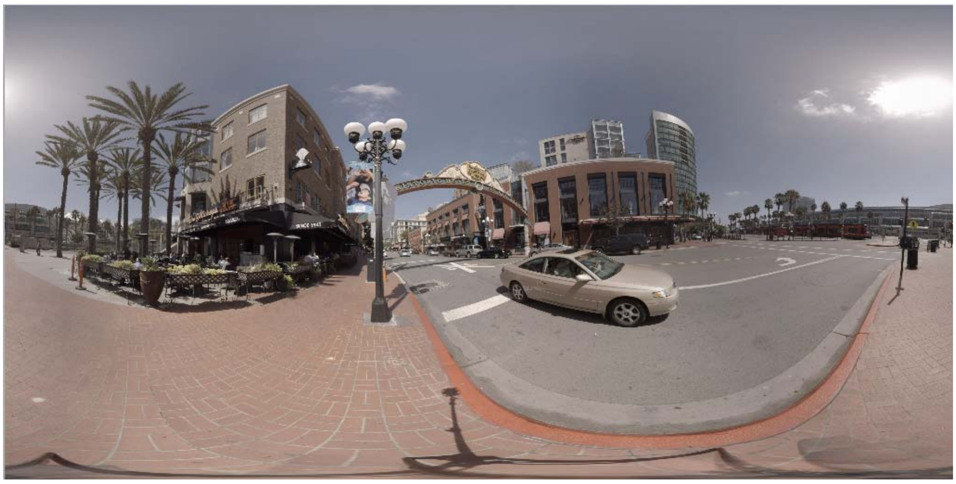


Figure 4: Gaslamp360_8192x4096_30fps_300frames_8bits.yuv

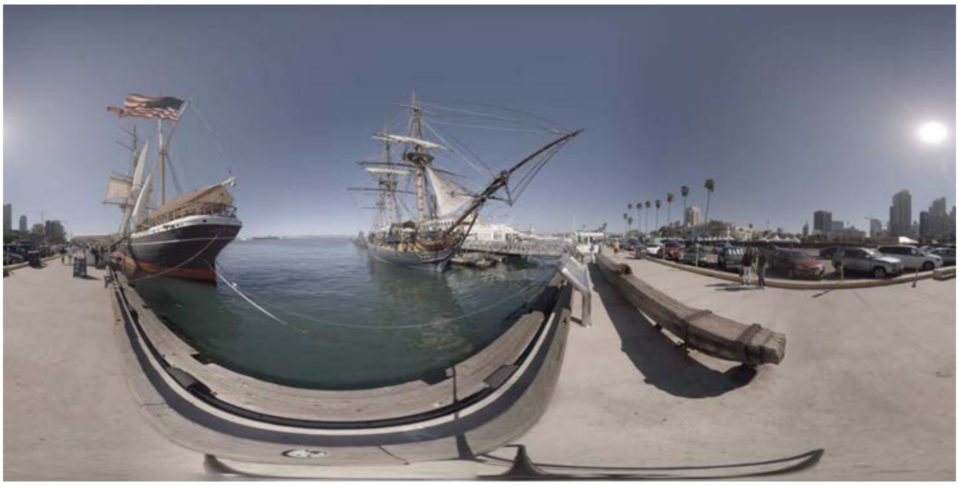


Figure 5: Harbor360_8192x4096_30fps_300frames_8bits.yuv

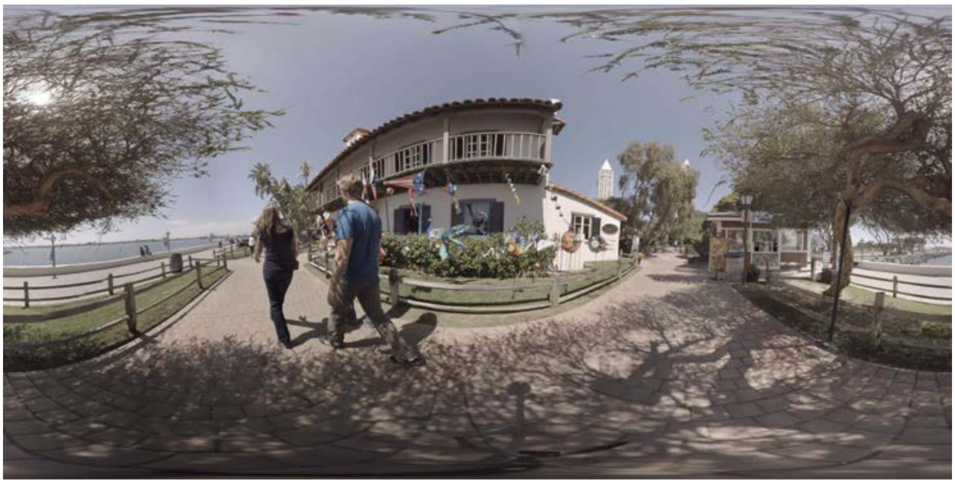


Figure 6: Kiteflite360_8192x4096_30fps_300frames_8bits.yuv

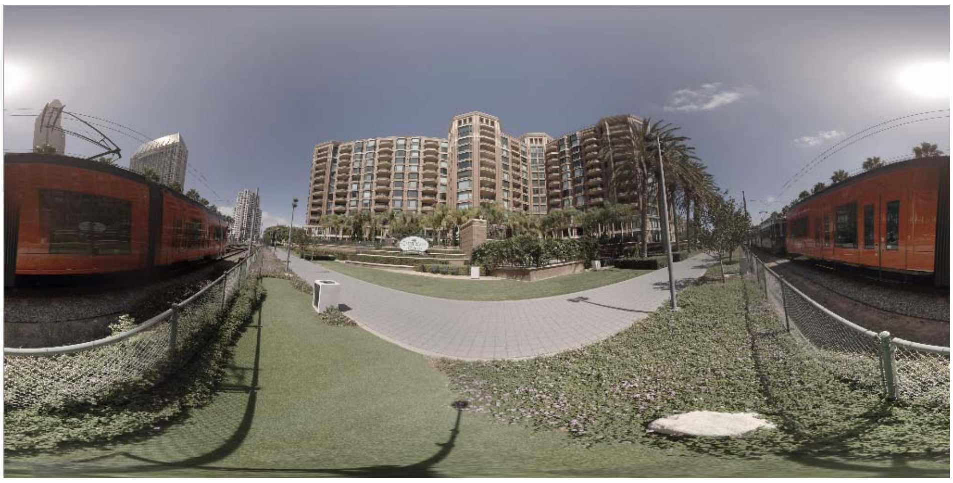


Figure 7: Trolley360_8192x4096_30fps_300frames_8bits.yuv

Figure 8 and Figure 9 provide screenshots from the two 8K (7680x3840) sequences in compressed format.

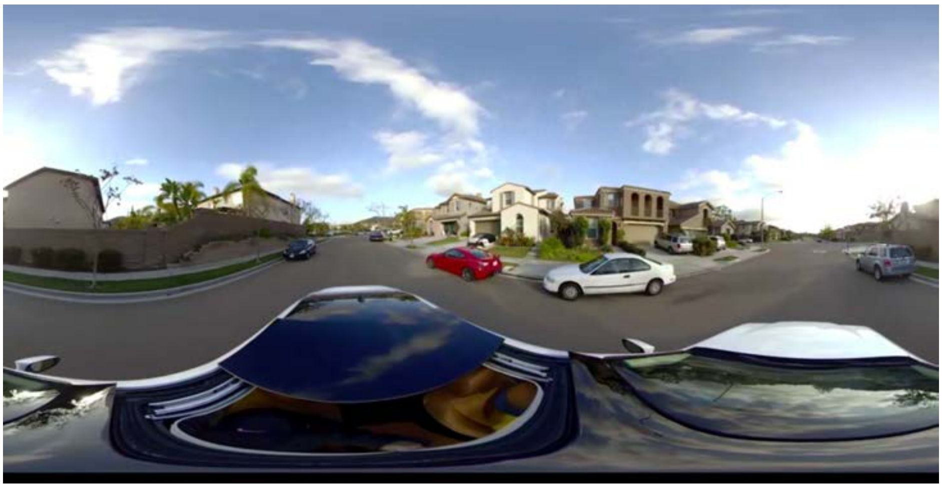


Figure 8: Community_7680x3840_29.97fps_150mbps_5mins.mp4

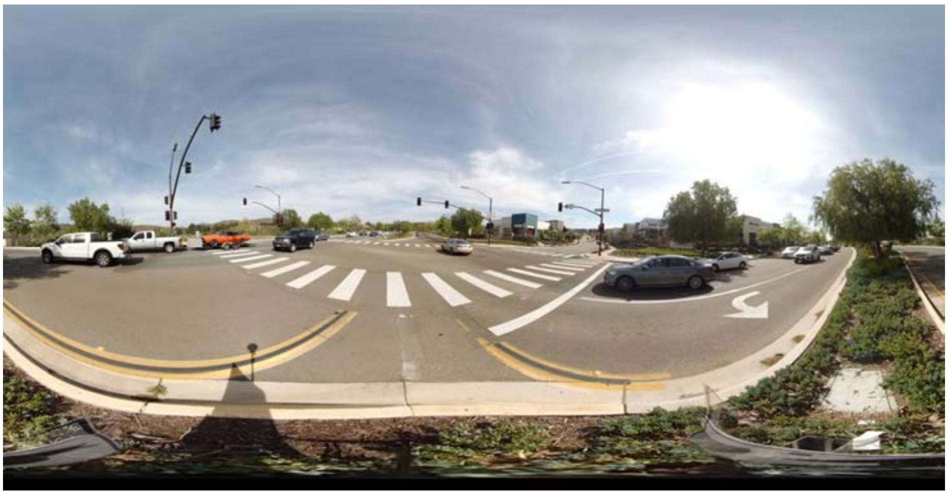


Figure 9: Intersection_7680x3840_30fps_150mbps.mp4

4.3.2 Video Capture

The first four video sequences were captured using a GoPro Omni camera rig, a synchronized camera array with 6 GoPro Hero4 Black cameras, each camera can capture 2.7K resolution at 60fps. Kolor Autopano Video Pro was used to stitch the video captured by the GoPro camera rig. No encoding is done after stitching. Therefore, the stitched content is provided in uncompressed raw YUV format.

The last two video sequences were captured using Insta360™ Pro camera with 6 F2.4 fisheye lenses. The maximum 360 video capture resolution is 7680x3840 at 30fps with post-processing stitching. The MP4 bitstream generated by Insta360™ Pro camera were stitched and encoded by the camera, the compression rate is 150mbps with build-in HEVC encoder.

Figure 10 and Figure 11 illustrate GoPro Omni camera rig and Insta360™ Pro camera used to capture the aforementioned test sequences.

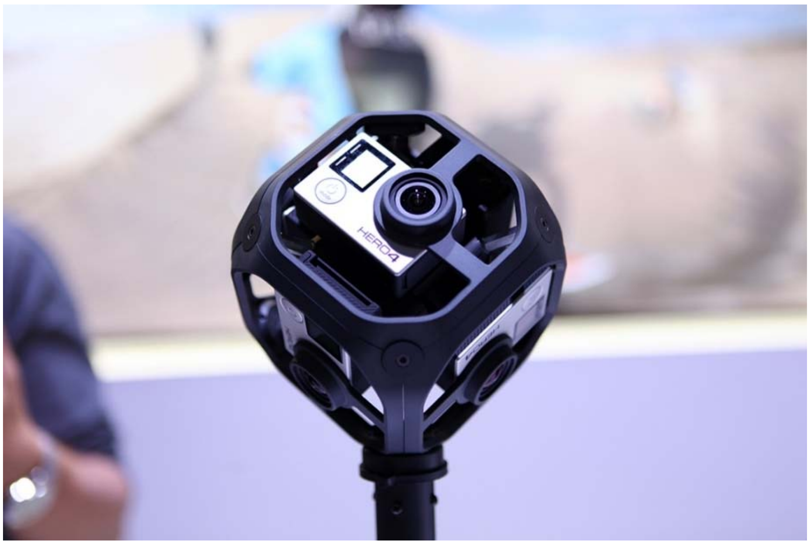


Figure 10: GoPro Video capture configuration

Figure 11: Insta360™ Pro Video Capture

4.3.3 Final Product

The resulting versions of the six sequences have the following characteristics:

- 8K resolutions (8192x4096 or 7680x3840)
- 30 frames or 29.97 frames per second
- Equirectangular projection
- 8 bits per pixel, YUV
- BT.709 color space
- from 10 seconds to 5 minutes

The MD5 checksum of each of the uncompressed 8-bit YUV420 sequences is listed in the below Table 1:

Table 1: MD5 checksum for video sequences

Sequence	**MD5**
Gaslamp360_8192x4096_30fps_300frames_8bits.yuv	858dfe4b7a2d463f1866c82dd14d51be
Harbor360_8192x4096_30fps_300frames_8bits.yuv	aa827fdd01a58d26904d1dbdbd91a105
KiteFlite360_8192x4096_30fps_300frames_8bits.yuv	18c0ea199b143a2952cf5433e8199248
Trolley360_8192x4096_30fps_300frames_8bits.yuv	84d6bfc93053ef28ddfcbe41d0864a9c

The proposed sequences are available for public download at https://www.interdigital.com/visual-technologies#

5 Test results

5.1 Introduction

This clause documents test results for VR streaming for tiled video. Adaptive bit rate algorithms are also used and results for different video bit rates and network configurations are shown.

5.2 ABR Streaming of Tiled Video

The primary challenge of using tiled VDS with ABR arises because the client receives only tile bit rates as part of the DASH manifest. At any time, depending on the viewport orientation, the number and size of the tiles in the viewport may differ, hence drastically changing the required bit rate for a particular viewport and non-viewport quality. This clause describes an ABR solution for tiled 360-degree video streaming [7]. The technique is about Full Sphere Bit rate Estimation (FriSBE) process to estimate the bit rate for each viewport quality level, considering viewport orientation and variation in tile bit rates. FriSBE was implemented in the Nokia OMAF player and tested it with a fetch time based ABR algorithm [5], as well as the Buffer Occupancy Lyapunov Algorithm (BOLA) from dash.js [6]. In this implementation, the algorithms were able to avoid stall events and adapt to varying network conditions.

5.2.1 Design

Traditional ABR algorithms for 2D/flat video have a single bit rate value for the full picture in the DASH manifest, which is used by the algorithms when selecting the appropriate quality for the next segment. The DASH manifest for OMAF tiled videos provides an average bit rate per tile to the player, where the full sphere (FS) is composed of multiple tiles. In order to save bandwidth, VDS is used so that viewport tiles are downloaded at a high quality, whereas tiles that are outside the viewport (consequently not visible to the user) are downloaded at a lower quality. While FS bit rate is simply the sum of bit rates for the tiles, several factors make estimating FS bit rate from the manifest difficult. Firstly, the content complexity and the tile sizes are not always homogenous for a 360-degree video; some tiles may have a much higher bit rate than others due to larger size or the encoder assigning more bits to it due to content complexity. Secondly, the viewport does not always align with tile-boundaries with some orientations requiring larger number of tiles than others. Therefore, the required FS bit rate can change drastically depending on the number of tiles in the viewport, their sizes, and the complexity of the content within those tiles. In Figure 12, the bit rate variation is shown for one of the test sequences using three vertical and 8 horizontal viewport orientations. It shows that slight head movements can create a large difference in the required bit rate values.
The idea for FriSBE is to estimate a set of FS bit rates for different quality levels that provide the client an approximation of the required bit rate at a particular quality regardless of the current viewport. The design consists of the following steps, which are discussed in detail later:

(i) find the representative viewport V
(ii) estimate the required FS bit rates for different qualities based on V, and
(iii) use the calculated FS bit rates from the previous step in the ABR algorithm for network adaptation.

Despite when using HEVC Motion Constrained Tile Sets it is possible to independently decode video tiles (possibly using multiple decoders), the FriSBE method treats all tiles of a single DASH segment collectively based on the assumption that a segment cannot be played until all tiles from that segment are available.

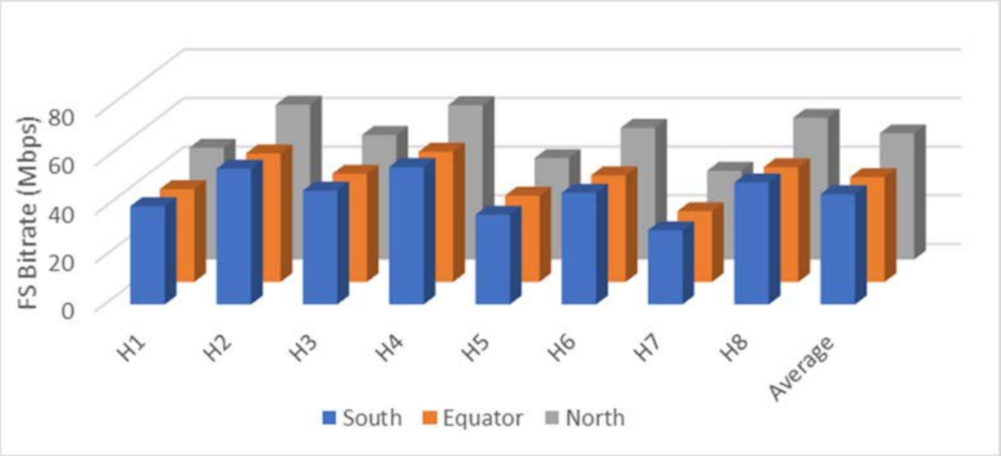


Figure 12: A chart showing variation in FS bit rate values for different viewports: 3 vertically and 8 horizontally adjacent positions. The FS bit rate is calculated using the advertised bit rate for the highest quality for viewport tiles and lowest quality for non-viewport tiles.

5.2.2 Representative Viewport V

As discussed previously, the FS bit rate can vary significantly based on the viewport orientation. Defining the required bit rate for several orientations for all levels is not only complicated, it also does not help in keeping a constant quality level, which is imperative for a good user experience. So, a method is defined for selecting a representative viewport for estimating the required bit rates. First the client identifies a multitude of viewports by moving the head orientation over the video's tile grid in granular steps, both vertically and horizontally. A viewport is selected whenever the tiles change. Once the viewports are identified, the FS bit rate is calculated using a higher quality level for the viewport tiles and a lower quality level for the non-viewport tiles. The highest and the lowest quality were used, respectively, to calculate FS bit rates at this stage. The viewport with the median FS bit rate was chosen as V. While it was found using median to be better than using the initial viewport (azimuth and elevation are 0 for the OMAF spherical coordinates) whether using a different bit rate percentile is more efficient is left as future work.

5.2.3 Full sphere bit rate estimation

To compute FS bit rate we aggregate the required bit rate of all tiles at the quality at which they will be downloaded using the representative viewport. The tiles are divided in two groups; the group of tiles (partially or fully) within the viewport, V, and the group of all remaining tiles, V′. We only modify the quality for V, whereas V′ is always downloaded at the lowest quality. Hence, for N quality levels (and corresponding required bit rates) advertised in the DASH manifest, we create N quality levels (and corresponding required bit rates) for FS where the required FS bit rate has a direct correlation with quality. To minimize the effects of delayed viewport update after head motion, we add a margin area to the actual viewport size of the device and use that as the viewport size. Formally, for a quality level q, the DASH manifest 17edimenns the tile bit rate B_T^q, where T is a tile that is fully or partially within the viewport area V, or it is part of the non-viewport tiles V'. Then the full sphere bit rate B_{FS}^q is:

$$B_{FS}^q = \sum_V B_T^q + \sum_{V'} B_T^0 \qquad (1)$$

where q has N levels, 0 is the lowest and N-1 is the highest.

Note that we use the term viewport/viewport area to imply the region including the margin and not just the device's viewport in the formula and also other sections of this clause 5.2, as both viewport and margin are treated equally (i.e., downloaded at the same quality).

More complex schemes with variable margin sizes or higher quality for all non-viewport tiles can also be used, but require more insight into the role of margins as it may affect the relationship between quality and bit rate (a viewport with a wide margin at a lower quality may require more bit rate than one at high quality with no margin). Further test results and considerations about using viewport margins are available in clause 5.3.

5.2.4 ABR Algorithms

Having a set of FS quality levels and bit rates, the player can now use existing ABR algorithms for adaptation. Some modifications in the computation of the indicators used in ABR algorithms is required to take into account all tiles for each segment. For instance, in this implementation, we define *buffer occupancy* as the number of segments in the buffer for which all tiles have been downloaded, and *throughput* values account for overall throughput for all tile downloads. Once the ABR has chosen the FS quality for the segment, the player uses the current viewport orientation to determine the tiles currently in viewport. Each quality for the FS is mapped to a particular V and V′ quality. The player downloads the viewport tiles at quality for V and non-viewport tiles at quality for V′ (lowest) based on currently chosen FS quality.

To summarize, Figure 13 illustrates the player operations described in this section. The manifest provides a KxN set of bit rates, B_T^q, for K tiles and N qualities. From this set, FriSBE creates a set of N FS bit rates, B_{FS}^q, that each represent V at quality q and V′ at quality 0, which are provided to the ABR algorithm. A mapping for each FS quality and the corresponding V/V′ quality is made available to the video downloader. The ABR algorithm uses the set $\{B_{FS}^q\}_N$ and the ABR indicators collected by the video downloader to choose the quality for the next segment. A further sanity check based on throughput is performed to ensure that the chosen quality by the ABR algorithm is sustainable in the current network conditions. If the sanity check passes, q = q′, otherwise q > q′. Finally, the downloader determines based on the current viewport, the tiles that qualify in V and those in V′, and downloads them according to the available quality mapping.

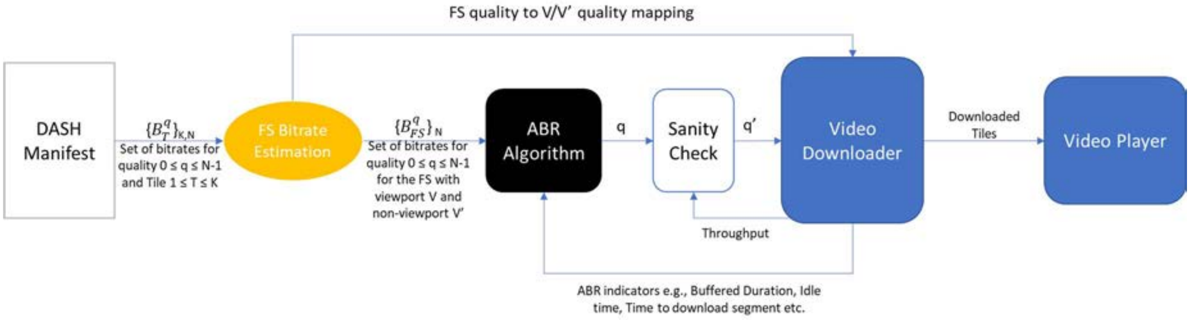

FS quality to V/V' quality mapping
DASH Manifest
$\{B_T^q\}_{K,N}$
Set of bitrates for quality 0 ≤ q ≤ N-1 and Tile 1 ≤ T ≤ K
FS Bitrate Estimation
$\{B_{FS}^q\}_N$
Set of bitrates for quality 0 ≤ q ≤ N-1 for the FS with viewport V and non-viewport V'
ABR Algorithm
q
Sanity Check
q'
Video Downloader
Downloaded Tiles
Video Player
Throughput
ABR indicators e.g., Buffered Duration, Idle time, Time to download segment etc.

Figure 13: Flow diagram of player operation with FriSBE

5.2.5 Implementation

The public sourced Nokia OMAF Player Engine was augmented with the FriSBE ABR technique. Two ABR algorithms were used:

a) the TIME algorithm based on work in [5], and
b) the BOLA algorithm adapted from the DASH reference player [6].

Since, the goal is not to compare adaptation logic, the ABR algorithms are treated as black boxes and their detailed operation is not described. As mentioned previously, the indicators such as throughput and buffer occupancy consider all tiles for each segment. The TIME algorithm uses time to download as an indicator for which we use single tile downloads, but the parameters are adjusted to consider that all tiles must be downloaded within a fraction of their playout time.
For segment download the player uses a parallel segment fetching method as described in [5]. The method maintains multiple HTTP connections at the same time; one HTTP connection for each tile, i.e., one HTTP thread per tile (12 or 24 tiles for our test sequences). The download is not strictly synchronized for segments; however, the HTTP thread of a tile may download up to one segment in the future if the download for a previous segment is still pending for any of the other tiles. For example, if 18ediment *I* is still being downloaded for one or more tiles, the HTTP threads of the other tiles may download segment *i+1*, but not segment *i+2*; the thread must wait for all tiles to finish downloading18edimentt *i* before sending a GET request for segment *i+2*. The simultaneous multiple HTTP connections for tiles that are part of the same segment can create race conditions; this is a limitation left for future work.

Finally, when the viewport changes, the player attempts to download the segments of any new tiles in the viewport at higher quality even when they are already buffered in the lowest quality. If this is done in time for the segment to be played out, the higher quality is rendered, otherwise the already buffered lower quality is rendered. We used a buffer duration of 3 seconds with a pre-buffering threshold of at least 1 second to begin playout. The short buffer was used to minimize bandwidth waste created by re-downloading viewport tiles at higher quality after head motion; the longer the buffer, the higher the number of segments that need to be downloaded again. The viewport size used was 110x110 degrees including margin area (device viewport size was 90x90 degrees).

5.2.6 Experiments

We evaluated the FriSBE based player using three sequences: PoleVault, Harbor Biking and Trolley encoded using Kvazaar. Table 2 summarizes the test sequences. Each sequence was created with two tiling schemes (4x3 and 6x4) using a single resolution, multiple quality scheme described in OMAF Annex D4.2. Smaller tiles were used in the polar regions; approximately 30 degrees high for each pole and about 120 and 60 degrees for the equator for the 4x3 and 6x4 grid respectively. All tiles had the same width. All sequences have a segment size of 566ms: a Group of Pictures (GOP) size of 16 + I frame at 30fps. A short segment size allowed quick viewport update after head motion. The experiments used monitor-based rendering of the viewport and the viewport information was fed using text files for the sake of automation and reporting. However, some basic testing with HMD was conducted to validate the findings and player operation.

Table 2: Test Sequences

Video	Resolution	Bit rate (Mbps)	Tiling Scheme
Trolley	7680x3840	25, 20, 15, 10	4x3, 6x4
HarborBiking	5760x2880	35, 30, 25, 20, 10	4x3, 6x4
PoleVault	3840x2160	20, 15, 10, 4	4x3, 6x4

5.2.6.1 Head Motion

The tests were conducted with

i) no head motion
ii) only horizontal head motion represented with the speed of head in degrees per second (dps)
iii) fast random head motion in all directions, and
iv) a human generated head motion specific to the content.

For the latter, we used a single test subject who explored each of the three sequences, focusing generally on interesting aspects of the video (e.g., reading texts, following the pole vault jumper, watching the approaching train etc.) while also exploring the surrounding at least once (looking towards the poles and behind). The viewport orientation over time for the three user-generated head motion files and the random fast head motion for 60 seconds is shown in Figure 14. The user generated motion has a gap at the back because a tethered HMD was used, and the user remained in the comfort zone where she did not have to readjust the cable. Also note that the points on the figure represent the centre of the viewport; the rectangular viewport can be imagined around it. The fast random head motion reaches maximum speeds of 60dps in the horizontal direction maintained over a few seconds. The human head motion has speeds of over 100dps horizontally. However, they only last for a second or less. Viewport was updated at 500ms intervals.

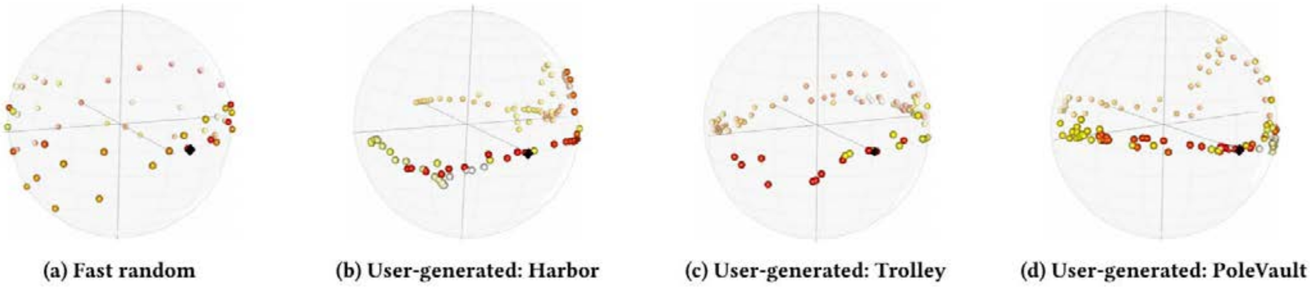


Figure 14: An illustration of the head motion; each point represents the center of viewport at a given time, the colour lightens with the passage of time changing in the order red-orange-yellow-white. The first viewport is marked with a black diamond. Note that the grid on the sphere is shown for clarity

5.2.6.2 Network conditions

The tests were carried out in two phases. The first phase consisted of testing the sequences (duration is approximately 60s for all) under stable network conditions. Here the goal was to evaluate the performance in the presence of head-motion; therefore, the test durations were short and the network conditions were stable. We tested with no head motion, five horizontal head motion with steady speeds (5, 10, 15, 20dps), fast random and human-generated head motion. Bandwidths of 50, 35, 25 and 15 Mbps were used. Each test case was repeated 10 times for statistical significance.

In the second phase, we performed 10 minute long tests by running the sequences in a loop. In this phase we used two different varying network conditions:

i) *SeeSaw*, where the bandwidth cycles between 50Mbps and 15Mbps every 30 seconds, starting at 50Mbps
ii) *Slide*, where the bandwidth starts at 50Mbps and then changes every 30 seconds to 35, 20, 10, 20, 35, 50 and then repeats in that order.

Since the human generated head motion does not terminate at the initial head position, looping it would have resulted in full viewport jumps. Therefore, we used the fast random head motion for this phase of testing. In addition, we also tested for horizontal head motion (15dps) and no head motion. The 10 minute testing took significantly longer to run; hence, the testing was repeated 5 times instead of 10 as in the first phase.

5.2.6.3 Metrics

The performance were evaluated using typical adaptive streaming metrics such as stall events, stall duration, throughput, quality levels and changes. In addition, to include the 360-degree aspect, we introduced the metric for rendered viewport quality. The viewport quality is calculated at the time of rendering using the following formula, where L is the total number of tiles visible in the viewport at a given time:

$$ViewportQuality = \sum_{i=1}^{I} (\mathrm{QR}I \cdot \mathrm{Coverage}\ [\mathrm{i}]) \qquad (2)$$

QR is the Quality Ranking of the tile and Coverage is the percentage of the viewport the tile is covering. The formula is borrowed from 3GPP TS 26.118. Since the ranking follows the OMAF specification, the highest quality has the lowest value. Hence, a low value of *ViewportQuality* indicates a better quality. Note that the ranking was used such that 1 is always the highest quality and the remaining are in uniformly descending order with a decrement of one. Since there are 5 bit rate levels for Harbor Biking (see Table 2), the lowest quality is 5, whereas it is 4 for the other two sequences. With this scheme, if the *ViewportQuality* value is not a whole number or close to a whole number, it can be deduced that the viewport is displaying tiles of two qualities at least.

5.2.7 Results

In this section, we present the results of our experiments, first under stable and then under variable network conditions.

5.2.7.1 Stable Network Conditions

Stable network conditions were used with different levels of head motion to study the effects of a changing viewport on the adaptation algorithm. The results for different metrics follow.

5.2.7.1.1 Stall Events

Stalls were generally not observed for any of the test conditions with steady horizontal head motion even at 20dps. The average stall duration for any sequence for all test cases was less than 17ms for the Time algorithm and below 2ms for BOLA. For the user-generated head motion and random head motion, the TIME algorithm showed some stalling. The total number of stall events was no more than 2, with 1 being more common. The total stall duration for any of the tests ranged from about 25ms to a little over 300ms. BOLA algorithm was more successful in avoiding stalls, with only one 25ms stall experienced for the PoleVault sequence with random fast head motion, too low to impact user experience.

5.2.7.1.2 Quality Variation

The average *ViewportQuality* (see Equation above) observed for the sequences was higher for the 6x4 grid than the 4x3 grid as shown in Figure 15. Furthermore, the 6x4 grid was better at avoiding stall events as well. This is expected, since the tiles are smaller and viewport changes lead to smaller overhead caused by segments download. However, the MCTS tiling implies that smaller tiles have lower encoding efficiency; so this is expected to be considered when analysing the overall benefits.

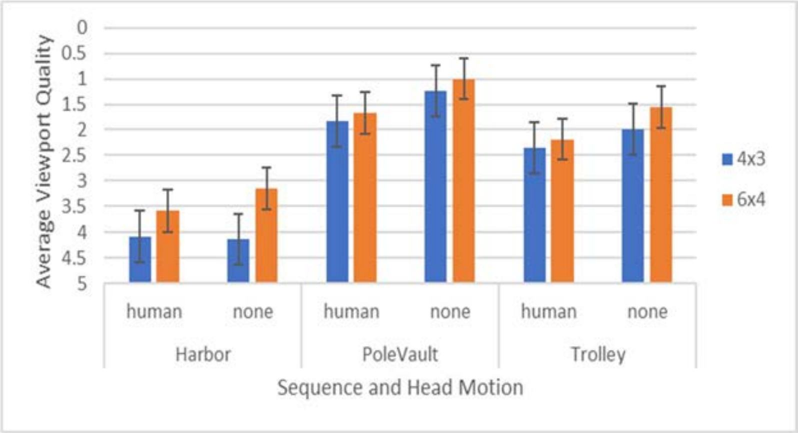
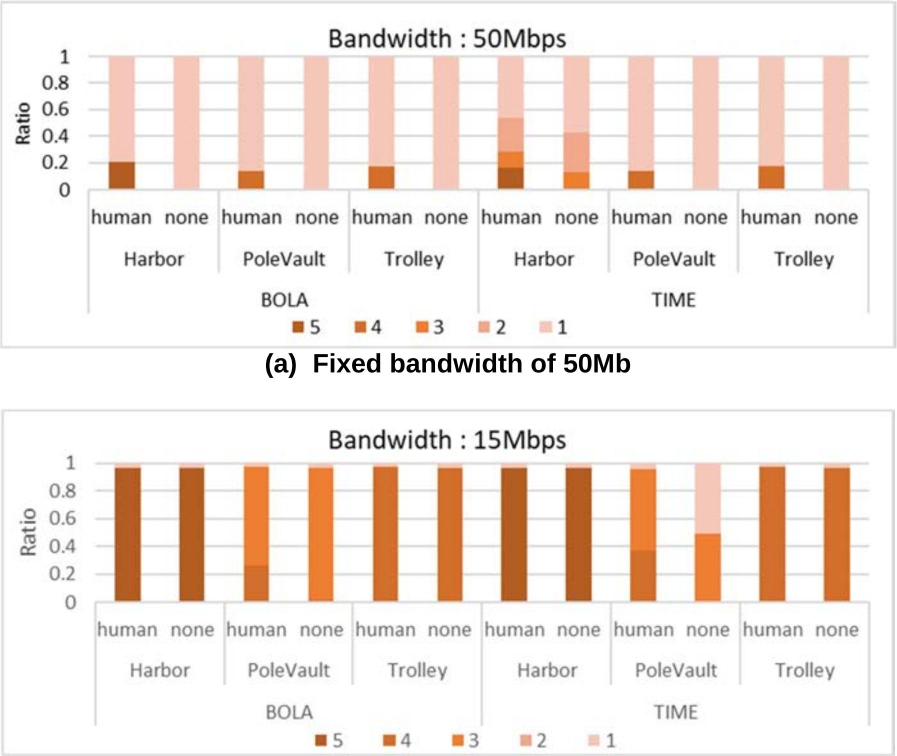


Figure 15: The 6x4 tile grid was able to not only achieve a higher quality viewport, but was able to maintain the quality during head motion better than the 4x3 grid for most cases. Results for human and no head motion (none) are shown.

To estimate the stability of the viewport quality, Figure 16 shows for the entire duration of the video, a stacked bar graph of the ratio of viewport tiles that were rendered at a given quality to all the viewport tiles rendered. For human head motion, the viewport is less stable than any of the horizontal head motion schemes we used. For 50Mbps, the quality was maintained at highest level (1) for most cases. For 15Mbps, each sequence maintains a different quality.



(a) Fixed bandwidth of 50Mb

(ixed bandwidth of 15Mbps

(2022)

Figure 16: Stacked graph showing ratio of the viewport tiles at different quality levels (1-5).

Hence, we used that in the graphs along with no head motion for comparison. At 50Mbps, most sequences remain at the highest level of quality for most of the time. At 15Mbps, Harbor maintains the lowest quality (5) and Trolley maintains the second lowest (4) for the whole sequence. PoleVault has a lower range of required bit rates and has more alteration between QR 3 and 4. Note, that the small share of QR 1 in all 15Mbps cases is because the player always starts at the highest quality before stepping down, leading to higher startup delays (mean: 4.5s).

5.2.7.1.3 Throughput

The average throughput was calculated as the sum of the sizes of the segments downloaded over the total duration of the test. Figure 17 shows the observed throughput for the different sequences under different network bandwidths. The values are indicative of all test conditions: single viewport, horizontal, random and human head motion. We found that the bandwidth utilization factors were not as high as some 2D ABR schemes that can be found in the literature. The reason was threefold: i) the encoding bit rate levels did not always match exactly with the bandwidth, ii) the chosen quality takes into account head motion and the possibility of sudden rise in throughput, and hence is more conservative and iii) potential race conditions caused by the use of multiple HTTP connections for each tile, which can penalize the player at times.

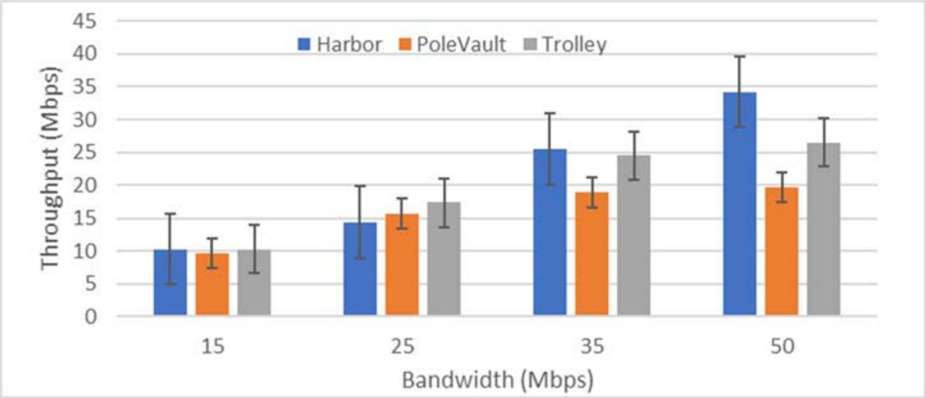


Figure 17: Average throughput for the different sequences and network bandwidth shown here with error bars.

5.2.7.2 Variable Network Conditions

The 10-minute tests show the adaptability of the algorithms to changing network conditions. Of the two configurations we used, *Slide* has one bandwidth level that is too low for two of the sequences we used, and stalls are expected. The other, *SeeSaw*, never falls to a bandwidth that is too low to sustain uninterrupted streaming but is more challenging as it sees sudden large drops in bandwidth.

5.2.7.2.1 Stall Events

We observed no stall events for PoleVault in our testing for variable network conditions. Short stall events, 100-200ms were sometimes observed when bandwidth dropped in *SeeSaw*. For, *Slide*, more stalls were observed because the bandwidth dropped to 10Mbps, which was too low for both Trolley and Harbor at even the lowest quality as can be seen in Figure 18a. The stall event duration per stall was short due to a short buffer duration. The average total stall duration for the entire duration for all test conditions is summarized in Table 3.

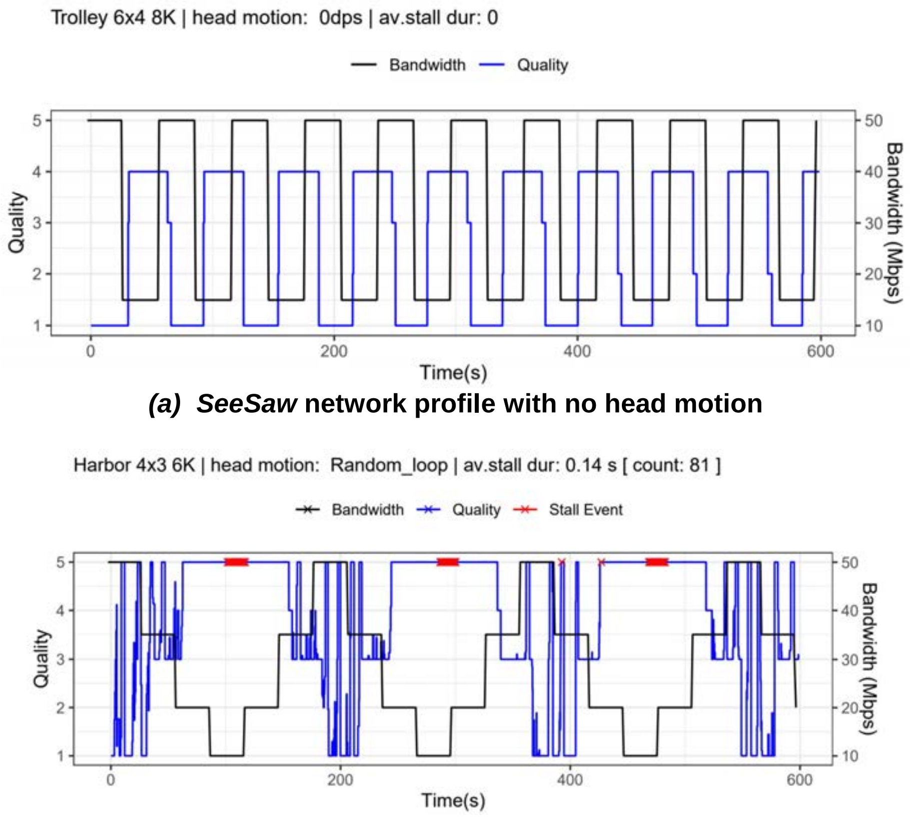
Table 3: Total Stall Duration

	BOLA			TIME		
	Stable	*SeeSaw*	*Slide*	*Stable*	*SeeSaw*	*Slide*
Mean	0.00s	0.55s	7.93s	0.01s	1.47s	8.00s
Std. Dev.	0.00s	0.78s	5.70s	0.05s	1.47s	5.87s

5.2.7.2.2 Adaptability

The implementation was able to adapt to changing network conditions with and without head motion. For reference, we show timeline graphs in Figure 18 for the BOLA algorithm.

The first one is the 10-minute looped Trolley sequence with the *SeeSaw* network profile. Note that Quality 1 is the highest, so the algorithm is switching to highest quality when the bandwidth is 50Mbps and to lowest quality when the bandwidth is 15Mbps; there is no head motion and no stall events for this test, although there were some short stalls for bandwidth drops in some cases. The second graph is for the 10-minute looped Harbor case with *Slide* network profile with random fast head motion. Note that the stalls are mostly in the region where bandwidth is too low for the sequence and the algorithm adapts otherwise. When bandwidth is large, the head motion causes the quality to drop occasionally (calculated based on the Equation above). Note that the graph in Figure 18b shows the most challenging test sequence and test condition.



***(a) SeeSaw* network profile with no head motion**

(*lide* network profile with random head motion

(2022)

Figure 18: A timeline showing the adaptation of quality levels

5.3 Streaming of Tiled Video using Viewport Margins

The use of margins in viewport dependent delivery is a technique to reduce the motion-to-high-quality delay (MTHQD) [8] by decreasing the low-quality ratio in the viewport. In [8], the MTHQD is defined as the delta time from the start of the head motion to the time when all the tiles in the new viewport are rendered in high quality. In [9], the viewport margins are described as an extension toward the background tiles from the current viewport. They can be used as an extension in only one dimension (e.g., horizontal considering the left-right head motion) or in both dimensions (e.g., horizontal and vertical considering the left-right and the top-bottom head motions) by adding extra safety areas around the current viewport.

Reference [9] describes two types of viewport margins: symmetric and directional. In that study, results showed that directional viewport margins performed better than symmetric margins in conversational video transmission. Directional viewport margins can be improved further by adding speed awareness. This clause, introduces *head-motion-aware (HMA) margins* and make a number of important observations with various head motion speeds and tile configurations (i.e., 6x4, 8x6, 12x8). With head-motion-aware margins, it is possible to reduce the average MTHQD by up to 64% [10].

5.3.1 Head Motion Aware (HMA) Margins

With HMA margins, the head motion direction (i.e., motion vector) and its speed are considered for the ABR to decide the set of HMA margin tiles. Figure 19 shows two cases: symmetric margins when the head is stationary, and HMA margins when the head moves slower than a given speed threshold.

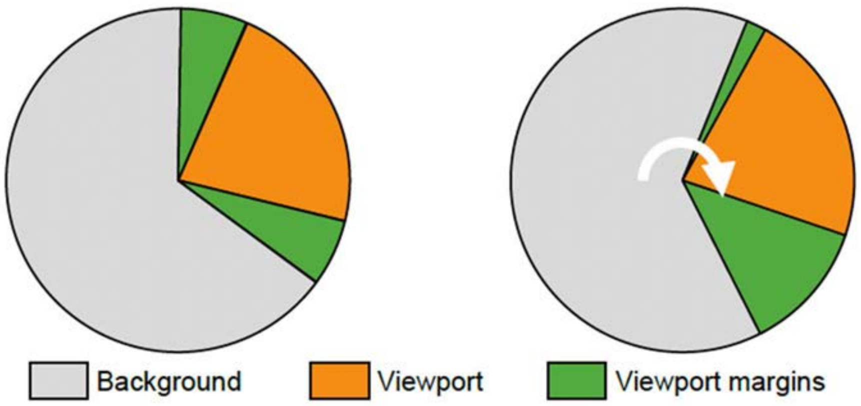


Figure 19: Symmetric (left) and HMS (right) margins

As the background tiles may end up as viewport tiles when there is head motion, the goal is therefore to predict these possible future viewport tiles and mark them as margin tiles. To accomplish this, the motion vector received from the head mounted display (HMD) is decomposed into horizontal and vertical components (i.e., HMD_x and HMD_y , both in degrees for ≈ 33 milliseconds interval) to calculate possible margin tiles.

5.3.2 Experimental Setup

Two 8K video sequences (Harbor and Trolley) were encoded to four quality levels at 20, 30, 40 and 50 Mbps. The mapping technique was the equirectangular projection and each sequence had three (6x4, 8x6 and 12x8) tiling schemes. The segment size was 300 ms and the total duration was 60 seconds. The quality levels were referred to as 1, 2, 3 and 4, where 1 indicated the highest and 4 indicated the lowest quality. Both artificial head motions along with human head motion data were used. The human head motion data was extracted from a user session on the Oculus Quest. The artificial head motions were generated at different speeds (i.e., 25, 45, 60, 90, 120 degrees per second (dps)) and they consisted of the same head motion pattern. Six head motions were in 10-second intervals where there were two head motions in opposite directions for each dimension (i.e., horizontal, vertical, diagonal). Four stable network conditions (i.e., 30, 40, 50 and 60 Mbps) were used. However, results are shown only for 40 and 60 Mbps, since the results for 30 and 40 Mbps, and 50 and 60 Mbps cases were similar. The experiments were conducted in the HMD simulator running on a Windows 10 (64-bit) computer. For statistical significance, each experiment scenario was iterated ten times.

The test results were evaluated by using MTHQD and throughput.

5.3.3 Results

In Figure 20, the MTHQD results can be seen. The numbers for Trolley and Harbor averaged since they have similar trends under same constraints (e.g., bandwidth, tile configurations). For the 6x4 tile configuration, the impact on MTHQD is more eminent, since there were less but bigger tiles and this affects the number of HTTP streams directly. When there are more tiles (e.g., for the 12x8 configuration), the viewport margin fetching logic changes more frequently depending on the head motion speed. As it can be seen from the results, margins can decrease the average MTHQD by up to 64%, which is a significant improvement.

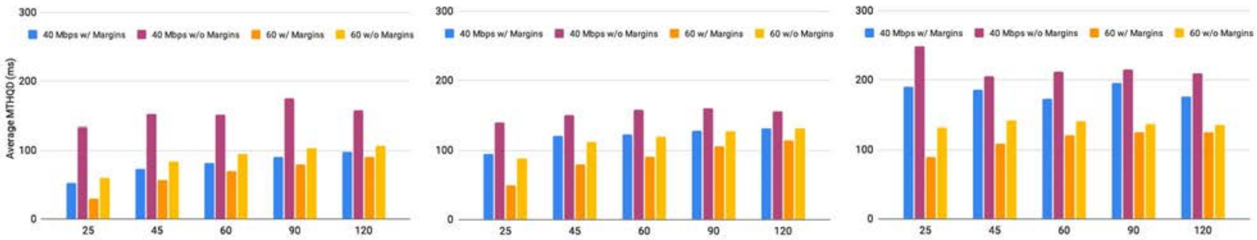


Figure 20: Average MTHQD results with tile configurations of 6x4 (left), 8x6 (center) and 12x8 (right).

At 40 Mbps, even though the best improvement is visible in the 6x4 configuration, the results of the 8x6 configuration are not much affected by the head speed. Regardless of the speed, it can be seen that the 8x6 configuration acts quite stable in terms of the MTHQD. When the head is following a motion pattern with a faster speed, the MTHQD decreases because the hit ratio of the downloaded margins increases. Overall, the MTHQD is always lower with margins compared to the no-margin case at 25 dps (slow) motion. Finally, as Figure 20 shows, margins reduce the average MTHQD for all configurations at 40 Mbps.

At 60 Mbps, the head motion does not play an important role since 60 Mbps is a sufficient bandwidth for our encoded content. In other words, improvement in the MTHQD is stable at all tile configurations. For higher bandwidth profiles, using viewport margins causes the ABR to start downloading possible future segment tiles using the head motion data and increase the performance of rendering viewport with high-quality tiles faster. When the head motion speed increases, it is also observed that the MTHQD gap between the margin and no-margin cases decreases since the downloaded margins are less likely needed for rendering and the possibility of hitting a background tile increases. At 60 Mbps bandwidth, the total MTHQD improvement is higher with slower head motions.

Looking at the 40 Mbps and 60 Mbps results together, we see that using viewport margins in slow head motions always improved the MTHQD. However, at faster head speeds, the possibility of rendering high-quality margin tiles decreases and downloading more high-quality tiles does not yield further improvements.

The experiments were also repeated using human head motion and the results in different tile configurations can be seen in Figure 21. It was observed that HMA margins decreased the MTHQD for all tile configurations and bandwidth conditions. For the 40 Mbps bandwidth, the improvement between the margin and no-margin cases did not change much based on the tile configuration, whereas in the 60 Mbps scenario, the absolute reduction in MTHQD increased as the number of tiles increased. At 60 Mbps bandwidth limit, the percentage MTHQD reduction from the no-margin case to the margin case was measured as 25.6%, 31.2%, 28.2% for the 6x4, 8x6 and 12x8 tile configurations, respectively. It is possible to safely state that a human head can move slow and fast, but a constant head motion speed higher than 90 dps is not realistic for a human. Yet, the tests with artificial speeds help understand the overall performance of the HMA margins with different head motions.

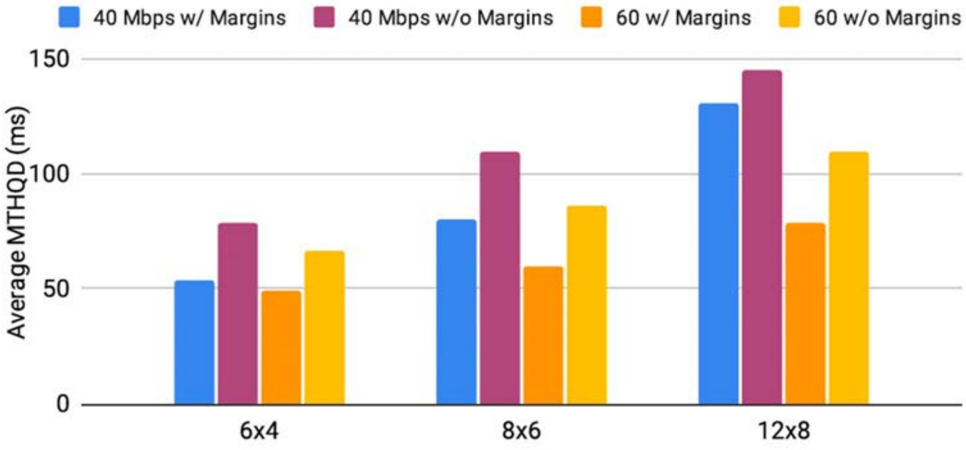


Figure 21: Average MTHQD results for human tests.

Annex A: Process steps for video

Required Processes	**Status Check and Proposal**
Test Sequences with the following parameters: - Basic Video Profile o 4096 × 2048 o BT.709 o 50Hz, 60Hz o 8 bit, 4:2:0 o May be processed to meet lower requirements - Main Video Profile o Mono: 6144 × 3072 o Stereo: 3840 × 1920 o BT.2020, BT.709 o 50Hz, 60Hz o 10 bit, 4:2:0 o May be processed to meet lower requirements - Flexible Video Profile o Mono: 8192 × 4096 o Stereo: 4320x2880 o BT.2020, BT.709 o SDR, HDR o 50Hz, 60Hz, 100Hz, 120Hz o 10 bit, 4:2:0 o May be processed to meet lower requirements Other Requirements: - At least 10 seconds duration - Encoded bitstreams can be published as part of a 3GPP TR	Status Check Interdigital informs on these sequences: o Gaslamp360_8192x4096_30fps_300frames_8bits.yuv o Full 360-degree ERP raw video sequence, resolution 8192x4096, frame rate 30fps, 4:2:0 format, bit depth 8, duration 300 frames, SDR, color space BT.709, no audio o Harbor360_8192x4096_30fps_300frames_8bits.yuv o Full 360-degree ERP raw video sequence, resolution 8192x4096, frame rate 30fps, 4:2:0 format, bit depth 8, duration 300 frames, SDR, color space BT.709, no audio o Kiteflite360_8192x4096_30fps_300frames_8bits.yuv o Full 360-degree ERP raw video sequence, resolution 8192x4096, frame rate 30fps, 4:2:0 format, bit depth 8, duration 300 frames, SDR, color space BT.709, no audio o Trolley360_8192x4096_30fps_300frames_8bits.yuv o Full 360-degree ERP raw video sequence, resolution 8192x4096, frame rate 30fps, 4:2:0 format, bit depth 8, duration 300 frames, SDR, color space BT.709, no audio o Balboa360_6144x3072_60fps_600frames_8bits.yuv o Full 360-degree ERP raw video sequence, resolution 6144x3072, frame rate 60fps, 4:2:0 format, bit depth 8, duration 600 frames, SDR, color space BT.709, no audio o Broadway360_6144x3072_60fps_600frames_8bits.yuv o Full 360-degree ERP raw video sequence, resolution 6144x3072, frame rate 60fps, 4:2:0 format, bit depth 8, duration 600 frames, SDR, color space BT.709, no audio o Community_7680x3840_29.97fps_150mbps_5mins.mp4 o Full 360-degree ERP HEVC video bitstream (150mbps), resolution 7680x3840, frame rate 30fps, 4:2:0 format, bit depth 8, duration 5 mins, SDR, color space BT.709, no audio o Intersection_7680x3840_30fps_150mbps.mp4 o Full 360-degree ERP HEVC video bitstream (150mbps), resolution 7680x3840, frame rate 30fps, 4:2:0 format, bit depth 8, duration 5 mins, SDR, color space BT.709, no audio o All sequences are available at https://www.interdigital.com/video-resources/, and can be further downsampled to 4K resolution or other projection format using JVET 360Lib software.
Content Generation Guidelines	Status check • A few statements in [2].

	Proposal: • Document content generation guidelines that can be used for the development of test material
Test Case Definition for running subjective tests - Needs to be a well-defined subset only To be based on the same content, so 8k is needed. - This is for characterization, not for selection. Hence, comparable and accessible tools are expected to be used such as reference software of MPEG, etc.	Status Check: o Nothing available until now Proposal: o Same content prepared for each media profile with the following parameters: o Simple profile, viewport-independent o Main profile, viewport-independent o Main profile, viewport-dependent ▪ Different viewport-switching latencies o Advanced profile, viewport-dependent ▪ One or two different configurations o Different bitrates, but no dynamic bitrates. o Mono, stereo o All cases must be supported by o Content generation guidelines o conformance bitstreams
Test Material Preparation o Encoding and decoding at different bitrates o Inclusion of relevant metadata o Exact definition of encoding parameters, preferably with a reference software	Status: o Nokia provided the OMAF Creator SW. More information is available in Annex C. Proposal: o Analyze the above SW o Check further work in MPEG VVC and HEVC o Document exactly how the material was prepared for tests.
Subjective Test Run Definition: o Clear definition of subjective tests o Emulate the testing by providing the decoded sequences on high-power PC	Status Check: o We have tests on [2], clause 7.2 Proposal: o Neglect for now. Focus on interop and objective measures. o Expected to be based on already executed tests o Do not include this as part of the study item as necessity, but make it nice to have.
Test Run Execution: o Preferably done in a fair manner o No selection, so less critical	Proposal: o not include this as part of the study item as necessity, but make it nice to have.

o Preferably done by multiple parties	
Objective Measures o Define reasonably good objective measures o Don't claim that they match subjective quality, but permit some amount of comparison	Proposal: o Investigate what MPEG has done for HEVC and VVC o Document the measures o Apply these measures to the subjective tests to identify quality. o Do a second set of tests for flatscreen rendering that applies objective PSNR measures.
Test Case Definition for conformance bitstreams - Preferred to be much richer and cover different aspects - including DASH Preparation - coverage restrictions, etc.	Proposal: o include conformance bitstreams for all tests done above o make this lower priority as it not essential, but permit the documentation
Hosting of Material o Expect huge amount of data o Some information needs to only be maintained temporarily o Conformance streams are expected to be available permanently	Proposal: o Learn from MPEG and DASH-IF and apply the appropriate procedures o Ask DASH-IF to host test material

Annex B: Test Vectors

B.1 Introduction

In the context of this Report, test vectors are provided. As these test vectors are of significant size, they are hosted by DASH-IF (http://www.dashif.org). DASH-IF is always interested to support the industry in interoperability efforts on DASH-related matters. However, note that DASH-IF is not able to provide any service and availability guarantees of such vectors. Finally, test vectors are preferably be provided following the licensing terms of DASH-IF Test assets.

In order to add test vectors, clause B.2 provides some procedures on how to host test vectors on DASH-IF website.

B.2 Uploading and Hosting Test Vectors

The following information is provided by the DASH-IF Test Asset Coordinator. For information on how to contact the test asset coordinator, please communicate with the co-rapporteur of the study item.

1. How hosting/uploading can be done?
    1. The preferred way is that the hosting is done on the CDN server by the DASH-IF Test Asset Coordinator.
    2. Uploading to the test assets database would be done:
        1. Preferably by 3GPP contributors
        2. Or alternatively by the DASH-IF Test Asset Coordinator.
    3. Maintenance would preferably be done by 3GPP Contributors via DASH-IF Test Asset Coordinator assigning the contributors to the github issue or us sending them an e-mail.
2. What information is needed
    1. For hosting, DASH-IF Test Asset Coordinator would need the related MPD files and the media content that each MPD points to be sent to them. They would also need the space required by the content (MPD + media).
    2. For uploading:
        1. Meaningful categorization is expected to be done.
            1. Feature Group (3GPP-VR_CoGui), Feature, Test Case and Test Vector field naming be meaningful.
        2. Preferably by 3GPP Contributors: DASH-IF Test Asset Coordinator would provide 3GPP Contributors with the MPD URLs on the Akamai server. DASH-IF Test Asset Coordinator would create 3GPP Contributors an account on the test assets database. 3GPP would add the test vectors by following ^{t}he 1s^{t} bullet on categorization.
        3. If by DASH-IF Test Asset Coordinator: 3GPP Contributors provide DASH-IF Test Asset Coordinator with the categorization in an excel sheet by following ^{t}he 1s^{t} bullet. DASH-IF Test Asset Coordinator add the test vectors.
    3. For maintenance, DASH-IF Test Asset Coordinator needs to gather the issues with the test vectors and point 3GPP Contributors to the issue via e-mail or assign them the issue if they have a github account. For this, DASH-IF Test Asset Coordinator would need an e-mail address or a github user account name.
3. What is the license that is commonly used

    1. Creative Commons Attrib–tion –-NonCommercial-NoDerivatives 4.0 International license

4. Any other information on the test data base.

    1. None

Annex C: MPEG OMAF 2nd Edition Nokia public source code

C.1 Introduction

In 2018, Nokia announced the public availability of source code for the ISO/IEC JTC1/SC29/WG11(MPEG) OMAF compliant creator and player implementations in GitHub.

This contribution announces the availability of publicly available source code for OMAF 2nd edition (ISO/IEC 23090-2:2021 [11][12]). The source code is publicly available at GitHub (https://github.com/nokiatech/omaf) for research, evaluation and standardization purposes.

C.2 New features in Nokia OMAF public release

The following OMAF 2nd edition features has been made available in the next major Nokia OMAF public release update on 11.10.2021:

- **OMAF Creator v3.0:**

In addition to the already available OMAF 1st edition features for OMAF files and DASH streams (including MPD generation), the following features are also provided:

  - **Support for multiple viewpoints** as defined in subclause 7.12 of [11]
    - viewpoint position structures (7.12.1.2), (7.12.1.3, 7.12.14, 7.12.1.5)
    - viewpoint group structure (7.12.1.6),
    - viewpoint switching list structure (7.12.1.7),
    - viewport looping structure (7.12.1.8),
    - viewpoint entity groups (7.12.2),
    - dynamic viewpoint information (7.12.3.1)
    - initial viewpoint (7.12.3.2),
  - **Support for overlays**: viewport-relative, sphere-relative, as specified in subclause 7.14 of [11]
    - viewport-relative overlays (7.14.3.2),
    - overlay with 2D source video and omnidirectional overlay with 360 mono video sources, in different distances, layer orders and opacities (7.14.2)(7.14.3.3)(7.14.3.4) (7.14.3.7)(7.14.3.8) (7.14.3.11)(7.14.3.13),
    - source region for overlay (7.14.3.6),
    - user interactions enabled with dynamically changing parameters (7.14.3.9)
    - overlays with activation regions (7.14.3.12)
    - overlay configuration information (7.14.4),
    - overlay timed metadata track (7.14.6),
    - grouping of overlays that are alternatives for switching (7.14.7.1),
  - **OMAF Tiling Video Profiles**:

        - Simple tiling OMAF video profile (10.1.7) with support for tile data segments and index segments as specified in Annex B.1.4

Since the OMAF Creator performs only HEVC bitstream rewriting instead of full video transcoding, the input data must be encoded with appropriate tiling and resolution variants. A step-by-step guide is provided in the Nokia OMAF public release GitHub site.

- **OMAF Player v3.0:**
    - Support for parsing and playback of the new features as defined for OMAF Creator above.
    - overlay as an item (7.14.5).

Annex D (informative): Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2019-10	SA4#106	S4-191285				First version. Added agreed content from S4-191139 and S4-191287	0.1.0
2019-10	SA4#106	S4-191294				Updates to section 6.2 (Orange test sequence)	0.2.0
2020-01	SA4#107	S4-200206				Added section 6.3 (InterDigital test sequence)	0.3.0
2020-01	SA4#107	S4-200208				Added Annex B on Test Vectors	0.3.0
2021-06	SA4#114	S4-210854				Added section 9.2 on ABR Streaming of tiled video	0.4.0
2021-11	SA4#116	S4-211521				Added section 9.3 on Streaming with viewport margins	0.5.0
2021-11	SA4#116	S4-211563				Added Annex C and Editorial clean-up	0.6.0
2022-02	SA4#117-e	S4-220160				Final editorial clean-up	0.7.0
2022-03	SA#95-e	SP-220252				For approval in SA	2.0.0
2022-03	SA#95-e					Under Change control	17.0.0
2024-03	-	-	-	-	-	Update to Rel-18 version (MCC)	**18.0.0**

History

Document history		
V18.0.0	May 2024	Publication