**Source: Apple Inc. (Rapporteur)**

**Title: VOPS Permanent Document v0.0.1**

**Version: 0.0.1**

**Agenda Item: 9.5**

**Document for: Information**

**Work Item information:**

|  |  |
| --- | --- |
| **Work Item Name** | Video Operating Points - Harmonization and Stereo MV-HEVC |
| **Acronym** | VOPS |
| **Unique ID** | 1030002 |
| **Target release** | 19 |

**Source details:**

|  |  |  |
| --- | --- | --- |
| **Rapporteur** | **Name** | Waqar Zia |
| **Company** | Apple Inc. |
| **Email** | waqar\_zia@apple.com |
| **Editor** | **Name** | Emmanuel Thomas |
| **Company** | Xiaomi |
| **Email** | thomase@xiaomi.com |

**Revision history:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Meeting** | **Subject/Comment** |
| 0.0.1 | 2025-05-20 | SA4#132 | * Added   + Conformance framework in clause 3.1.   + Conformance content in clause 4.1.1, 4.1.2, 4.1.3. |

# Contents

[Contents 2](#_Toc198654556)

[1 Introduction 3](#_Toc198654557)

[2 Definitions, symbols and abbreviations 3](#_Toc198654558)

[2.1 Definitions 3](#_Toc198654559)

[2.2 Abbreviations 3](#_Toc198654560)

[3 Conformance Requirements 3](#_Toc198654561)

[3 Conformance Framework 3](#_Toc198654562)

[3.1 Sample bitstream platform overview 3](#_Toc198654563)

[3.1.1 Suggested overall architecture 3](#_Toc198654564)

[3.1.2 Thoughts on implementations 4](#_Toc198654565)

[4 Conformance Material 5](#_Toc198654566)

[4.1 HEVC Conformance Material 5](#_Toc198654567)

[4.1.1 Source content 5](#_Toc198654568)

[4.1.2 Compressed Bitstreams 5](#_Toc198654569)

[4.1.3 Reference Software 7](#_Toc198654570)

[11 References 7](#_Toc198654571)

[Annex A – 8](#_Toc198654572)

# Introduction

During SA4#132 it was considered that it is necessary to gather all material related to conformance provided to SA4 meetings. This permanent document is created to serve this purpose.

# 2 Definitions, symbols and abbreviations

## 2.1 Definitions

## 2.2 Abbreviations

# 3 Conformance Requirements

This subclause gathers requirements for conformance testing. This includes features that need to be covered by the conformance.

# 3 Conformance Framework

This subclause gathers documentation on a potential framework for conformance, e.g. tools to tabulate, gather, and present conformance content.

## 3.1 Sample bitstream platform overview

### 3.1.1 Suggested overall architecture

At high level, the sample bitstream platform is composed of three main components that are:

1. A database which contains the description of the available sample bitstreams
2. One or more servers to host the submitted bitstreams
3. A portal for external users to search through and download the sample bitstreams.

In addition, a bitstream validator validates prior to upload that the submitted bitstream does comply with the constraints defined in TS 26.265. To harmonize the bitstream description in the database, the bitstream validator would also generate a bitstream description file for the bitstream database.

A black screen with white text

Description automatically generated

Figure - Overall view of the sample bitstream platorm

### 3.1.2 Possible approaches

#### 3.1.2.1 Approach #1: Extending existing validation tools from the DASH-IF conformance suite

The DASH-IF Conformance test suite [1] is a collection of tools to validate DASH MPD and segments. For segments, the ISOSegmentValidator parses ISOBMFF segments and validates their structures. Since NAL units can be present in decoder configuration box, this tool has some parsing functionality of NAL units. However, it does not implement video bitstream validation.

Other open-source tools such as GPAC and its MP4Box tool have the capability to parse NAL units and optionally to dump to an XML format the content of a video bitstream.

##### 3.1.2.2 Approach #2: Developing XML-based validation as part of the SA4 specification work

In this approach, a specification which defines bitstream constraints would be developped along with an XML schema specifying those normative rules. This XML Schema would for instance be developped for each Operation Point in VOPS.

As an example, a basic Python has been developped to produce the following XML document in Figure 2. This XML example relies on the MPEG-B BSDL standard which defines built-in types for bitstream description, e.g. fields of 1 to 32 bits, undefined payload in the form of a byte range, variable-length encoded field, etc.

|  |
| --- |
| A screenshot of a computer program  AI-generated content may be incorrect.  […]  A screenshot of a computer code  AI-generated content may be incorrect. |

Figure 2 - HEVC XML bitstream description

Once generated, a further tool can validate this HEVC XML description document against a given XML schema specifying the conformance rules. Figure 3 provides such an example of such XML schema. If more complex validation rules, e.g. like those for DASH MPD, are needed, this approach would also require the definition of a schematron which provides higher-level validation capabilities than the XML schema.

A computer code with many text

AI-generated content may be incorrect.

Figure 3 - HEVC XML schema using BSDL built-in elements and custom elements

### 3.1.2 Thoughts on implementations

#### 3.1.2.1 The bitstream validator

The bitstream validator shall:

1. validate that the bitstream is a compliant bitstream according to the corresponding video coding specification and profile.
2. validate that the bitstream is compliant with the bitstream constraints defined in TS 26.265.

For the point 1), the reference video decoder developed by JVET typically have conformance checks embedded. As a result, one way to achieve this validation is to execute the reference decoder with no picture output and verify the absence of error.

For the point 2), we need a way to programmatically express the constraints defined in TS 26.265 to validate the input bitstream. One way would be to extend the reference decoder to add the additional constraints. However, this would require SA4 delegates to learn and modify the reference decoder software project and write those rules directly in the decoder. Alternatively, the input bitstream could be first parsed and based on this parsing, the tool could generate a text dump of the bitstream information and structure (e.g. XML, JSON, etc…). Then, the set of VOPS constraints could be expressed against this text representation of the bitstream. Note that there has been in the past several MPEG standard developed for this purpose such the MPEG-21 Binary Syntax Description Language (BSDL) which could be potentially reused. This approach has the advantage of i) being codec agnostic, i.e. the constraint document could be used for any codec, ii) writing constraints does not require to be familiar with language programming, and iii) the bitstream description generated for the validation can be reused for the database, possibly filtered to keep the relevant information.

#### 3.1.2.2 The bitstream database

The bitstream database should contain the description of each available bitstream. In addition, each bitstream would be linked to a TS number and possibly a profile defined within this TS (using URNs defined to identify each profile).

Since using JSON (or other markup text file) for this purpose seems advantageous, the bitstream database could simply be a git repository on any web-base git service platform (e.g. GitHub, GitLab, etc…)

#### 3.1.2.3 The bitstream hosting server

For storing the bitstreams, any http server could be used. Using git LFS would add a finer control with versioning functions but also comes with added constraints for space and effort to set-up. This does not seem necessary to use git LFS at this point.

#### 3.1.2.4 The public portal

Ideally, the external users would be able to search for TS number, profiles and retrieve the corresponding bitstream. Additionally, it should also be possible to browse the database by TS number, codec, etc…

In a first version, the home page of the git repository (e.g. GitHub repository’s home page) could be sufficient. It times allows a simple static web frontend could also be developed. The web frontend would retrieve the information about the available bitstream from the database and present them to the user. Using static website would also have the advantage to maintain them on the same web-base git service provider, i.e. GitHub. GitLab, etc…

# 4 Conformance Material

This subclause gathers all materials e.g. source content, encoding, decoding tools and conformance software.

## 4.1 HEVC Conformance Material

### 4.1.1 Source content

#### 4.1.1.1. Polytech Nantes database

Uncompressed stereoscopic 3D video content dataset which can be used to, for example, generate MV-HEVC conformance streams by Polytech Nantes was pointed to in SA4 contribution [1]. At the time of writing this paper, all the links provided were down. This has already been observed in the past. The content itself, composed of 31 sequences, 1920 x 1080, 10-bit 4:2:2 YUV at 25 fps, was made freely available via the Creative Commons Attribution-ShareAlike License, and hence suitable for 3GPP SA4 purposes.

At the time being the proposal is:

1. Rechecking if this resource gets back online
2. Checking with the source
3. Check if this can be moved to a more stable home.

### 4.1.2 Compressed Bitstreams

#### 4.1.2.1 JVET1

A set of conformance bitstreams (elementary streams) for testing Multiview Extended 10 and Multiview Main 10 profile are provided by JVET [2]. The streams themselves are available for download at [3]. The various parameters of the bitstream including the profile to tested, resolution of content, frame rate and number of frames are tabulated below.

|  |  |  |  |  |  | Base layer profile | |  | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Categories | Subcategory | Conformance Profile | Bitstream | File name | Resolution | Main | Main 10 | Level | Frame rate (Frames/sec) | | # of frames |
| Prediction Structure | Inter-view prediction | Multiview Extended 10 profile | MVHEVCS-J | MVHEVCS\_J\_APPLE\_2 | 1024x768 |  | X | 4 | 30 | | 300 |
| (2-view) | Inter-view prediction | Multiview Extended 10 profile | MVHEVCS-K | MVHEVCS\_K\_APPLE\_2 | 1024x768 |  | X | 4 | 30 | | 300 |
|  | Inter-view prediction | Multiview Extended profile | MVHEVCS-L | MVHEVCS\_L\_Bytedance\_2 | 1024x768 | X |  | 4 | 30 | | 300 |
|  | Inter-view prediction | Multiview Extended profile | MVHEVCS-M | MVHEVCS\_M\_Bytedance\_2 | 1024x768 | X |  | 4 | 30 | | 300 |
|  | Inter-view prediction | Multiview Main 10 profile | MVHEVCS-N | MVHEVCS\_N\_APPLE\_1 | 1024x768 |  | X | 4 | 30 | | 300 |
|  | Inter-view prediction | Multiview Main 10 profile | MVHEVCS-O | MVHEVCS\_O\_APPLE\_1 | 1920x1088 |  | X | 4 | 30 | | 300 |

#### 4.1.2.2 MV-HEVC stream

An MV-HEVC UHD streaming example is available at [4]. This stream playable only on specific devices listed on the page.

### 4.1.3 Reference Software

1. Reference software for non-multiview content is HM reference software for HEVC [5]
2. For multi-view content the reference software is: HEVC multi-view (MV-HEVC) and 3D video coding (3D-HEVC) extensions HTM reference software [6]

# 11 References

[1] S4-231297 [FS\_HEVC\_Profiles] Discussion source MV-HEVC content, Goteburg, August 2023

[2] Conformance testing for HEVC multiview extended and monochrome profiles, JVET-AM1008

[3] [Online 5.2025]: <https://www.itu.int/wftp3/av-arch/jvet-site/bitstream_exchange/HEVCMultiview/under_test/>

[4] Streaming Examples [Online 5.2025]: <https://developer.apple.com/streaming/examples/>

[5] HEVC HM reference software [Online 5.2025]: <https://vcgit.hhi.fraunhofer.de/jvet/HM>

[6] HEVC multi-view (MV-HEVC) and 3D video coding (3D-HEVC) extensions HTM reference software [Online 5.2025]: <https://vcgit.hhi.fraunhofer.de/jvet/HTM>

# Annex A –