**3GPP TSG SA WG4#119-e S4-220738**

**E-meeting, 11th- 20th May 2022**

|  |
| --- |
| *CR-Form-v12.0* |
| **PSEUDO CHANGE REQUEST** |
|  |
|  | **26**.**955** | **CR** |  | **rev** | **-** | **Current version:** | **1.6.0** |  |
|  |
| *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 |  | Radio Access Network |  | Core Network |  |

|  |
| --- |
|  |
| ***Title:***  | **[FS\_5G\_Video] Proposed Editorial Updates to Clause 4 and 5** |
|  |  |
| ***Source to WG:*** | Tencent |
| ***Source to TSG:*** |  |
|  |  |
| ***Work item code:*** | FS\_5GVideo |  | ***Date:*** | 05/05/2022 |
|  |  |  |  |  |
| ***Category:*** |  |  | ***Release:*** | 17  |
|  | *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-12 (Release 12)**Rel-13 (Release 13)Rel-14 (Release 14)Rel-15 (Release 15)Rel-16 (Release 16)* |
|  |  |
| ***Reason for change:*** | * Proposed editorial updates to Clause 4
 |
|  |  |
| ***Summary of change:*** |  |
|  |  |
| ***Consequences if not approved:*** |  |
|  |  |
| ***Clauses affected:*** |  |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  |  |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  |  |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  |  |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** |  |

**===== CHANGE =====**

# 4 Overview of video codec capabilities in 3GPP services in Release 16

## 4.1 Introduction

This clause summarizes the video coding capabilities in relevant existing 5G services as of Release-16. Video codecs introduced in this clause are referred to as 3GPP video codecs.

In Release 17, two video codecs are prominently referenced and available, namely H.264 (AVC) [7] and H.265 (HEVC) [8].

Both codecs are defined as part of the TV Video Profiles in 3GPP TS 26.116 [3] and are also the foundation of the VR Video Streaming Profiles in 3GPP TS 26.118 [4]. The highest defined profile/level combinations are:

- H.264 (AVC) Progressive High Profile Level 5.1 [7] with the following additional restrictions and requirements:

o the maximum VCL Bit Rate is constrained to be 120Mbps with cpbBrVclFactor and cpbBrNalFactor being fixed to be 1250 and 1500, respectively.

o the bitstream does not contain more than 10 slices per picture

- H.265 (HEVC) Main 10 Profile Main Tier Level 5.1 [8] without any restrictions

Both codecs are also defined for other 3GPP-based services. More details on the codec capabilities and the necessary interoperability requirements for different services are collected in the remainder of this clause.

**===== CHANGE =====**

## 4.3 VR Video Profiles

The VR profiles for streaming applications defined in TS 26.118 [4] address the coded representation of 360 VR distribution signals. Table 4.3-1 provides an overview of the 360 VR relevant formats considered in the context of 3GPP VR profiles. The VR profiles follow the same logic as the TV Video profiles, they represent a list of interoperability points that are amended by the 3GPP services such as 5G Media Streaming (5GMS) described in the following clause.

**===== CHANGE =====**

## 5.2 Reference Sequences

This document provides reference sequences that are used to generate anchors and are also made available in order to generate test bitstreams for other codecs. Reference sequences are selected to be representative for a scenario.

Reference sequences are described in Annex C of this document along with their properties and their licenses. A format for raw reference sequences based on a JSON schema is defined in clause B.2.

Annex D describes how to upload new proposed reference sequences and how to download the reference sequences.

**===== CHANGE =====**

## 5.3 Anchors

Anchors provide a baseline that a tested method can be compared against. Anchors defined in this specification use a codec/profile/level that exists in an existing 3GPP specification as introduced in clause 4.

Anchor tuples are collected to address different qualities and bitrates that can then be used for evaluation over a larger set of operation points.

The following principles apply to anchor definitions:

[…]

**===== CHANGE =====**

## 5.4 Reference Software Tools

Anchors bitstreams for each anchor codec are generated with their corresponding reference software. Reference software and reference configurations are documented in each scenario under clause 6 to permit repeatability of the anchor generation.

For H.264/AVC generated anchor bitstreams, H.264/AVC reference software (AVC Joint Model) has been used.

AVC reference software implementing H.264 (AVC) High Profile called JM and its versions can be downloaded from in the repository http://iphome.hhi.de/suehring/tml/download/.

For H.265/HEVC generated anchor bitstreams, H.265/MPEG-H HEVC reference software (HEVC Test Model) has been used.

HEVC reference software implementing H.265 (HEVC) Main Profile and H.265 (HEVC) Main 10 Profile called HM and its versions can be downloaded from in the repository <https://vcgit.hhi.fraunhofer.de/jct-vc/HM-/tags/>.

HEVC reference software implementing H.265 (HEVC) Screen-Extended Main Profile is called SCM (Screen Content Coding Model) software and is available from [https://vcgit.hhi.fraunhofer.de/jct-vc/HM/-/tags/HM-16.21+SCM-8.8](https://vcgit.hhi.fraunhofer.de/jct-vc/HM/-/tags/HM-16.21%2BSCM-8.8).

Test bitstreams for tested codecs are generated with their corresponding reference software. Test reference software and configurations are documented in clause 8.

**===== CHANGE =====**

### 5.5.1 General

Each anchor bitstream gets assigned multiple performance metrics, in particular:

- the bitrate as defined in clause 5.5.2,

- If Standard Dynamic Range (SDR) is used, then the quality metrics in clause 5.5.4 apply.

- If High Dynamic Range (HDR) is used, then the quality metrics in clause 5.5.5 apply.

An overview of the metrics is provided below. These metrics are implemented in software scripts defined in Annex E. This software is used to compute and report all the metrics. The detailed configuration for the software is provided in clause 5.5.7 for SDR metrics and clause 5.5.8 for HDR metrics.

A detailed reporting schema for metrics is provided in clause 5.5.6.

Metrics for test streams are expected to follow the same principles.

Subjective evaluation of anchors and test streams is not considered in this report.

**===== CHANGE =====**

#### 5.5.4.1 Overview

Based on the introduction in clause 5.5.3, for standard dynamic range (SDR) sequences, the following metrics are used:

- Peak-Signal to Noise Ratio *PSNR*(*Y*) of luma component as defined in clause 5.5.3,

- Peak-Signal to Noise Ratio *PSNR*(*U*) of chroma component U as defined in clause 5.5.3,

- Peak-Signal to Noise Ratio *PSNR*(*V*) of chroma component V as defined in clause 5.5.3,

- Average colour component PSNR, *PSNR* over all colour components *PSNR*:

 *PSNR* = (6\**PSNR*(*Y*) + *PSNR*(*U*) + *PSNR*(*V*))/ 8

- Multi-Scale Structural Similarity Metric *MS-SSIM*, as specified in [54], [55] and [56] and defined in clause 5.5.4.2

- Video Multimethod Assessment Fusion (VMAF) *VMAF*, as specified in [57]. See clause 5.5.7 on the use of the associated software implementation [59] to compute VMAF.

The exact definition for all SDR quality metrics is based on the software scripts as defined in clause 5.5.7. A reporting scheme for SDR metrics is defined in clause 5.5.6.

#### 5.5.4.2 Structural similarity metric MS-SSIM

The multi-scale SSIM method is illustrated in Figure 5.5.4-1. Taking the original reference sequence $orig(x,y,t,c)$ and reconstructed and decoded sequence $dec(x,y,t,c)$ image signals as the input (referred to as **x** and **y** in Figure 5.5.4-1), the system iteratively applies a low-pass filter and downsamples the filtered image by a factor of 2. The original image is indexed as scale 1, and the highest scale as scale *M*, which is obtained after *M*-1 iterations. At the *j*-th scale, the contrast and structure components are calculated and denoted as Cj (**x**, **y**) and S*j*(**x**, **y**), respectively. The luma component (inappropriately named as the luma component in the references) is computed only at scale *j*=*M* and is denoted as l*M* (**x**, **y**). The overall metric for each frame is obtained by combining the measurement at different scales.



Figure 5.5.4-1: Multi-scale structural similarity measurement system (L: low-pass filter, 2↓: downsampling by 2)

The MS-SSIM between the original image, *I,* and the reconstructed image component, *I’*, is calculated as:

$$MS\\_SSIM\left(I,I^{'}\right)=\frac{1}{N}\sum\_{k=1}^{N}\left([L\_{M}(I\_{k}, I\_{k}^{'})]^{α\_{M}}\prod\_{j=1}^{M}[C\_{j}(I\_{k}, I\_{k}^{'})]^{β\_{j}}[S\_{j}(I\_{k}, I\_{k}^{'})]^{γ\_{j}}\right)$$

$$L\left(I\_{k}, I\_{k}^{'}\right)=\frac{2μ\_{I\_{k}}μ\_{I\_{k}^{'}}+C\_{1}}{μ\_{I\_{k}}^{2}+μ\_{I\_{k}^{'}}^{2}+C\_{1}}$$

$$C\left(I\_{k}, I\_{k}^{'}\right)=\frac{2σ\_{I\_{k}}σ\_{I\_{k}^{'}}+C\_{2}}{σ\_{I\_{k}}^{2}+σ\_{I\_{k}^{'}}^{2}+C\_{2}}$$

$$S\left(I\_{k}, I\_{k}^{'}\right)=\frac{σ\_{I\_{k}I\_{k}^{'}}+C\_{3}}{σ\_{I\_{k}}σ\_{I\_{k}^{'}}+C\_{3}}$$

where $I\_{k},I\_{k}^{'}$ are pixels of original and reconstructed frames, $N$ is the number of pixels per frame, $α\_{M}$, $β\_{j}$, $γ\_{j}$, $C\_{1}$, $C\_{2}$, $C\_{3}$ are the constants:

β1 = γ1= 0.0448, β2 = γ2 = 0.2856, β3 = γ3 = 0.3001, β4 = γ4 = 0.2363, and α5 = β5 = γ5 = 0.1333.

*C*1= (*K*1\*maxValue)2

*C*2= (*K*2\*maxValue)2

 *C*3=*C*2/2,

 *K*1 = 0.01, *K*2 = 0.03, maxValue =(1<<bitDepth)-1, and $M=5$.

As bitDepth is set to 10, the maxValue is set to 1020.

Average values $μ\_{I\_{k}}$,$ μ\_{I\_{k}^{'}}$ at pixels $I\_{k}$ and $I\_{k}^{'} $ are computed as weighted sum of $11×11$ neighbors with Gaussian weights. $σ\_{I\_{k}}$, $σ\_{I\_{k}^{'}}$ represent the variance at pixels $I\_{k}$ and $I\_{k}^{'}$. $σ\_{I\_{k}I\_{k}^{'}}$ is covariance of these two pixels.

The MS-SSIM value for each bitstream is computed as the sum of all individual frame MS-SSIM values divided by the number of frames in the sequence.

See clause 5.5.7, on the use of [59] for computation of MS-SSIM.

It is quite common to convert the MS-SSIM numbers to a dB representation since that representation can be more easily interpreted and is somewhat similar to the PSNR representation. Such computation also has an impact in the BD-rate numbers since the resulting points end up having more similar properties to the PSNR ones for the BD-rate computation. Both HDRTools and VMAF support this conversion, using:

 *log\_MS\_SSIM* = (-10.0 \* log10(1 – *MS\_SSIM*)) in dB;

The *MS\_SSIM* value for each bitstream is computed as the sum of all individual frame *log\_MS\_SSIM* values divided by the number of frames in the sequence. A reporting scheme for SDR metrics is defined in clause 5.5.6.

**===== CHANGE =====**

### 5.5.8 Reference computation of HDR metrics

Computation of *wPSNR(Y)*, *wPSNR(U)* and *wPSNR(V)* metrics as defined in clause 5.5.5 is performed with HDRMetrics tool version 0.23 [67] with reference config file HDRMetrics\_wtPSNR.cfg and hdrTable.txt attached to this report.

Computation of *DeltaE100* and *PSNRL100* metrics requires conversion from YUV to linear light RGB data which is performed with the use of HDRConvert tool [67] and the reference config file HDRConvertYCbCr420ToEXR2020.cfg as attached. After the conversion for reference and decoded video clips is done, computation of *DeltaE100* and *PSNRL100* metrics as defined in clause 5.5.5 is performed with HDRMetrics tool version 0.23 [67] with reference config file HDRMetrics\_deltaE100.cfg as attached.

**===== CHANGE =====**

## 5.7 Tests

Tests may be executed to compare codecs not yet in 3GPP specifications against anchors already defined in 3GPP specification. Tests, equivalently to anchors, are collected in tuples to address different quality and bitrates that can then be used for evaluation over a larger set of operation points.

A test is developed against an anchor and is a combination of:

- The corresponding anchor, which includes

- Scenario

- Reference Sequence

- Test encoder

- Test encoder configuration that provides an equivalent setting to the anchor configuration based on the general encoding constraints in clause 5.6.

- Test tuples creating multiple variants, each including

- Test bitstream

- Test Metrics

- Additional recommended test information includes

- MD5 check sum of the complete reconstructed yuv file (reconstructed test sequence)

- Output picture log for reference encoder

- Output picture log for reference decoder

- Tests are an integral part of the Technical Report

The generation of test tuples is shown in Figure 5.7-1.



Figure 5.7-1: Test Tuple Generation Framework and Test Tuple Metrics Generation

For any coding technology being characterized in the study and reported in this document

* the evaluation is conducted consistently with the framework and test designs defined for the anchors as defined in clauses 5.5 and 6.
* technical documentation to conduct the study is available and provided. Such information includes normative specification text, reference software and description of configuration files, and codec description.
* additional data such as subjective test results (with description of test methodology and conditions) not conducted as part of this study item, encoding tool and configuration file description is also important information that could be provided

**===== CHANGE =====**

## 5.8 Characterization

[…]

The JVET Excel files for the CTC include a VBS script bdrate() to compute the BD-Rate performance between a test codec and a reference from four or five rate-distortion points. However, to address different corner cases as well as to ensure applicability to the reported csv metrics files, the above algorithm is converted into a script python to generate the *BD-Rate Gain* for a codec under test vs. an anchor. Details on the script compare.py are available here https://github.com/haudiobe/5GVideo, please refer to Annex F.

For providing the *BD-Rate Gain* values for a single reference sequence and configuration, the following is used

Usage: python3 compare.py -s <key anchor> <key test>

Example: python3 compare.py -s S1-T01-264 S1-T01-265

For providing the *BD-Rate Gain* values for all reference sequences of one configuration, The following is used

Usage: python3 compare.py -c <key anchor configuration> <key test configuration>

Example: python3 compare.py -c S1-JM-01 S1-HM-01

[…]

**===== CHANGE =====**

## 5.9 Verification

### 5.9.1 Principles

Submissions of anchors, tests and associated metrics are based on 3GPP member input. However, such input to be considered valid is expected of to be verified. Verification in the context of this document includes the following according to Figure 5.9.1-1:

1. Anchor bitstream verification: Anchor bitstreams are correct. By using a defined reference sequence, an anchor configuration as well as a reference encoder, two different executions of this process result in the same anchor bitstream.
2. Anchor reconstruction is correct. By using a verified anchor bitstream, a reference decoder in two different implementations results
	1. in the same anchor sequence,
	2. in the same quality metrics.

The equivalent process is applied for tests.



Figure 5.9.1-1: Verification and cross-check processes

Verification is primarily supported by cross-checks from independent members who carry out the execution of the same process and done by the initial submitter of the anchors/tests. Whereas process 2 is deemed to be of manageable complexity, process 1 is of significantly higher complexity. Based on this it is expected that anchor reconstruction (process 2a) and anchor metrics (process 2b) are cross-checked and verified for every anchor, whereas anchor bitstreams are only cross-checked and verified for one anchor of the entire anchor tuple.

A successful cross-check is defined as follows:

1. Anchor bitstream verification: Initial run and cross-check result in the same size and md5 for the anchor bitstream
2. Anchor reconstruction verification. By using a verified anchor bitstream, a reference decoder in two different implementations results in the same md5 for the anchor sequence.
3. Anchor metrics verification. By using a verified anchor bitstream, a reference decoder in two different implementations results in the same quality metrics identical up to 2 decimal digits for all metrics that are required for characterization.

This technical report includes verified metrics in the main body of the text (i.e. successful completion of stage 3). Verified metrics are agreed to be correct and have been sufficiently cross-checked.

This technical report may also include non-verified anchors, tests and metrics in the Annex G. In this case, the reason for the lacking verification is explained.