INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11 MPEG/M53266
April 2020, Alpbach (AT) Virtual

Source	Philips, Sony
Status	Input document
Title	Report on cross-test model testing of 3VC
Author	Bart Kroon, Danillo Bracco Graziosi

Abstract

This contribution reports on 3VC coding tests (mainly HLS) using TMC2 and TMIV. The contribution lists the issues found and provide TMC2 and TMIV with bug fixes. The recommendation is to integrate these changes into TMC2 10 and TMIV 5 respectively.

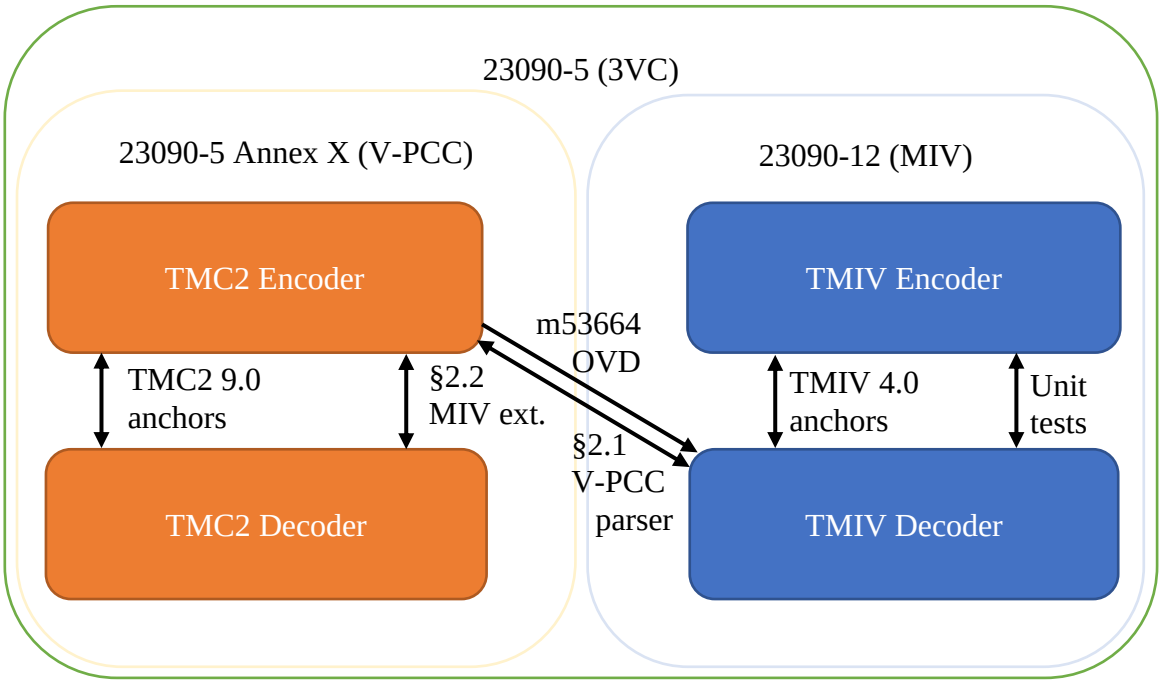


Figure 1: Overview of contributions that contribute to testing 3VC

1 Introduction

In the 129th MPEG meeting it was decided to normatively reference ISO/IEC 23090-5 *Video-based Point Cloud Coding* (V-PCC) [N19089] for Working draft 4 of ISO/IEC 23090-12 *MPEG-I Immersive Video* (MIV) [N19001]. MIV is written as if it is an Annex of Part 5 in the same style as HEVC Annex F. The FDIS of Part 5 will have a common part called Video-based Visual Volumetric Coding (3VC) and V-PCC will become Annex *X*. The *Test Model for Immersive Video*

(TMIV) 4.0 now has an implementation of 3VC and MIV, and naturally TMC 9.0 has an implementation of 3VC and V-PCC.

The step from working draft 3 to 4 was a huge change for MIV and TMIV and it is unlikely that no errors were made in the implementation of all 3VC syntax and decoder processes. Because of that we have performed multiple cross-test model software tests. These tests are in addition to other contributions that also help to test 3VC (Figure 1):

- TMC2 anchors
- TMIV anchors [m53053]
- TMIV unit tests of all parsers/formatters
- Contribution m53664 on encoding a point cloud with MIV syntax

This contribution is an indirect follow up of contributions m51044 (Sony) and m52350 (Philips) on the alignment of MIV with V-PCC.

2 Cross-test model testing

This section describes the implementation and testing effort.

2.1 TMIV V-PCC parsing test

2.1.1 Introduction

The TMIV project includes a suite of unit tests. In particular all HLS parsers/formatters have been unit tested. In addition to that m52350 had provided an integration test that parses a V-PCC bitstream that was produced by TMC2. This work was unfinished and since then many syntax changes have been made to V-PCC and MIV. The m53266 branch of TMIV continues with that effort and adds a V-PCC parsing test to the TMIV test suite.

The goal is not to add full V-PCC support to TMIV but some parts of V-PCC that were used in the V-PCC bitstreams have been implemented in TMIV. For instance, the V-PCC bitstream that was used for m52350 has 1 x 1 tile groups and although this is not supported by MIV WD4, TMIV 4 is able to parse that.

This document is based on the newest available drafts of 3VC and MIV:

- V-PCC d71 [N19089]
- MIV d33 [N19001]

In the remainder of the document we will refer to these documents simply by V-PCC or MIV.

2.1.2 Features to enable efficient testing of 3VC

The TMIV decoder has a validating parser meaning that when something is wrong in the bitstream it is very likely that there will be an error at or a few syntax elements after the mismatch. Both TMC2 and TMIV produce accurate log-files that further help to identify the exact cause of the mismatch. While the TMC2 HLS log-file prints out all descriptors, TMIV prints every syntax element name and value in decoding order. With those log-files and the 3VC specification it is fairly easy to make quick progress.

2.1.3 How to run the tests

Some small V-PCC bitstreams are included in TMIV under MivBitstream/test/ . These need to be replaced every time that there is a new release of TMC2.

1. Download and "build" Catch2
2. Build TMIV with Catch2 support (set `Catch2_DIR` in the CMake GUI)
3. Run the test suite or execute `VpccBitstreamTest`:
    a. Unix Makefiles: `make test`
    b. Visual Studio: build the `ALL_TESTS` target

2.1.4 TMIV Parser executable

This proposal also adds a new executable Parser that is used as follows:

```
Parser -b BITSTREAM
```

The program runs the MIV decoder to print all syntax elements to the standard output but performs no additional operations.

2.2 TMC2 MIV extensions

To continue the support of MIV bitstream encoding using TMC2, the V-PCC+TMIV branch from m51044 was updated with the latest TMC2 v9.0 as well as TMIV v4.0. The MIV extension syntax elements were added to the PCC classes that handle bitstream read/write, following the description from MIV WD4.

One difference between the current branch and MIV is the number of bits to read the pdu_view_id. In MIV, the number of bits is determined by AspsMaxProjections[vuh_atlas_id], which is only transmitted if asps_extended_projection_enabled_flag is set to true. However, this flag is used by V-PCC for another purpose (to indicate 45 degree transformations).

In our recollection the following was already agreed, but still needs to be added. The ASPS has to be extended like this:

asps_extended_projection_enabled_flag	u(1)
if(asps_extended_projection_enabled_flag)	
asps_max_projections_minus1	ue(v)

and the semantics of **pdu_projection_id** needs to be updated to have bit depth:

```
asps_extended_projection_enabled_flag
        ? Ceil(Log2(asps_max_projections_minus1 + 1))
        : 3
```

For V-PCC content, the branch creates bitstream following syntax described in V-PCC. Some of the issues identified during the integration phase with MIV content (see next Section for details) have been address and fixed in the current branch.

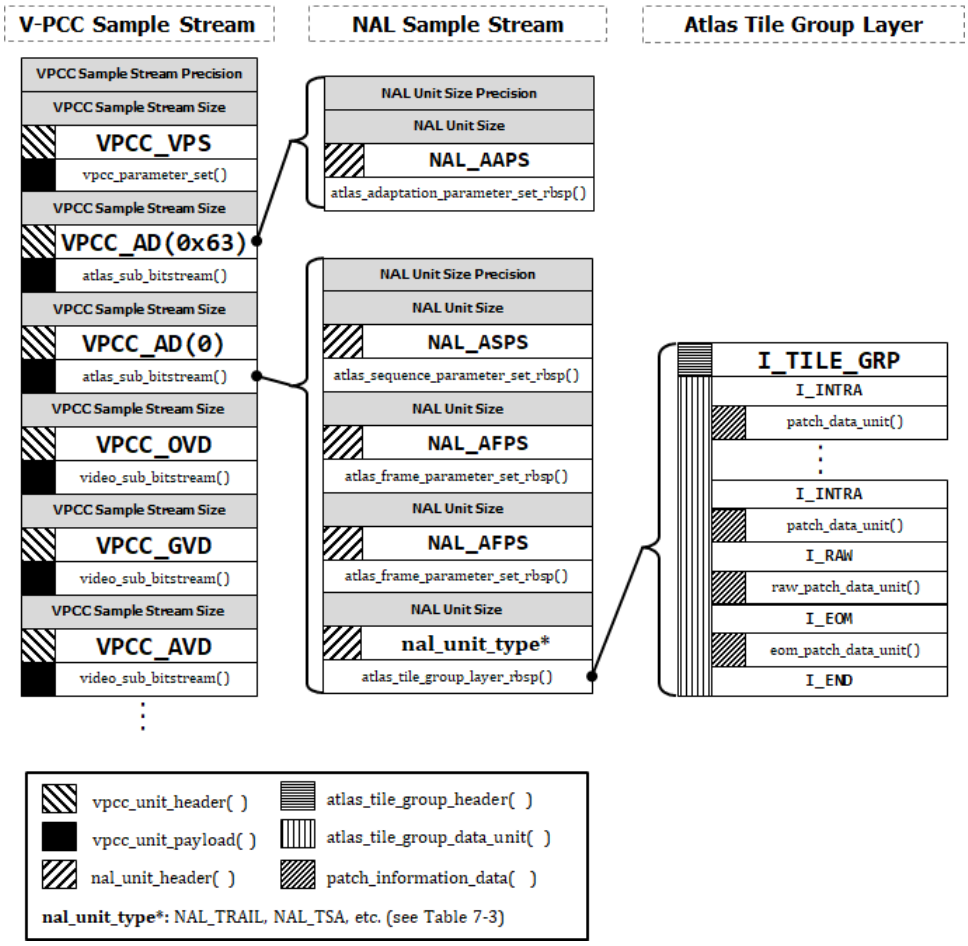
2.2.1 Geometry scaling

In the V-PCC specification, we define only one nominal resolution, and the encoder is free to use any resolution for coding geometry, occupancy and attribute videos. However, it is part of the

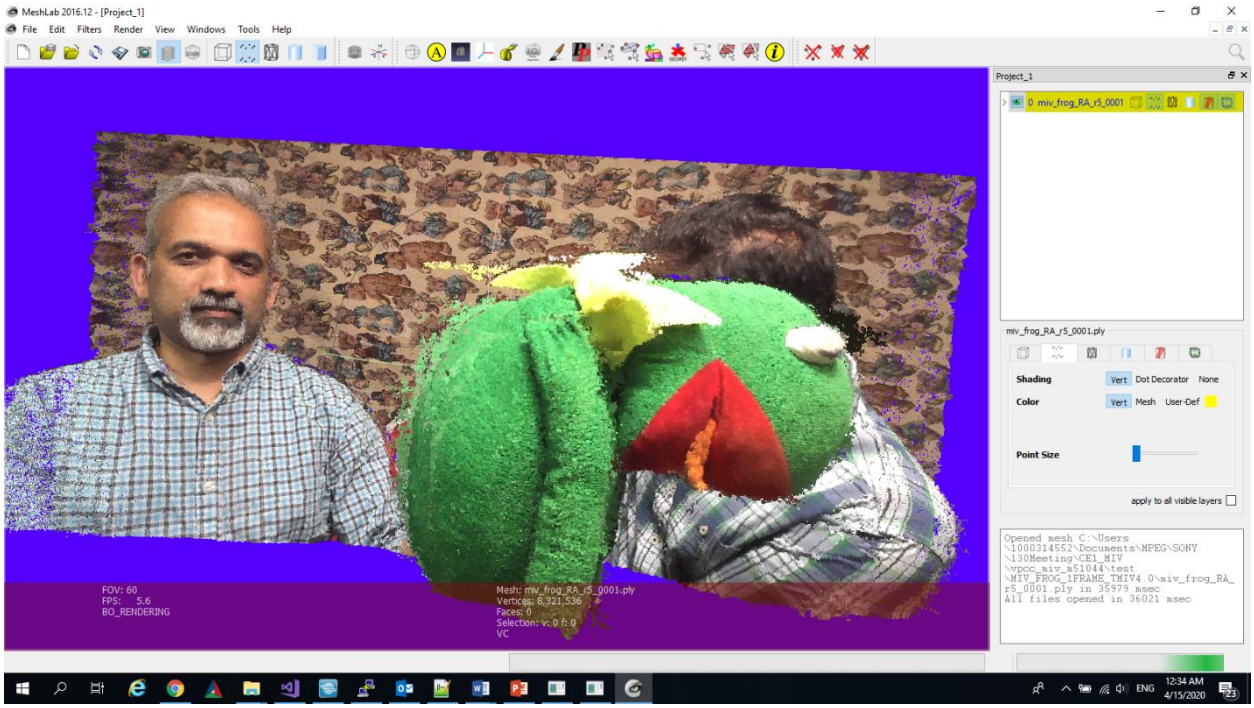
specification, but not mandatory, to have the videos converted to the nominal resolution for further processing/reconstruction. The current branch implements only nearest neighbor upscaling, however the upscaling method already available in the TMIV software could also be used. One recommendation would be to actually create a SEI message to recommend upscaling/filtering parameters to the decoder.

2.2.2 MIV bitstream

The current TMC2 branch generates V-PCC bitstream carrying MIV content as shown in the Figure below. One special atlas (0x63) is generated with only one AAPS, where the cameras are defined. The other atlases do not current have the AAPS structure, and the atlas tile groups have the id of the AAPS equals to 0, which can only be found in the special atlas. Since the id is unique across atlases, the software implemented a search to verify where the indicated AAPS id can be found either on the current atlas, or on the special atlas.



The current structure was implemented and the software is working. We are still conducting more test, but the m51044 decoder is able to read MIV content, and generate a point cloud from the data, as shown in the picture below.



3 Detected issues

Within the context of this work we have identified multiple issues.

3.1 TMC2 issues

The following issues have been identified in TMC2 9.0 and some issues have already been addressed in TMC2 9.1. Remaining issues were preferably addressed directly in the provided TMC2 branch[1]. When that was not possible they were added to the TMC2 issue tracker[2].

1. TMC2 container header (#32)
2. Add vps_reserved_zero_8bits (#33)
3. asps_raw_patch_enabled_flag is not written
4. afps_3d_pos_x_bit_count_minus1 is not written
5. afps_3d_pos_y_bit_count_minus1 is not written
6. afps_lod_mode_enabled_flag is not written
7. arbitrary afps_additional_lt_afoc_lsb_len value (UB)
8. afps_fixed_camera_model_flag has been removed
9. sei_rbsp( ) has to write rbsp_trailing_bits( )
10. I_TILE_GRP and SKIP_TILE_GRP values are swapped
11. First ACL NAL unit is of a non-IRAP type (NAL_TSA)
12. There is no APS in the bitstream

3.2 TMIV issues

The following issues have been identified in TMIV. These issues were preferably be addressed directly in the provided TMIV branch. When that was not possible they were added to the TMIV issue tracker[3].

[1] https://gitlab.com/mpeg-i-visual/vpcc_miv_m51044
[2] http://mpegx.int-evry.fr/software/MPEG/PCC/TM/mpeg-pcc-tmc2/issues
[3] http://mpegx.int-evry.fr/software/MPEG/MIV/RS/TM1/issues

1. ue(v) descriptor
2. afps_2d_pos_x_bit_count_minus1 has been removed
3. afps_2d_pos_y_bit_count_minus1 has been removed
4. afps_output_flag_present_flag is missing
5. atgh_atlas_output_flag is missing and MIV should handle it
6. afps_fixed_camera_model_flag has been removed
7. atgh_adaptation_parameter_set_id is no longer optional
8. byte_alignment() did not start with a one bit
9. Indexing of ai_attribute_MSB_align_flag has changed
10. Implement and apply atgh_pos_min_z_quantizer
11. Implement and apply atgh_pos_delta_max_z_quantizer
12. pdu_view_id coding order has changed
13. Infer pdu_depth_occ_map_threshold when miv_atlas_sequence_params() not present
14. Set gi_geometry_3d_coordinates_bitdepth_minus1
15. GOP structure (see below)

3.2.1 GOP structure

Regarding the ref_list_struct( ), the TMIV Encoder currently outputs the minimum needed to create a *parsable* ATGH although this is unlikely to be correct:

```
asps_long_term_ref_atlas_frames_flag=false
asps_num_ref_atlas_frame_lists_in_asps=1
num_ref_entries( 0 )=0
atgh_ref_atlas_frame_list_sps_flag=true
atgh_atlas_frm_order_cnt_lsb=0
```

The V-PCC CTC bitstreams has:

```
asps_long_term_ref_atlas_frames_flag=false
asps_num_ref_atlas_frame_lists_in_asps=1
num_ref_entries( 0 )=1
DeltaAfocSt( 0, 0 )=-1
```

Should TMIV do the same or should the DeltaAfocSt values reflect the GOP structure of the video sub bitstreams?

3.3 Issues with the 3VC specification

The following issues have been identified in 3VC or MIV. They have been added to the V-PCC[4] and/or MIV[5] issue trackers.

1. Add asps_max_projections_minus1 (see Section 2.2)
2. Update pdu_projection_id semantics (see Section 2.2)
3. Editorial: is it Atlas Adaptation Parameter Set or just Adaptation Parameter Set?
4. rbsp_trailing_bits( ) is not specified nor is there a normative reference

[4] http://mpegx.int-evry.fr/software/MPEG/PCC/Specs/23090-5/issues
[5] http://mpegx.int-evry.fr/software/MPEG/MIV/Specs/23090-12/issues

3.4 Issues with the MIV specification

The following issues have been identified in MIV. They have been added to the MIV issue tracker.

1. How should MIV handle atgh_atlas_output_flag?
2. atgh_pos_delta_max_z_quantizer
3. pdu_depth_start semantics are incomplete
4. masp_depth_occ_map_threshold_flag inconsistent spelling
5. depth_bits unused and wrong

4 Recommendations

The authors recommend to:

1. Integrate the provided TMC2 branch[6] into TMC2 10, including:
    a. Implementation of (part of the) MIV extensions.
    b. TMC2 bug fixes.
2. Integrate the provided TMIV branch `m53266` into TMIV 5, including:
    a. Adding V-PCC bitstream test to the test suite,
    b. TMIV bug fixes.

[6] https://gitlab.com/mpeg-i-visual/vpcc_miv_m51044