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/M52994
April 2020, Online

Source	Philips Research Eindhoven (PRE)
Status	Input document
Title	Proposed simplifications to MIV
Author	Bart Kroon, Bart Sonneveldt

Abstract

This contribution proposes simplifications to the MIV specification based on the design rule that each atlas should be self-contained. In other words: each atlas is its own group and can be decoded and rendered independently of the other atlases. Because of this special atlases are no longer needed, and some parts of TMIV can be simplified. CTC conditions are recommended.

1 Introduction

Working draft 4 of ISO/IEC 23090-12 *MPEG-I Immersive Video* (MIV) [N19001] has been aligned to ISO/IEC 23090-5 *Video-based Volumetric Visual Coding* (3VC). On top of the rich concepts already available in 3VC, MIV has added multiple atlases, special atlases, groups, entities, pruning graphs, persistence of patch data, adaptation parameter sets, etc.

We expect that not all of these extensions of 3VC have a motivation that is strong enough for inclusion in the standard. In this document we propose a simplification of MIV that should:

- improve clarity
- improve spatial access
- allow us to have a committee draft of sufficient quality
- reduce implementation effort

2 The proposal

All proposed changes arise from the addition of a single design rule:

Every atlas is renderable

Another way to say this is:

Each atlas is its own group

We do not know of a use case that break the design rule. We have considered the following use cases (in alignment with the MPEG-I use cases):

- Omnidirectional video (3DoF+)
- Capture and transmission of large events (e.g. sports)
- Cinematic experiences

The reasons that we know exist for having multiple atlases are:

1. Video encoder restrictions (e.g. 4K @ 60Hz)
2. Enabling spatial access
3. Compensating for HM run-time

Reason 1 and 2 can be combined as "spatioangular access" and are important. Reason 3 is not a real-world reason and should not be taken into consideration.

2.1 *Group-based encoding and spatial access*

Let's consider that MIV will be used to capture a large scene that require spatial access. Views are grouped and each group is encoded separately. Clients will receive a subset of the groups, typically only one or two. How this is achieved, is in scope of ISO/IEC 23090-10 *Carriage of 3VC data* and out of scope of MIV.

The original meaning of "grouping" was "to group views". It is not necessary to group atlases because each group can have a single atlas. Note that group-based rendering is no longer used in TMIV because the performance is higher without. Grouping in MIV WD4 is purely to achieve spatial access.

2.2 *Angular access*

Angular access could be implemented in two ways:

- Using (virtual) cameras with smaller FOV's and placing them in different spatioangular groups.
- Tile groups functionality of 3VC.

In both cases all atlases can be independently renderable.

2.2.1 Changes to MIV

- Remove **msp_num_groups_minus1** because msp_num_groups_minus1 is the same as **vps_atlas_count_minus1**.
- Remove **masp_group_id** because masp_group_id is the same as vuh_atlas_id.
- Remove TargetGroupFlag and related text

Alternatively, "groups" could be a more abstract concept that is defined on Systems level and not used in any decoding process of MIV.

2.2.2 Changes to TMIV and CTC

The TMIV encoder and atlas constructor output a single atlas. The TMIV group-based encoder combines multiple atlases. All these components will become simpler: some vectors become scalars and looping over atlases is removed.

We believe that the TMIV encoder and atlas constructor should calculate the atlas frame size based on the projection planes of the views within that group and the following parameters (with suggested values):

- blockSize = 8 (AtlasPatchPackingBlockSize)
- maxAtlasWidth = 4096
- numAtlases = 3
- maxLumaSamplesPerFrame = 2^{25} = 33,554,432

The other parameters are derived as follows:

- maxBlocksPerAtlas = maxLumaSamplesPerFrame / (2 · numAtlases · blockSize^{2})
- atlasBlockWidth[a] = min(
  maxAtlasWidth / blockSize,
  maxBlocksPerAtlas / (maxViewHeight[a] / blockSize))
- atlasBlockHeight[a] = maxBlocksPerAtlas / atlasBlockWidth[a]
- atlasWidth[a] = atlasBlockWidth[a] · blockSize
- atlasHeight[a] = atlasBlockHeight[a] · blockSize

This would result in the per-sequence parameters of Table 1. The maxViewHeight is not actively used but when maxLumaSamplesPerFrame would be lowered (not part of this proposal), then the full equations become important. (Otherwise the basic view would not fit in the atlas.) Table 2 provides values for when maxLumaSamplesPerFrame is 3 2^{23} = 25,165,824.

Table 1: Per-sequence atlas frame parameters (according to the proposal)

Sequence	**Number of atlases**	**Max. view height**	**Nominal atlas width**	**Nominal atlas height**
SA, SB, SC, SN	1	2048	4096	4096
SD, SP, SU, ST	3	1088	4096	1360
SE, SL	3	1080	4096	1360

Table 2: Per-sequence atlas frame parameters (for illustration only)

Sequence	**Number of atlases**	**Max. view height**	**Nominal atlas width**	**Nominal atlas height**
SA, SB, SC, SN	1	2048	4096	3072
SD, SP, SU, ST	3	1088	3880	1088
SE, SL	3	1080	3848	1080

2.3 Special atlases and adaptation parameter set

We could not identify a need for a special atlas:

- Patch data requires more bitrate than camera data

- Special atlases complicate things at the Systems level. (It is a bad design.)

- When MIV is used to transmit full views (no frustum slices) than each view is renderable and thus can be a separate atlas. In that case each atlas would have an APS with a single view.

- When MIV is used with group-based encoding each group has a different set of views with little or no overlap between groups.

The proposal is to remove this concept from MIV.

2.3.1 Changes to MIV

The vuh_atlas_id value 0x3F will not be special anymore. Remove all occurrences.

2.3.2 Changes to TMIV

Remove support for special atlases. The decoder will become simpler.

2.4 Pruning graph

It is possible to have some overlap in views between atlases (groups), but pruning cannot be across atlases because it would not be possible to render one atlas without the other. The pruning graphs will need to be disjoint. Current TMIV pruning graphs are already disjoint across group boundaries because there is no overlap in views.

The proposal is to keep a pruning graph per atlas like currently defined.

2.4.1 Changes to MIV

No additional changes are required to the current specification.

The values of pdu_view_id index into the view parameters that are part of the same atlas.

2.4.2 Changes to TMIV

- The ViewWeigthedSynthesizer needs to be modified. It uses one list of "projection helpers" across all views. This modification was already needed to fully support WD4.
- The AdditionalSynthesizer needs a small modification because the FOV calculation uses all views. We propose to modify the FOV calculation to only use the views in the current atlas.

3 More TMIV simplifications

While less important than the other proposed changes, we also propose to remove components of TMIV that are not actively used and do not contribute (anymore) to the standardization process but require effort to maintain.

For discussion. Is there a need or should we remove:

- AdditiveSynthesizer and support classes,
- GroupBasedRenderer,
- MultiPassRenderer?

We request to investigate if the EntityBasedAtlasConstructor can be merged with the AtlasConstructor without increasing the complexity of AtlasConstructor too much. Many WD4 changes had to be applied twice to both atlas constructors.

We noticed that most of the real encoding work is now within the AtlasConstructor while the Encoder is nothing more than a pass-through. We recommend to merge the AtlasConstructor and Encoder classes. We believe that this improves the readability of TMIV 5.

4 Conclusions

The proposed changes to MIV result in a CD that is clearer, enables spatial access, allows us to have a committee draft of sufficient quality and reduces implementation effort.

The proposed changes to TMIV result in a simpler test model with equal performance.