**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 4**

**MPEG VIDEO CODING**

**ISO/IEC JTC 1/SC 29/WG 4 m** **56631**

**April 2021, Online**

**Title: Long-term memory requirement proposal for MIV future work (MIV v2)**

**Source: Renaud Doré (Interdigital), Bertrand Chupeau (Interdigital)**

**Abstract**

This contribution proposes one requirement for MIV v2 based on a long-term memory functionality. It would in particular allow to address the bandwidth efficient transmission of stable scene parts in the context of a client-server broadband architecture. The connection to Scene Description is questioned.

# Introduction

A discussion has recently been launched on the topics which MIV v2 should address [1]. In this context this contribution proposes one topic of requirements related to the transmission of scene parts which are stable in geometry and appearance over a long-term period (significantly longer than a typical video encoding intra-period of 32 frames).

# Broadband context

MIV can be used in broadcast or broadband context. But the non-trivial ways of acquiring contents with rigs of multiple cameras and the generation of geometrical information on one side, and the use case on another side, suggest that immersive contents will typically be more short length and high value contents to be consumed in a broadband context than a commodity media in a broadcast channel.

This is why it seems important to (non exclusively) have dedicated requirements to broadband MIV applications.

# A few facts about MIV v1 technology

## Bandwidth limitation

The bitrates addressed by MIV range from a few Mbits/s to 50 Mbits/s. The maximum 50 Mbits value – although already a challenging bandwidth for a lot of delivery networks – constitutes a limitation for a number of MIV contents which leads to visual artefacts even at this highest target.

The reasons for MIV being so “bandwidth hungry” are multiple:

* Large field of view content,
* Need to transmit a geometric information in addition to texture, should it be under the form of depth maps or transparency attribute information,
* Complex self and inter-objects occlusions to deal with, the amount of which increases in proportion to the scene field of view and complexity.

Yet there cannot be any compromise done on visual quality because otherwise why targeting “immersive video experience”?

We should therefore try to make a better use of the available bandwidth and obtain unquestionable visual quality for the high bitrate range.

## Immersive contents with large stable scene parts

The nature of multicamera rig acquisition also suggests that the capture equipment will often be set up at a given place and will focus on foreground objects surrounded by a largely static environment. As a matter of facts, almost all current MPEG contents are shot in these conditions, as illustrated in Figure 1.



Figure 1: Snapshots of typical MIV test contents exhibiting static environment

Of course a video codec knows how to exploit the temporal redundancies of an input video to reduce the bitrate. But what is presented as input to e.g. VVC video encoders by an MIV encoder are patch atlases which layout is typically reset every other 32 frames, without any specific attention paid to what belongs to the static environment and what belongs to moving foreground objects. Whatever the performance of the video encoders, the related bitrate is not negligible, and the pixel rate remains extremely high.

A broadband configuration would allow to prevent the scene environment to be re-sent at the video frame rate, but only at the request of the client, similar to what is done with a server-client game architecture.

## Imperfect inpainting process

MIV v1 integrates a feature to help inpainting at the client side. Yet, it requires extra bandwidth and above all, inpainting remains hazardous.

The presence of an important static part could be beneficial for preventing uncertain inpainting process. In the figure below where a character enters a room, it would be safer to reuse the environment shot when the room was empty (left part of the figure) than to try to guess the parts occluded by the character as currently done with MIV v1. Of course, lighting may have changed a bit but the results have a good chance to be visually more acceptable than weird things appearing sometimes around the character silhouette.

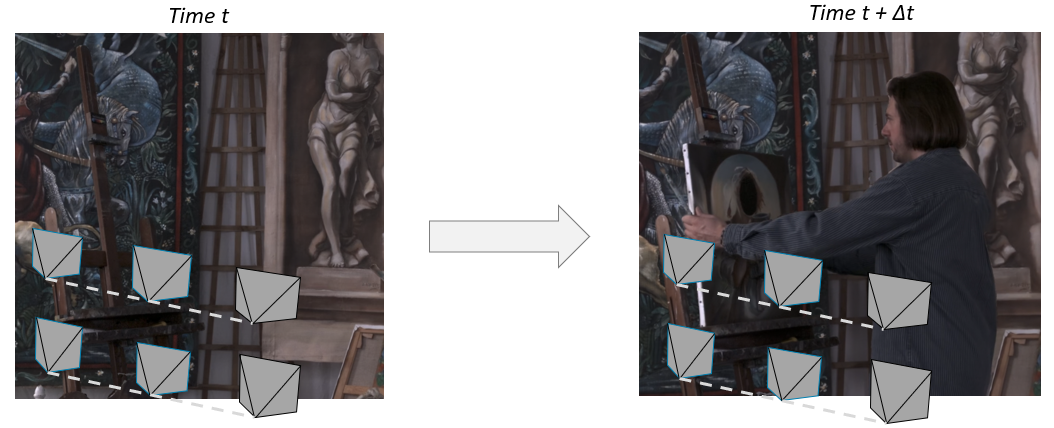


Figure 2: Reuse previous static component rather than try inpainting

# Requirement for a long term memory feature

The previously mentioned wish to stream parts of the scene on request in a game-server like scheme would require that:

1. the immersive scene can be divided into relatively stable parts (scene partition)
2. these parts should be referenceable in the server-client exchanges and transmitted in a differentiated manner
3. there is a local long term storage function in the client allowing to compose those parts for the final rendering

The storage function corresponds to a long term memory beyond the today’s intra-period horizon of the video codecs.

A generic architecture for an MIV broadband application is depicted in Figure 3 hereunder with such feature.

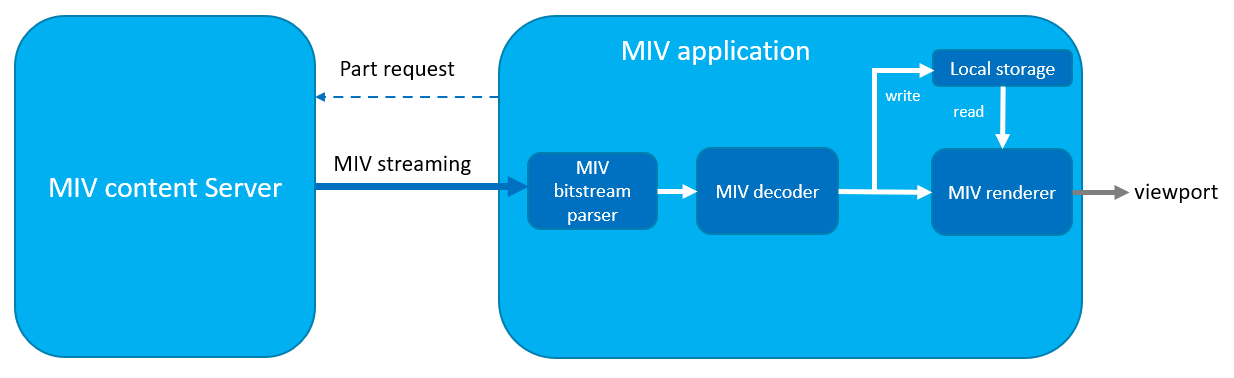


Figure 3: Generic architecture for an MIV broadband application

In the Figure 4 below, the proposed new feature for MIV v2 is illustrated for the MIV client side. The current MIV v1 specification is delineated in red. The long term memory feature corresponds to the green part and would consist in a local storage and some means to create and maintain this database: make new item entries, delete some items or may-be update some of them.

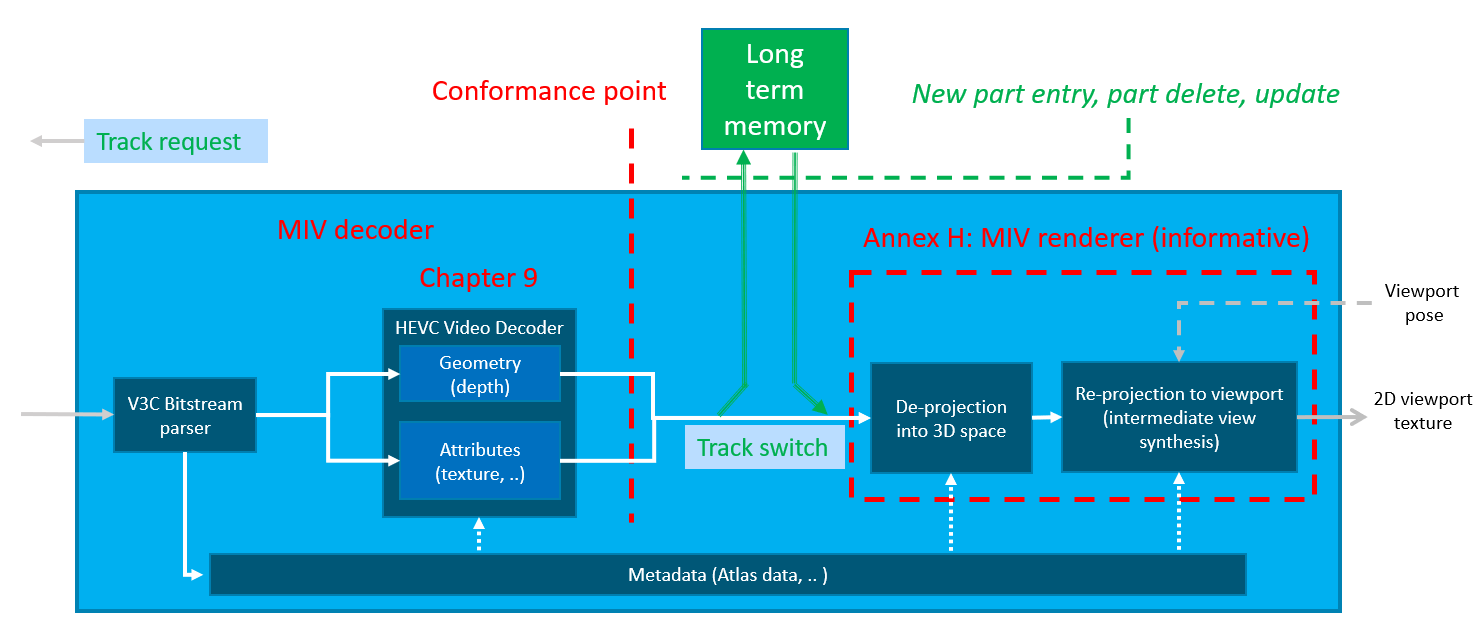


Figure 4: Long term memory feature proposal (in green)

# Is this about Scene Description?

Scene Description (SD, part 14) is in charge of composing an immersive scene from a variety of media resources, such as MIV contents, V-PCC objects, CGI environments, etc.

There may be some flavour of scene composition in the proposed differentiated transmission and processing of stable scene parts within an MIV encoded content. But contrary to elements composed at SD level and generated by **out-of-band means** like CGI elements, what we propose here is to allow the generation of composition elements by **in-band means**, where these composition elements are conveyed by the regular MIV streaming pipe.

In order to compose the MIV stream with these in-band recorded and properly referenced elements, the MIV renderer would typically operate within the presentation engine which would fetch by the Media Access Function both what is streaming from the MIV pipeline and why is stored as previously recorded MIV elements (cf Figure below), but that requires a deeper analysis and goes beyond the sole requirement aspect of this contribution.



Figure : Media pipeline in Scene Description

So while it is relevant to address the proposed requirement at the MIV standard level already, there should be some investigations on possible connections with other requirements at the higher SD level.

# MIV technology used for camera captured scene generation

Let’s try to figure out this requirement more in perspective.

The distinction has already been done between CGI scene generation and camera captured generation [2]. We believe that MIV v2 can be linked to the “camera captured scene’ and positioned in-between two extreme situations:

* On one side, a current MIV v1 scene is continuously streamed. The rendering is quite light at the reception side since no interaction with light is computed, but the bandwidth is high.
* On another side, a CGI scene like in a game is composed of a multiplicity of objects which are transmitted when needed. The bandwidth is limited since objects are transmitted only once, but the rendering requires important resources if photorealistic effect is targeted, should it be in rasterization mode and above all in ray-tracing mode.
* In-between, a MIV v2 could be a mix of previous both situations where the dynamic parts of the scene are continuously streamed but some objects are transmitted only once, leading to a drastic pixel rate and bitrate reduction in some cases. Such a camera captured scene generation would consist in defining a number of scene parts which can then be addressed individually, e.g. fetched, updated or deleted.

# Conclusion and recommendations

We propose to draft a requirement for MIV v2 which would enable broadband scenarios where temporally stable scene parts (in geometry and appearance) can be:

* identified within the MIV bitstream and referenced in server-client exchange
* transmitted in a differentiated manner (from the rest of the bitstream)
* locally stored at client side to allow composition at final rendering

The advantage would be first to allow a differentiated streaming strategy of different scene parts, making distinction of scene parts to be streamed continuously versus rather static parts to be sent less often, which could benefit:

* either to exhibit better visual quality for a given bitrate,
* or to use less bitrate for a given quality.

Another benefit would be to prevent uncertain inpainting.

In a more advanced perspective, such feature would allow to consider MIV technology as a step further towards a scene transmission with some commonalities with game scene transmission.

We recognize that implementing this requirement could come up with possible technical issues, but that would be namely what MIV v2 should investigate.

We therefore recommend to add such a “long term memory” requirement to MIV v2 investigations and also to check how it would articulate with the Scene Description specification.

# References

1. Discussions on MIV future work, ISO/IEC JTC1/SC29/WG4 MPEG134/M56350, April 2021
2. Peter Ludé (CableLabs), Arianne T. Hinds (Tencent America), [MPEG-I Future] Immersive media content creation workflows for immersive displays, ISO/IEC JTC1/SC29/WG2 MPEG133/M56057, January 2021