**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC1/SC29/WG11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 MPEG2020/m54417**

**July 2020, MPEG131 online**

|  |  |
| --- | --- |
| **Source** | **Interdigital** |
| **Status** | **Input document** |
| **Title** | **Cardboard patches for MIV** |
| **Author** | Julien Fleureau, Bertrand Chupeau, Renaud Doré |

# Abstract

This contribution is about signaling a new type of so-called “cardboard” patches, which geometry boils down to a single depth value per patch, with the MIV syntax. For that purpose, a new flag is proposed, at patch level and atlas level, together with solutions to express this single depth value without resorting to the depth atlas component frame.

# Introduction

The MIV standard conveys the volumetric information through a set of patches consisting of a pair of geometry (depth) and texture components. However, some simple 3D scenes that we call “cardboard” scenes in the following may not require such level of complexity to be described because of their intrinsic nature. Indeed, they are simply composed of a set of static stacked planar (or spherical, depending on the projection) layers, such as illustrated in Figure 1. Due do this specific shape, transmitting a depth map information to represent a carboard object is overkill as it can be represented by a simple constant value from a given camera point of view. In a typical scenario, theses patches may be found away from the very first foreground in order to complete a scene and model an omnidirectional background (so-called “envmap”) in a cheap way. They could also be used in combination with regular patches to considerably ease the production of volumetric content by avoiding the need to estimate complex geometries. An example of such a content among the CTC ones is InterdigitalHijack. Furthermore, cardboard only contents could be rendered with dedicated synthesis algorithm (Painter’s algorithm especially) which may dramatically decrease the overall rendering complexity and ease the implementation on low-end devices.

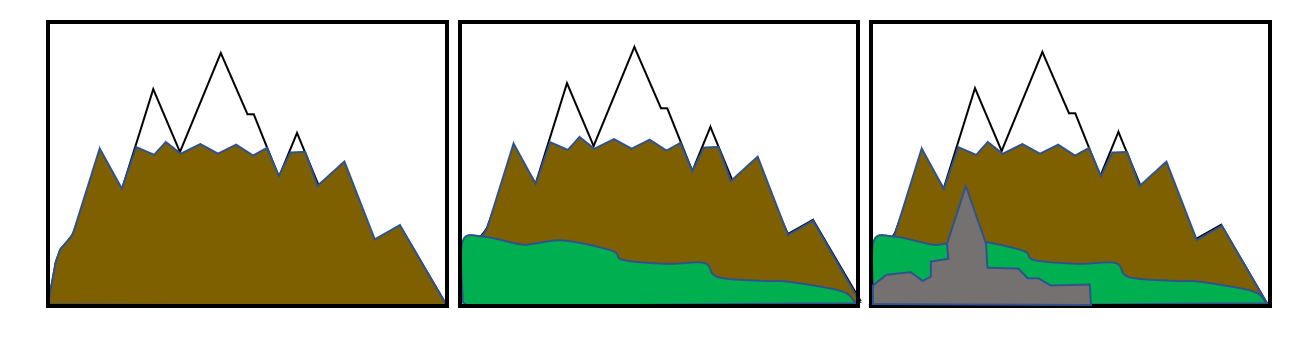


Figure 1: Example of “cardboard-like” scene where successive planar objects are simply stacked to represent the 3D scene.

In this contribution, we propose a new flag in the MIV syntax – at patch and atlas levels – to support the signaling and efficient encoding of cardboard-like objects. Specific types of patches requiring no explicit geometry information in the depth atlas but only a single depth value would thus be considered. This would allow reducing the depth atlas size leading to an overall pixel rate as well as bitrate reduction. The decoder could also take advantage of this information to speed-up the scene rendering. Finally, since their depth information is embedded in the metadata bitstream only, such patches would not suffer from any depth coding artifacts.

# Syntax updates for cardboard patches

Different options can be envisioned to encapsulate the signaling of cardboard patches in the MIV syntax, implying different levels of flexibility for the standard. Two different solutions at the patch and atlas levels are exposed hereafter.

## Patch level

### Signaling of cardboard patches

The signaling of cardboard patches can be expressed at the patch level, within the MIV extension of the patch\_data\_unit( p ) structure, as described in Table 1.

|  |  |
| --- | --- |
| pdu\_miv\_extension( p ) { | **Descriptor** |
| if( vme\_max\_entities\_minus1 > 0 ) |  |
| **pdu\_entity\_id**[ p ] | u(v) |
| if( asme\_depth\_occ\_threshold\_flag ) |  |
| **pdu\_depth\_occ\_threshold**[ p ] | u(v) |
| **pdu\_cardboard\_flag**[ p ] | u(1) |
| if ( pdu\_cardboard\_flag[ p ] ) { |  |
| **pdu\_fully\_occupied\_flag**[ p ] | u(1) |
| **pdu\_zconst**[ p ] | fl(16) |
| } |  |
| } |  |

Table 1: MIV syntax update at patch level (changes in red).

**pdu\_cardboard\_flag[** **p** **]** equal to 1 indicates that the patch is a “cardboard” patch, i.e. of constant depth value and without attached patch depth component in the depth patch atlas. pdu\_cardboard\_flag[ p ] equal to 0 indicates that the patch is a regular patch. When not present, the value of pdu\_cardboard\_flag[ p ] is inferred to be equal to 0.

**pdu\_fully\_occupied\_flag[** **p** **]** equal to 1 indicates that all pixels within the patch are valid pixels. pdu\_fully\_occupied\_flag[ p ] equal to 0 indicates that there may be invalid pixels within the patch.

### Signaling of cardboard patch constant depth value

To signal the depth information of the carboard patches, a first solution is **to add a new pdu\_zconst[** **p** **] syntax element** to the MIV extension of the patch\_data\_unit( p ) structure as described in Table 1.

An alternative solution is **to reinterpret (new semantics) the value of the pdu\_depth\_start syntax element** of the patch\_data\_unit( p ) syntax structure as the actual depth of the patch (and pdu\_depth\_end should be enforced to be equal to pdu\_depth\_start).

## Atlas level

A second and complementary way to do is to embed the signaling at the atlas level. In that scenario, the asme\_cardboard\_flag flag within the ASPS syntax structure (see Table 3) would inform the decoder that all patches associated to this specific atlas are carboard patches and that depth information should be retrieved from per patch information rather than from a geometry atlas frame. One could indeed decide to group all cardboard patches within a dedicated atlas which would therefore not require any associated depth / geometry component.

|  |  |
| --- | --- |
| pdu\_miv\_extension( p ) { | **Descriptor** |
| if( vme\_max\_entities\_minus1 > 0 ) |  |
| **pdu\_entity\_id**[ p ] | u(v) |
| if( asme\_depth\_occ\_threshold\_flag ) |  |
| **pdu\_depth\_occ\_threshold**[ p ] | u(v) |
| if ( asme\_cardboard\_flag) { |  |
| **pdu\_fully\_occupied\_flag**[ p ] | u(1) |
| **pdu\_zconst**[ p ] | fl(16) |
| } |  |
| } |  |

Table 2: MIV syntax update at atlas level – patch data unit (changes in red).

|  |  |
| --- | --- |
| asps\_miv\_extension( ) { | **Descriptor** |
| **asme\_group\_id** | u(v) |
| **asme\_auxiliary\_atlas\_flag** | u(1) |
| **asme\_depth\_occ\_threshold\_flag** | u(1) |
| if( vme\_geometry\_scale\_enabled\_flag == 1 ) { |  |
| **asme\_geometry\_scale\_factor\_x\_minus1** | u(v) |
| **asme\_geometry\_scale\_factor\_y\_minus1** | u(v) |
| } |  |
| **asme\_cardboard\_flag** | u(1) |
| } |  |

Table 3: MIV syntax update at atlas level – ASPS (changes in red).

If asme\_cardboard\_flag equals to 1, the single depth value per patch is read from the patch\_data\_unit( p ) structure similarly to what have been described before (see Table 2).

# Geometry packing and occupancy

As mentioned before, cardboard patches do not require to transmit any geometry component anymore but there still is a need to transmit a key information in the case where a cardboard patch is not a fully occupied rectangle. The 3 following options may be considered:

1. Use an occupancy map, if present and validated in a profile. To our opinion, this is the only interest of having a specific bitstream for occupancy not embedded in the depth map.
2. Use the current geometry atlas to embed the occupancy information only, making use of the existing occupancy map threshold mechanism. The depth information is still read from the patch\_data\_unit( p ) structure. The depth value in the geometry atlas may be set to any constant value above the threshold. Note that in the case of an atlas with cardboard-like patches only, the associated geometry frame could be 8-bit (instead of 10) dividing by 2 the required pixel rate.
3. Validate an alpha attribute component, which would open the door to non binary key or transparency support as well.

As far as packing is concerned:

* If all carboard patches are “fully occupied” (envmap for instance) then, no need for any geometry / occupancy packing (vps\_geometry\_video\_present\_flag[ j ] equal to 0),
* If all cardboard patches are “not fully occupied”, then occupancy and texture attribute should be symmetrically packed (similarly to what is currently done in MIV)
* If a mix of “fully occupied” and “non fully occupied” carboard patches are packed in a same atlas, then specific packing strategies at the encoding stage have to be used to ensure the symmetry between occupancy and texture attribute packing of “non fully occupied” patches (pack “non fully occupied” patches first for instance).

# Conclusion and recommendations

The ability to signal “cardboard” patches, i.e. patches which geometry is defined by a single depth value, could yield significant pixel rate and bitrate saving for a number of 3D contents. The decoder could also take advantage of this information to speed-up the scene rendering. A better robustness to video coding is also ensured by the way the depth information is stored within the metadata bistream.

We therefore recommend the adoption within the MIV specification of a cardboard patch signaling mechanism, at the atlas level, completed with a signaling of the depth and occupancy status at the patch level (as described in section 2.2).

We draw attention that the use of non-fully occupied cardboard patches requires the activation of an occupancy map mechanism.