ISO/IEC JTC 1/SC 29/WG 03 N0987

**ISO/IEC JTC 1/SC 29/WG 03  
MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title:** Exploration of carriage of depth map and alpha map as a new media type in ISOBMFF

**Status:** Approved

**Date of document:** 2023-08-04

**Source:** ISO/IEC JTC 1/SC 29/WG 03

**No. of pages:** 3 (with cover page)

**Email of Convenor:** young.L @ samsung . com

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg3>

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 03 MPEG SYSTEMS**

**ISO/IEC JTC 1/SC 29/WG 03 N0987**

**August 2023, Geneva, CH**

|  |  |
| --- | --- |
| **Title** | **Exploration of carriage of depth map and alpha map as a new media type in ISOBMFF** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **22985** |

# Introduction

This document collects technologies that are relevant for carriage of depth and alpha maps and identifies their possible gaps in enabling current use cases. To address these possible gaps, this document also aims at providing directions for possible future technical work based on ISOBMFF-related technologies .

# Table of contents

[Introduction 1](#_Toc142051468)

[Table of contents 1](#_Toc142051469)

[1. Use Cases and examples 1](#_Toc142051470)

[2. Existing Technologies 2](#_Toc142051471)

[3. Gaps and Shortcomings 4](#_Toc142051472)

[4. Possible future work 4](#_Toc142051473)

[5. Links and References 5](#_Toc142051474)

# Use Cases and examples

When it comes to carriage of depth, it is commonly assumed to be associated with a video stream, and encoded using a video codec. However, this paradigm is shifting, especially with the advent of AR, since there could be depth streams not necessarily tied to a video stream and new non video-based, depth-specific codecs emerge. Following are some concrete examples that can put the carriage of depth data technologies in perspective.

(1) ***AR recording***: Android has the Depth API (of ARCore), that can be used to record the depth of a scene. This recording can be stored by an AR-enabled device (e.g. AR Glasses, Smartphone…) and it can use this depth information to add 3D Effects during the playback of the recorded video of the scene.

(2) ***Depth from multiple videos***: In this case, multiple video cameras can be used to record a scene, and from the video recordings the depth is calculated. This depth data is standalone data (i.e. does not correspond to a single video), therefore it should be handled undependably. This use case cannot be addressed with current standards since it is assumed that depth always accompanies a video. This becomes more and more the case with AR glasses having several sensors for depth but a smaller number for RGB recording.

(3) ***Non-video coded depth***: Recently, there has been research on other ways to code depth with specific depth information compression technique, e.g. the RVL by Microsoft [2].

(4) ***Split rendering***: When the rendering of a scene takes place at a different device than the end device (e.g. rendering on a smartphone – viewing on tethered AR glasses), a depth layer of the rendered scene can be used to correct the image right before the final display.

(5) ***MR Placement***: Placing a virtual object (rendered either locally or remotely) on a scene can require both depth and alpha maps to appear more realistic.

# Existing Technologies

Currently most of the technology for carriage of depth and alpha maps is within 14496-12 ISOBMFF and 23001-17 Carriage of Uncompressed Video, with references from other specifications (e.g. 23002-3 for metadata).

[Editor’s Note: The sections currently focuses on depth, similar study of existing technologies needs to be done for alpha maps]

## ISO/IEC 14496-12: ISO base media file format

In ISOBMFF, depth can be carried in an auxiliary video track (that can be referenced to a video track), used as following:

Auxiliary video media uses the 'auxv' handler type in the HandlerBox of the MediaBox, as defined in 8.4.3.

An auxiliary video track is coded the same as a video track, but uses this different handler type, and is not intended to be visually displayed (e.g. it contains depth information, or other monochrome or color two-dimensional information). Auxiliary video tracks are usually linked to a video track by an appropriate track reference.

The reference\_type can be set to 'vdep' in reference to a video track:

The reference\_type shall be set to one of the following values, or a value registered or from a derived specification or registration:

…

* 'vdep' this track contains auxiliary depth video information for the referenced video track.
* 'auxl' this track contains auxiliary media for the indicated track (e.g. depth map or alpha plane for video).

…

NOTE 1 A track with reference type 'auxl' could have a coding dependency; its use is clarified by specifications that use it.

NOTE 2 When multiple track references would describe an auxiliary video track, derived specifications might constrain or recommend which track references are used. For example, derived specifications might constrain or recommend whether to use 'vdep' or 'auxl' or both for auxiliary depth video track

The only other provision in ISOBMFF spec is for auxiliary video metadata (item of type 'auvd'), that references ISO/IEC 23002-3 (which only provides near and far pane and informative methods for calculating stereo views):

An auxiliary video track used for depth or parallax information may carry a metadata item of type 'auvd' (auxiliary video descriptor); the data of that item shall be exactly one si\_rbsp() as specified in ISO/IEC 23002-3. (Note that si\_rbsp() is externally framed, and the length is supplied by the item location information in the file format). There may be more than one of these metadata items (e.g. one for parallax info and one for depth, in the case that the same stream serves).

## ISO/IEC 23001-17: Carriage of Uncompressed Video and Images in ISOBMFF

The provisions for depth in 23001-17 are as follows. Note that the carriage is not defined as the specification assumes any time of video content:

**Depth Mapping Information**

**Definition**

Box Type: 'depi'   
Container: Video sample entry, ItemPropertyContainerBox  
Mandatory: No  
Quantity: Zero or one per video sample entry or associated per item

The DepthMappingInformationBox may be used to describe how values in a depth map are transformed into distance values. If not present, the mapping from depth value to distance values follow the default values (nknear=128, nkfar=128) as defined in ISO/IEC 23002-3.

If this box is used as an item property, it shall be marked as essential.

**Syntax**

class DepthInfoBox extends FullBox('depi', 0, 0) {  
 unsigned int(16) component\_count;  
 {  
 unsigned int(16) component\_index;  
 } [component\_count]  
 unsigned int(8) nknear;  
 unsigned int(8) nkfar;  
}

**Semantics**

component\_count indicates the number of components to which the depth mapping information. If this value is 0, the depth mapping information applies to all disparity components of the image.

component\_index indicates the 0-based index of the component listed in the associated ComponentDefinitionBox.

nknear, nkfar near and far distances have the same semantics as defined in ISO/IEC 23002-3.

# Gaps and Shortcomings

From the use cases above, the following assumptions can be made regarding the features that should be enabled in the work for carriage of depth data.

*Assumption 1*: The depth is not necessarily accompanying a (single) video – therefore should be able to be stored on its own.

*Assumption 2*: Depth data might not be coded using a video codec – therefore carriage of non-video codec coded depth should be supported.

*Assumption 3*: The depth and texture (i.e. video) are not necessarily spatiotemporally aligned – therefore alignment information should be signalled, if needed.

*Assumption 4*: Legacy players will not be able to play depth content – therefore it should be distinguishable from regular media content.

# Possible future work

There is need to further elaborate on how the current shortcoming can be addressed, including possibilities such as improving the support in ISOBMFF.

## High-level design in ISOBMFF

To address the insights from Section 3, the following high-level design can be formulated:

- Define a new media handler for depth data (e.g. 'depth'), to enable carriage of depth data (assumption 1 and 2)

- Possibly defining brand(s) for warning about the presence of depth and possible metadata so that legacy player don’t attempt to play back a depth track when it is video encoded (assumption 4)

- Define necessary information metadata about spatiotemporal alignment information for carrying depth data (assumption 3)

Additionally to the above, existing features might need to be maintained, revisited in light of the new proposed high-level design (e.g. referencing between video and depth tracks) but this could be treated as legacy techniques and no backward compatibility seems to be needed.

# Links and References

[1] Google – Depth Image Encoding - https://sites.google.com/site/brainrobotdata/home/depth-image-encoding

[2] A. Wilson, “Fast Lossless Depth Image Compression”, ACM ISS '17, October 17–20, 2017, Brighton, United Kingdom