ISO/IEC JTC 1/SC 29/WG 03 N1218

**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:** 2024-04-26

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

**No. of pages:** 13 (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 N1218**

**April 2024, Rennes, FR**

|  |  |
| --- | --- |
| **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** | **23828** |

# Table of contents

[Table of contents 1](#_Toc165022281)

[1. Introduction 1](#_Toc165022282)

[2. Definitions 1](#_Toc165022283)

[3. Use cases and examples 4](#_Toc165022284)

[4. Existing technologies 7](#_Toc165022285)

[5. Gaps in existing technologies 9](#_Toc165022286)

[6. Discussion 10](#_Toc165022287)

[7. Possible future work 10](#_Toc165022288)

[8. Terminology 10](#_Toc165022289)

[9. Links and References 12](#_Toc165022290)

# Introduction

This document collects technologies that are relevant for carriage of depth, alpha, semantic and other 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.

# Definitions

**Depth image**

We define a ***depth image*** as a set of 3D coordinates[[1]](#footnote-2) captured from a viewpoint by a virtual or physical camera. A depth image can be recorded and used regardless of whether a colour image is recorded too or not.

**Depth map**

We define a ***depth map***[[2]](#footnote-3) as a set of depth or distance values obtained after projecting a depth image onto a 2D plane – i.e. the x and y coordinates of the depth image are transformed or mapped into x and y positions of a 2D image or matrix. Therefore, a depth map is a 2D image (matrix) with the intensity of each xy pixel (entry) representing a depth or distance value. This information can be conveyed as a grayscale image, after potential quantization of acquired floating-point depth/distance values to a suitable range (i.e., bit depths), depending on the required resolution. A depth map may (or may not) accompany a colour texture and may be spatially aligned to the accompanying colour texture (if there is one). Aligned images of colour and depth are called RGBD images.

For example, if a device is equipped with both a depth and a colour camera (e.g. MS Kinect v1), the raw capture data of the depth sensor is a depth image, while the depth map for the corresponding colour texture can be obtained after processing and mapping to a 2D plane the depth image. In such cases, even if the resolution of the depth map differs from the colour texture, the two images are spatially aligned, i.e. when they are resized, a colour value corresponds to a depth value at the respective x, y positions.

.

A close-up of a purple flower

Description automatically generated A black and white flower

Description automatically generated

Image from[[3]](#footnote-4)

**Alpha map**

An alpha map contains transparency information for the pixels of a colour texture (e.g. image, video frame, surface). As such, it is used paired with the corresponding colour image. This information can be conveyed as a grayscale image where a gray level corresponds to a transparency level (e.g., 0 indicates full transparency, 1 indicates full opacity). The alpha map is spatially aligned to the colour texture, and an alpha blending operation takes place to produce the final rendition. The colour texture can be either in RGB or YUV color format. The transparency information can also be represented as an extra channel to a given color format, such as RGBA.

A white circle with black arrows

Description automatically generated

Image from[[4]](#footnote-5)

**Semantic map**

A semantic map is a map that contains semantic information for the pixels of an image. A semantic map can be represented via an image format, in which every sample is assigned to one label from a set of pre-defined labels, that indicates the semantic class of the corresponding image pixel. Other representation formats for semantic map include structured data, e.g. csv files.

A semantic map is obtained through semantic segmentation. The semantic segmentation can be seen as an object detection process combined with a classification task, that is typically carried out by learning-based models. The portion of the semantic map that is characterized by the same label, hence falling under the same semantic class, is called a semantic segmentation mask, and in the case of semantic map images, it is commonly visualised using a specific colour or gray level. An example of an image and corresponding semantic segmentation masks, indicated by the corresponding numeric labels, are shown below.

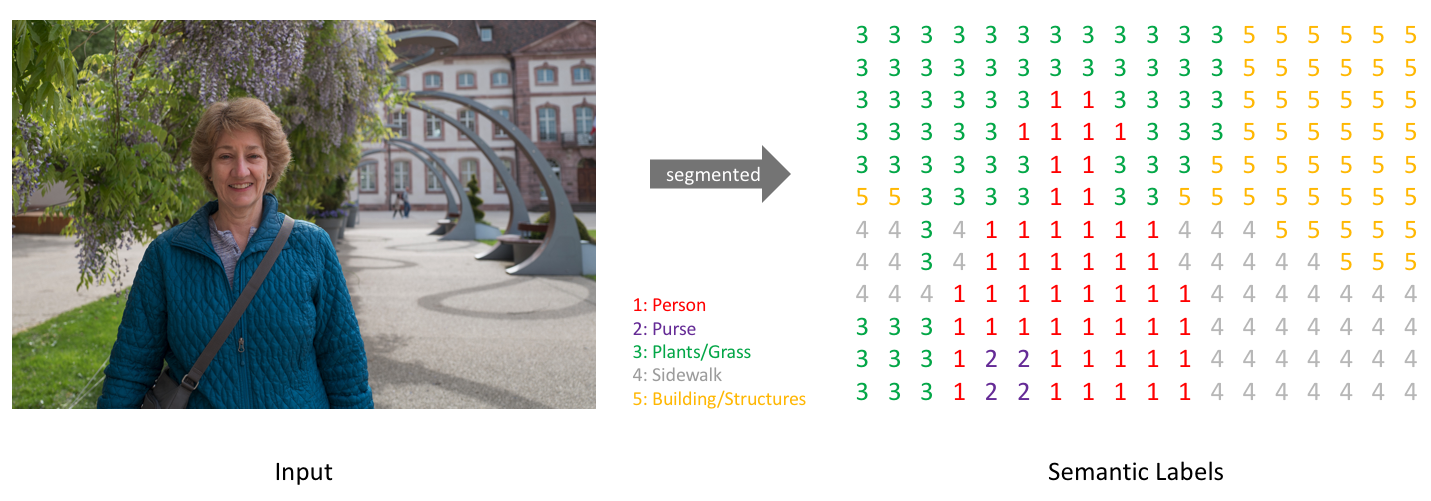


Figure 1: Example of an image and semantic segmentation masks[[5]](#footnote-6).

# Use cases and examples

[Editor’s Note: The sections currently focus on depth and semantics. A similar study of existing technologies needs to be done for alpha maps]

## Depth maps

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.

(6) ***Immersive visualization:*** 3D displays can use RGBD information as input in order to provide immersive visualization experiences to users. As an example, the Looking Glass manufactures 3D displays, for which the RGBD format can be used as input.

(7) ***Cars/Robots Navigation:*** Depth information is needed for navigation of cars/robots. As an example, the Xiaomi Cyberdog makes use of depth sensing technologies.

## Semantic maps

The semantic map information assists computers to understand/interpret visual information. As such, it can assist several applications across a variety of industries, such as eXtended Reality (XR), autonomous driving, medical imaging, robotic vision, industrial inspection, and aerial and satellite imaging, among others, leading to the availability of an increasing amount of frameworks that support and implement corresponding technologies[[6]](#footnote-7),[[7]](#footnote-8),[[8]](#footnote-9). Some examples use cases are provided below.

(1) ***XR applications***: The ability to interpret user’s surroundings is useful for XR applications in a number of tasks including scene understanding, prediction, detection, and recommendations, among others. Recently, the Scene Semantics API[[9]](#footnote-10) was introduced in Android ARCore[[10]](#footnote-11), providing ML model-based, real-time semantic information, and enabling the understanding of the user’s surrounding scene throughout an AR experience. The Scene Semantics API can return a semantic map from the camera feed automatically identifying outdoor scene components, and assigning them corresponding labels, as illustrated in Figure 2.

A screenshot of a video game

Description automatically generated

Figure 2: A frame of a test video[[11]](#footnote-12). The entire video can be accessed here[[12]](#footnote-13).

The Scene Semantics API can additionally return confidence values, as shown in Figure 3.

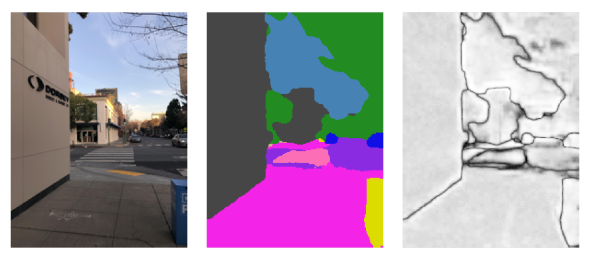


Figure 3: Input image (left), semantic map (middle) and confidence image (right)[[13]](#footnote-14).

The semantic map is an ImageFormat.Y8 image, where each pixel corresponds to a semantic label. Information regarding this image format is provided below[[14]](#footnote-15).

A screenshot of a computer

Description automatically generated

(2) ***Self-driving cars***: Autonomous vehicles must perceive their surroundings, make decisions, and navigate appropriately. Semantic segmentation enables these vehicles to effectively distinguish between various objects. For instance, semantic segmentation can be used to identify pedestrians crossing a road allows the car to determine when it should stop and yield. In another example, semantic segmentation can classify road and lane markings aiding the car in following a designated path. As an example, a challenge on 2D video panoptic segmentation was organized in 2023 by Waymo.[[15]](#footnote-16)

(3) ***Robotics***: To increase their operational efficiency, robots require the capability to map and understand their surrounding scene. Semantic segmentation can assist robots with navigation, interaction, and execution of tasks. NVIDIA, for example, provides an SDK for semantic image segmentation[[16]](#footnote-17).

A person standing in a hallway

Description automatically generated

Figure 4: Isaac ROS image segmentation from NVIDIA.

(4) ***Aerial imaging***: Semantic segmentation for aerial imaging is useful for applications such as urban planning, environmental monitoring, infrastructure development, security and surveillance, enhancing analysis and decision-making.

# 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 focus on depth. A 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. However, this is very specific, since it only targets stereoscopic applications, evident from the definition of “Depth map” (Section 5 of ISO/IEC 23002-3):

**Depth map**

Depth maps are one type of auxiliary video data, addressing stereoscopic video applications. Given an image I and a depth map z, a new image for a new viewpoint on the display can be created by shifting the camera. The stereoscopic view-rendering from 2D+depth map is left open to implementers. Depending on the desired precision and complexity level, it can be done using exact calculations or approximations (some examples are given in the non-normative **Error! Reference source not found.**).

Thus, 23002-3 only provides near and far plane and informative methods for calculating stereo views, and the specification is referenced by ISOBMFF as follows:

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 in existing technologies

Considering the use cases for depth maps presented in section 3 and existing technologies reported in section 4, the following gaps can be observed regarding the carriage of depth data.

*Gap 1*: Existing technologies assume that depth always accompanies a (single) video. However, there are use cases where depth should be able to be stored on its own.

*Gap 2*: Existing technologies assume that depth data should be coded using a video codec. However, there are use cases where carriage of non-video codec coded depth should be supported.

*Gap 3*: Existing technologies assume that depth and texture (i.e. video) are always spatiotemporally aligned. However, there are use cases where depth and texture are not aligned and, thus, alignment information should be signalled.

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

# Discussion

* 1. ***Differences between Alpha and Depth maps***

Some key differences we identified between the two types of maps:

* Alpha maps are meaningful only with the existence of a corresponding colour texture, while this is not the case for depth maps; i.e., ***there are use cases for transmitting just the (captured) depth information, while transparency information is only used paired with the respective colour image.***
* ***Grayscale image representation*** is almost always used for alpha maps, which is *not* the case for depth maps.
* ***The dimensions of the alpha map match the dimensions of the corresponding colour image (even if the resolutions differ)***, while depth maps might have to be aligned.
* ***The alpha map has the same projection as the colour image, while the depth map might not.***
* ***Alpha maps have a fixed value range (from opaque to transparent), while in depth maps the range must be provided.***

# 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 players don’t attempt to play back a depth track when it is video encoded (assumption 4)

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

Additionally, existing features might need to be maintained and 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.

# Terminology

**alpha frame**

a single alpha map, part of an *alpha video*

**alpha map**

the transparency information for the pixels of an image

**alpha video**

a series of *alpha frames* over time

**camera plane**

the default origin plane used as reference for a depth map or depth image (if applicable)

**camera center**

the default origin point used as reference for a depth map or depth image (if applicable)

**depth frame**

a single depth image or depth map, part of a *depth video*

**depth image**

a collection of 3D coordinates (i.e., x,y,z) relative to an origin point (i.e., camera center) or plane (i.e., camera plane)

**depth map**

a collection of distances (e.g., z-values) relative to an origin point (i.e., camera center) or plane (i.e., camera plane), which are expressed as elements of a 2D image or matrix

**depth video**

a series of *depth frames* over time

**point**  
fundamental element of a *point cloud* comprising a position specified as *Cartesian coordinates* and zero or more attributes

**point cloud**

unordered list of *points*

Note: The definitions of the terms “point” and “point cloud” are taken from ISO/IEC 23090-9, Information technology — Coded representation of immersive media — Part 9: Geometry-based point cloud compression.

These terms are *not* defined (nor referenced) at all in ISO/IEC 23090-5, Information technology — Coded representation of immersive media — Part 5: Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC).

**RGB**

the colour elements of an image

**RGBA**

the colour and alpha elements of an image

**RGBD**

the colour and depth elements of an image

**RGBDA**

the colour, depth and alpha elements of an image

# 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

1. also known as range image (typically used for collection of distance between various points – mostly for large-scale imaging) and others [↑](#footnote-ref-2)
2. [↑](#footnote-ref-3)
3. https://developers.google.com/depthmap-metadata [↑](#footnote-ref-4)
4. https://forum.unity.com/threads/artifacts-when-drawing-background-with-dynamically-generated-2d-texture.953336/ [↑](#footnote-ref-5)
5. <https://www.jeremyjordan.me/semantic-segmentation/> [↑](#footnote-ref-6)
6. <https://catalog.ngc.nvidia.com/orgs/nvidia/collections/imagesegmentation> [↑](#footnote-ref-7)
7. <https://www.ibm.com/topics/semantic-segmentation> [↑](#footnote-ref-8)
8. <https://learn.microsoft.com/en-us/azure/machine-learning/component-reference-v2/image-instance-segmentation?view=azureml-api-2> [↑](#footnote-ref-9)
9. <https://developers.google.com/ar/develop/scene-semantics> [↑](#footnote-ref-10)
10. <https://developers.google.com/ar> [↑](#footnote-ref-11)
11. <https://developers.google.com/ar/develop/scene-semantics> [↑](#footnote-ref-12)
12. <https://developers.google.com/static/ar/develop/scene-semantics/videos/sample_use_case.mp4> [↑](#footnote-ref-13)
13. <https://developers.google.com/ar/develop/scene-semantics> [↑](#footnote-ref-14)
14. <https://developer.android.com/reference/android/graphics/ImageFormat#Y8> [↑](#footnote-ref-15)
15. <https://waymo.com/open/challenges/2023/2d-video-panoptic-segmentation> [↑](#footnote-ref-16)
16. <https://nvidia-isaac-ros.github.io/repositories_and_packages/isaac_ros_image_segmentation/index.html> [↑](#footnote-ref-17)