**ISO/IEC JTC 1/SC 29/WG 03 N0626**

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

**Document type:** Output Document

**Title:** Technologies under Consideration for ISO/IEC 23090-6

**Status:** Approved

**Date of document:** 2022-07-26

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

**No. of pages:** 8 (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 N0626**

**July 2022, Virtual**

|  |  |
| --- | --- |
| **Title** | **Technologies under Consideration for ISO/IEC 23090-6** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **21755** |

# Introduction

This document includes technologies under consideration for 23090-6 (Metrics)

This includes the following:

1. Viewpoint mismatch duration metric (m56832)
2. Definition of ViewportDataType and ViewpointDataType (m59969)

# Viewpoint Mismatch Duration Metric

It is asserted that, V3C/OMAFv2 Content could be shown from a different viewpoint (viewer perspective) than its corresponding “content-viewpoint” (e.g. outside the indicated viewing space) even when no viewpoint switching occurs. A metric showing how long this situation persists would be useful as a perceived quality indicator.

A new metric is proposed (m56832) to show the duration that content is shown for a different viewpoint (viewer perspective) that that corresponding “content-viewpoint” (e.g. outside the indicated viewing space).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | | | **Type** | **Description** |
| ViewpointMismatchDuration | | | List | List of Viewpoint mismatch durations. |
|  | *Entry* | | Object | An entry for a Viewpoint mismatch duration. |
|  |  | t | Real-Time | Specifies the measurement time of the viewpoint mismatch duration in wall-clock time. |
|  |  | DeltaViewpoint | ViewpointMismatchType | Delta of the viewpoint between the user position and the viewpoint used for content generation |
|  |  | duration | Integer | Specifies the duration in milliseconds for which the user shows content with a mismatch of the real viewpoint corresponding to the shown content and the viewpoint used for rendering. |

With

|  |  |  |  |
| --- | --- | --- | --- |
| **Key** | | **Type** | **Description** |
| ViewpointMismatchType | | Object | Viewpoint Mismatch. |
|  | vp\_diff\_x | Float | Indicates the difference in x coordinate of the position of the viewpoint in centimetres in the global reference coordinate system, as defined in clause 9.2.2.2 of ISO/IEC 23090-10 compared to the center viewpoint of the corresponding viewport\_id. The values shall be expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754. |
|  | vp\_diff\_y | Float | Indicates the difference in y coordinate of the position of the viewpoint in centimetres in the global reference coordinate system, as defined in clause 9.2.2.2 of ISO/IEC 23090-10 compared to the center viewpoint of the corresponding viewport\_id. The values shall be expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754. |
|  | vp\_diff\_z | Float | Indicates the difference in z coordinate of the position of the viewpoint in centimetres in the global reference coordinate system, as defined in clause 9.2.2.2 of ISO/IEC 23090-10 compared to the center viewpoint of the corresponding viewport\_id. The values shall be expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754. |

# Definition of ViewportDataType and ViewpointDataType

This section provides definitions of ViewportDataType and ViewportDataType, for FDAM1 text of MPEG-I Part 6, "Immersive media metrics for V3C Data and OMAF" [1], in response to the last-meeting’s dispositions of

* the USNB comments, with US 1 and US 6 in particular, in the draft DoC [2], and
* the contribution [3], “Comments on MPEG-I Part 6 Immersive Media Metrics” in the OMAF meeting minutes [4].

The proposed definitions are based on the text from the WD of ISO-IEC 23090-7 AMD1 Common Metadata for Immersive Media [5], and are given in a general form to cover the data types of viewports and viewpoints for both OMAF [7] and V3CD [6] immersive media content.

1. **Definitions of Camera Information [6]**
   1. **Extrinsic Camera Information**
      1. **Syntax**

aligned(8) class ExtCameraInfo () {  
 unsigned int(8)[4] cam\_pos\_x;   
 unsigned int(8)[4] cam\_pos\_y;   
 unsigned int(8)[4] cam\_pos\_z;   
 signed int(32) cam\_quat\_x;   
 signed int(32) cam\_quat\_y;   
 signed int(32) cam\_quat\_z;   
}

* + 1. **Semantics**

cam\_pos\_x, cam\_pos\_y, and cam\_pos\_z, respectively, indicate the x, y, and z coordinates of the position of the camera in metres in the global reference coordinate system. The values shall be expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754.

Cam\_quat\_x, cam\_quat\_y, and cam\_quat\_z, indicate the x, y, and z components, respectively, of the rotation of the camera using the quaternion representation. The values shall be in the range of – 230 to 230, inclusive. When the component of rotation is not present, its value shall be inferred to be equal to 0. The value of rotation components may be calculated as follows:

qX = cam\_quat\_x ÷ 230, qY = cam\_quat\_y ÷ 230, qZ = cam\_quat\_z ÷ 230

The fourth component, qW, for the rotation of the current camera model using the quaternion representation is calculated as follows:

qW = Sqrt( 1 – ( qX2 + qY2 + qZ2 ) )

The point (w, x, y, z) represents a rotation around the axis directed by the vector (x, y, z) by an angle 2\*cos ^{-1}(w)=2\*sin ^{-1}(sqrt(x^{2}+y^{2}+z^{2})).

NOTE – As aligned ISO/IEC FDIS 23090-5, qW is always positive. If a negative qW is desired, one can signal all three syntax elements, cam\_quat\_x, cam\_quat\_y, and cam\_quat\_z with an opposite sign, which is equivalent.

* 1. **Intrinsic Camera Information**
     1. **Syntax**

aligned(8) class IntCameraInfo () {  
 unsigned int(10) camera\_id;   
 bit(3) reserved = 0;   
 unsigned int(3) camera\_type;   
 if (camera\_type == 0) {  
 signed int(32) erp\_horizontal\_fov;   
 signed int(32) erp\_vertical\_fov;   
 }  
 if (camera\_type == 1) {  
 signed int(32) perspective\_horizontal\_fov;   
 unsigned int(8)[4] perspective\_aspect\_ratio;   
 }  
 if (camera\_type == 2) {  
 unsigned int(8)[4] ortho\_aspect\_ratio;   
 unsigned int(8)[4] ortho\_horizontal\_size;   
 }  
 unsigned int(8)[4] clipping\_near\_plane;   
 unsigned int(8)[4] clipping\_far\_plane;   
}

* + 1. **Semantics**

camera\_id is an identifier number that is used to identify a given viewport camera parameters.

Camera\_type indicates the projection method of the viewport camera. The value 0 specifies ERP projection. The value 1 specifies a perspective projection. The value 2 specifies an orthographic projection. Values in the range 3 to 255 are reserved for future use by ISO/IEC.

Erp\_horizontal\_fov specifies the longitude range for an ERP projection corresponding to the horizontal size of the viewport region, in units of radians. The value shall be in the range 0 to 2π.

Erp\_vertical\_fov specifies the latitude range for an ERP projection corresponding to the vertical size of the viewport region, in units of radians. The value shall be in the range 0 to π.

Perspective\_horizontal\_fov specifies the horizontal field of view for perspective projection in radians. The value of shall be in the range of 0 and π.

Perspective\_aspect\_ratio specifies the relative aspect ratio of viewport for perspective projection (horizontal/vertical). The value shall be expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754.

Ortho\_aspect\_ratio specifies the relative aspect ratio of viewport for orthogonal projection (horizontal/vertical). The value shall be expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754.

Ortho\_horizontal\_size specifies the horizontal size of the orthogonal in metres. The value shall be expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754.

Clipping\_near\_plane and clipping\_far\_plane indicate the near and far depths (or distances) based on the near and far clipping planes of the viewport in metres. The values shall be expressed in 32-bit binary floating-point format with the 4 bytes in big-endian order and with the parsing process as specified in IEEE 754.

1. **Suggested Definitions of ViewpointDataType and ViewportDataType**
   1. **ViewpointDataType**

aligned(8) class OmnidirectionalViwepointStruct ()  
{

unsigned int(32) viewpoint\_id;  
 signed int(32) centre\_azimuth;  
 signed int(32) centre\_elevation;  
 signed int(32) centre\_tilt;  
}

aligned(8) class V3cdViewpointStruct ()   
{  
 ExtCameraInfo extCamInfo();  
}

aligned(8) class ViwepointDataType (unsigned int media\_type)  
{  
 if (media\_type == 0) { // omnidireactional media  
 OmnidirectionalViwepointStruct viewpoint();  
 }  
 if (media\_type == 1) { // V3CD media  
 V3cdViwepointStruct viewpoint();  
 }  
}

* 1. **Definition of ViewportDataType**

aligned(8) class OmnidirectionalViweportStruct ()  
{  
 OmnidirectionalViwepointStruct viewpoint ();  
 unsigned int(32) azimuth\_range;  
 unsigned int(32) elevation\_range;  
}

aligned(8) class V3cdViewportStruct (unsigned int ext\_camera\_flag, unsigned int int\_camera\_flag) {  
 if (ext\_camera\_flag == 1) {  
 unsigned int(1) center\_view\_flag;   
 bit(6) reserved = 0;   
 if (center\_view\_flag == 0) {  
 unsigned int(1) left\_view\_flag;   
 } else {  
 bit(1) reserved = 0;   
 }  
 ExtCameraInfo extCamInfo();  
 }  
 if (int\_camera\_flag == 1) {  
 IntCameraInfo intCamInfo();  
 }  
}

aligned(8) class ViweportDataType (unsigned int media\_type, unsigned int ext\_camera\_flag, unsigned int int\_camera\_flag)  
{  
 if (media\_type == 0) { // omnidireactional media  
 OmnidirectionalViweportStruct viewpoint();  
 }  
 if (media\_type == 1) { // V3CD media  
 V3cdViwepointStruct viewport(ext\_camera\_flag, int\_camera\_flag);  
 }  
}

----