**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

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

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 MPEG2018/N17617**

**April 2018, San Diego, USA**

|  |  |
| --- | --- |
| **Source** | Video |
| **Status** | Approved |
| **Title** | Updated Call for Test Materials for 3DoF+ Visual |
| **Editors** | Renaud Doré, Gauthier Lafruit |

# Input content

This document is an update of former N17471 in order to take into account - in addition to ERP cameras - linear perspective cameras, also called pinhole camera or rectilinear as in the document below.

For the investigation on 3DoF+ visual compression [1], video/image material in the following format is requested: Multi-view 360° texture + depth.

They should have the following characteristics:

Capture

* Texture+depth 360° x 180° video/image from two or more simultaneously operating cameras in equirectangular format or in rectilinear, with possibly different viewing positions. By “camera” in this document, one understands either a full capture of the scene from a camera device or a portion of this scene, as explained in [1] fig A and B.
* The divergence of the multiple capture shall be enough in order to describe a 3DoF+ scene with at least 180° azimuth range.
* There are no “holes” meaning that all views together form a consistent set.
* A view shall always have a depth map, except when there is at least one other view with the exact same position that does have a depth map. Therefore, several videos may also be provided with the same viewing position.
* There is no limit in the number of videos. The number of cameras should define the 3DoF+ scene so that the operation of subjective tests is possible.
* A desirable content duration for subjective test should be 10 seconds. Mandatory content duration is 1 frame, e.g just enough to apply objective metrics.
* fps shall be 30 or 60 or 90 fps.

Texture

* Resolution shall be higher or equal to 4096x2024 and lower or equal to 8192×4096 (with exact 2:1 aspect ratio) for the full 360°x180° FoV. If the FoV is reduced, the resolution limits are adapted accordingly, so that angular resolution remains in the same range.
* Each capture can have a reduced Field of View (FoV) below 360° x 180°, eg 180° x 180° or 70° x 60° provided that:
  + Radial distortion is removed from the camera views, with the optical axis passing through the center of the image as in a perfect pinhole camera model (possibly cropping the image to align the center of the image (Width /2, Height / 2) with the optical axis)
  + Camera parameters are therefore simplified:
    - Intrinsics are reduced to a single parameter: a single focal length expressed in pixels
    - Extrinsics are represented by a position vector [x,y,z] and rotation vector [yaw, pitch, roll] as exemplified in section 3
  + Depth maps are available in the same resolution and shape as the texture,
* All test material is progressively scanned and uses 4:2:0 colour sampling with 8 or 10 bits per sample per color component.
* The file format should be Planar YUV (ant not Packed YUV)
* Color primaries, sample range and transfer function used for RGB to YCbCr conversion (and needed for YCbCr to RGB conversion at display side) should follow ITU-R Recommendation BT 709. In particular, limited (219/255) sample range is expected for Y component, and limited (224/255) sample range is expected for Cb and Cr components. Also, the BT 709 non-linear transfer function with 0.45 exponent is expected.
* It is possible to provide content under a form of png or ppm / pgm or tiff, provided that they are convertible from RGB to YUV 4:2:0 exactly like the anchor, through a tool (like deriving from HDRTools) to be identified and made explicit by the content provider. In any case, the contents serving as references are those in YUV 4:2:0 only.
  + For example, a png file provider could ask to use first a PNG-uncompressed TIFF converter:
    - *ffmpeg -vcodec png -i $1.png -compression\_algo raw $1.tiff*
  + Then a TIFF to YUV converter:
    - *HDRConvert -f HDRConvertTiff8ToYuv420.cfg -p SourceFile=$1.tiff -p OutputFile=$1.yuv*

Depth

* When a view has a corresponding depth map, then this depth map shall be provided by default as a raw monochrome stream with a default bit depth of 16-bit, at the resolution of the texture and in little-endian file format. The depth can be delivered with 10 bits only, which is a typical case when output from the360lib software (cf [4]). In this case, this value of 10 should be indicated in the metadata .json file.
* When the format is omnidirectional, the depth is here meant to be the radius from the optical center of the omnidirectional camera.
  + Depth values, in case there are, shall be coded as the normalized disparity, as described in section 3.2 of approved document [2] and adapted for radius dimension instead of z-distance. This requires the definition for each content of Rnear and Rfar values mentioned here below in the metadata section. The content provider is free to put the value Rfar value to infinite, which simplifies the depth relation. By convention, Rfar value put to 1000.0 is understood to mean infinite value.
* When the format is perspective, the depth is here meant to be z coordinate along the optical axis and from the optical center of that perspective camera
  + Depth values, in case there are, shall be coded as the normalized disparity, as described in section 3.2 of approved document [2]. This requires the definition for each content of Znear and Zfar values mentioned here below in the metadata section. The content provider is free to put the value Zfar value to infinite, which simplifies the depth relation. By convention, Zfar value put to 1000.0 is understood to mean infinite value.
* In both previous case, a reserved null value (0) of depth means non-available pixels, thus corresponding to a binary alpha mask channel.

Delivery Packaging

* The default packaging is to have individual images of texture and of depth. Frame range does not have to start by frame index 1 but the sequence shall be continuous.
* There is an option to package textures into a raw texture video and depth into a raw depth video, and this should be indicated in the metadata .json file.
* The name of the texture file shall include for convenience all necessary information useful by a viewer of an elementary image, as described below, where *Camera\_name* can be any name relevant for the content producer, *width* and *height* are image resolution, *nb\_bits* is the number of bits for each color component, and *####* shall be replaced by the frame index.
  + *[Camera name]\_[width]\_[height]\_420\_[nb\_bits] b\_####.yuv*
  + Exemple: *camS1\_2048\_2048\_420\_10b\_1551.yuv*
* The name of the depth file – when present – shall include all necessary information useful by a viewer of an elementary depth image. The fields *Rnear* and *Rfar* are added in the format where integer part is separated from fractional part by “\_”, as illustrated here below where *Rnear* and *Rfar* are 0.5m and 25.0m respectively. Frame index range and camera name shall be aligned with the frame index range and camera or view name of the texture. The suffix is “.depth” for the file name of depth coded on 16bits, and becomes .yuv when the depth is coded on 10 bits. File name examples are given below in both cases:
  + *[Camera name]\_[ width]\_[ height]\_[ Rnear]\_[ Rfar]\_ ####.depth*
  + Exemple: *camS1\_2048\_2048\_0\_5\_25\_0\_1551.depth*
  + *[Camera name]\_[ width]\_[ height]\_[ Rnear]\_[ Rfar]\_420\_10b.yuv*
  + Exemple: *v1\_2048\_2048\_0\_5\_25\_0\_420\_10b.yuv*
* The texture+depth files should be zipped before uploaded to the server, so that it can be conveniently retrieved by one or a couple of zipped files downloads.

Metadata

* Metadata shall be provided under the form of a JSON file that enlists in any order the following properties per video, the properties being the same for all frames of the content, and listed here below
* There is a block of general information:
  + A general name linking this file to a given content
  + The fps of the content (30 / 60 / 90)
  + The total number of frames
  + The center of the bounding box, expressed in OMAF referential
  + An optional informative part can be added for clarity and not used by the 3DoF+ encoder.
* For each camera
  + Camera Name of the file, as used in the file names described here above
  + Video optional field (0: image, 1: video). When this field is not present, textures and depth inputs for that camera are made of images
  + BitDepth optional field specifies the couple of 2 integer [ bit depth for texture, bit depth for depth ]. When absent, the default value is [8,16]
  + Position of the center of the camera as three values [x, y, z] in meters in OMAF referential, as explained in figures 5.3 & 5.4 of [3], Orientation of the related camera [yaw, pitch, roll] expressed in degree and in OMAF referential, as explained in figures 5.3 & 5.4 of [3],
  + If the view has a depth map or not (Boolean 1:true, 0:false),
  + If the view is background or not (Boolean 1:true, 0:false)
  + If so, the *Rnear* and *Rfar* values in meters. This “R” denomination should be understood here as generic: it is either a radius value if format is equirectangular or a z value if format is perspective. The *Rfar* value is permitted to be infinite. When the *Rfar* value is meant to be infinite, it will be arbitrarily written as 1000.0 value.
  + Image/video resolution [width x height]
  + Image/video horizontal and vertical range [Phimin ; Phimax] x [ Thetamin; Thetamax]. Full FoV is [-180; 180] x [-90;90]. These ranges are expressed in the camera referential. This field is only valid for ERP. When this field is present, the camera is understood to be ERP
  + Focal expressed in pixel: this field is only valid for linear perspective camera. When this field is present, the camera is understood to be in linear perspective (pinhole camera).
* Format of real numbers is eee.ffff where eee and ffff are respectively integer and fractional part of any length.

An example of JSON file is given here below.

# Copyright notice

Content owners should provide a copyright notice along with the dataset to inform MPEG about copyright and usage restrictions.

# Informative annex: example of a JSON file

The following file is an example of JSON metadata file with 2 semi-ERP cameras and 1 ERP camera.

*{*

*"Content\_name": "street\_scene",*

*“BoundingBox\_center”:[0.0,0.0,1.65],*

*"Fps": 30,*

*"Frames\_number": 300,*

*"Informative":*

*{*

*"BoundingBox\_size": 0.5*

*},*

*"cameras":*

*[*

*{*

*"Name":" CamS1,”*

*"Position": [0.3, 0.4, 1.65]*

*"Rotation": [60.00, 30.00, 0.00]*

*"Depthmap": 1,*

*"Background": 0,*

*"Rmin": 0.0,*

*"Rmax": 25.0,*

*"Resolution": [2048,2048],*

*"Hor\_range": [-90.0, 90.0],*

*"Ver\_range": [-90.0, 90.0]*

*},*

*{*

*"Name":" CamS2",*

*"Position": [0.3, -0.4, 1.65]*

*"Rotation": [60.00, -30.00, 0.00]*

*"Depthmap": 1,*

*"Background": 0,*

*"Rmin": 0.0,*

*"Rmax": 25.0,*

*"Resolution": [2048,2048],*

*"Hor\_range": [-90.0, 90.0],*

*"Ver\_range": [-90.0, 90.0]*

*},*

*{*

*"Name":" CamO3",*

*"Position": [0.0, 0.0, 1.65]*

*"Rotation": [0.0, 0.0, 0.0]*

*"Depthmap": 0,*

*"Background":0,*

*"Rmin": 0.0,*

*"Rmax": 25.0,*

*"Resolution":[4096,2048],*

*"Hor\_range":[-180.0, 180.0],*

*"Ver\_range":[-90.0, 90.0]*

*}*

*]*

*}*

The following file is an example of JSON metadata file with 3 pinhole / rectilinear cameras with texture and depth coming as 10 bits video

*{*

*"Content\_name": " scene\_shot\_with\_rectilinear\_cameras",*

*“BoundingBox\_center”:[0.0,0.0,1.65],*

*"Fps": 30,*

*"Frames\_number": 300,*

*"Informative":*

*{*

*"BoundingBox\_size": 0.5*

*},*

*"cameras":*

*[*

*{*

*"Name":" V1,”*

*“Video”: 1,*

*“BitDepth”: [ 10, 10],*

*"Position": [0.3, 0.4, 1.65]*

*"Rotation": [-30.00, 0.00, 0.00]*

*"Depthmap": 1,*

*"Background": 0,*

*"Rmin": 0.0,*

*"Rmax": 25.0,*

*"Resolution": [2048,2048],*

*"Hor\_range": [-35.0, 35.0],*

*"Ver\_range": [-30.0, 30.0]*

*“Focal”:1210*

*},*

*{*

*"Name":" V2,”*

*“Video”: 1,*

*“BitDepth”: [ 10, 10],*

*"Position": [0.3, 0.4, 1.65]*

*"Rotation": [0.00, 0.00, 0.00]*

*"Depthmap": 1,*

*"Background": 0,*

*"Rmin": 0.0,*

*"Rmax": 25.0,*

*"Resolution": [2048,2048],*

*"Hor\_range": [-35.0, 35.0],*

*"Ver\_range": [-30.0, 30.0]*

*“Focal”:1210*

*},*

*"Name":" V3,”*

*“Video”: 1,*

*“BitDepth”: [ 10, 10],*

*"Position": [0.3, 0.4, 1.65]*

*"Rotation": [30.00, 0.00, 0.00]*

*"Depthmap": 1,*

*"Background": 0,*

*"Rmin": 0.0,*

*"Rmax": 25.0,*

*"Resolution": [2048,2048],*

*"Hor\_range": [-35.0, 35.0],*

*"Ver\_range": [-30.0, 30.0]*

*“Focal”:1210*

*}*

*}*

# References

[1] N17612 Investigation of 3DoF+ Video

[2] N16730 Depth map formats used within MPEG 3D technologies

[3] W17399 Revised text FDIS 23090-2 OMAF clean.doc Version 2.2;

[4] N17618 Common Test Conditions for 3DoF+