INTERNATIONAL STANDARD

ISO/IEC 19566-6

First edition
2019-07

AMENDMENT 1
2021-06

Information technologies — JPEG systems —

Part 6: JPEG 360

AMENDMENT 1: Addition of new JPEG 360 image types and accelerated ROI rendering

Technologies de l'information — Systèmes JPEG JPEG 360 —

Partie 6: JPEG 360

AMENDEMENT 1

Reference number
ISO/IEC 19566-6:2019/Amd.1:2021(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO/IEC 19566 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Information technologies — JPEG systems —

Part 6: JPEG 360

AMENDMENT 1: Addition of new JPEG 360 image types and accelerated ROI rendering

3.2

Add the following abbreviations terms:

FOV field of view

URI uniform resource identifier

5.3

Add a new paragraph at the end of clause 5.3 as follows:

In addition, an accelerated viewport rendering to support an efficient and low latency viewport transmission shall be done as defined in Annex C.

5.4

Renumber subclause 5.4 as 5.5 and add a new subclause 5.4 as follows:

5.4 Stereoscopic 360 image

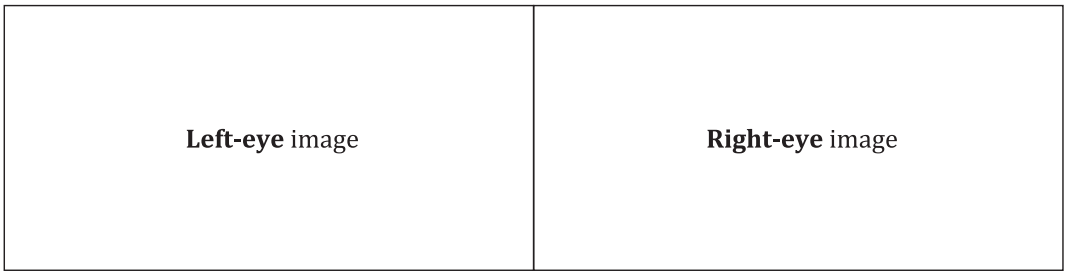
5.4.1 General

A stereoscopic 360 image is a pair that consists of two images of the same scene or object but a slightly different angle or perspective for the left and right eye. Each pair should be an image with an ERP format. This pair is stored in three formats for JPEG 360 in addition to an existing monoscopic format which provides the only single view.

5.4.2 Stereoscopic formats

The stereoscopic format indicates an arrangement of the left and right images. This subclause defines three formats for organizing the pair of the stereoscopic 360 images as follows.

— Side-by-side: left- and right-eye images are packed horizontally.

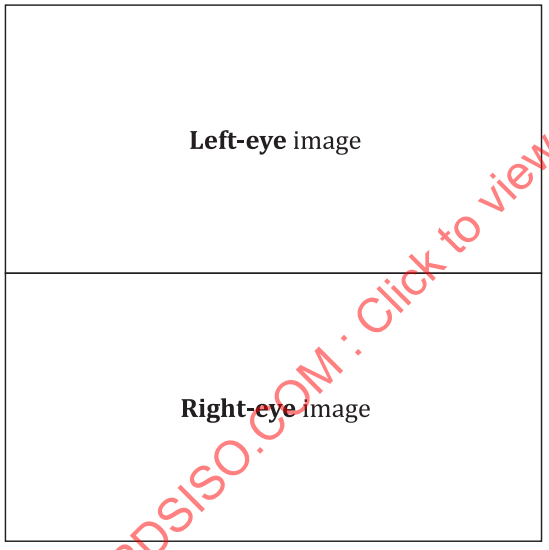


a) Structure

b) Example of the format

Figure 11 — Side-by-side format

— Top-bottom: left- and right-eye images are packed vertically.



a) Structure

b) Example of the format

Figure 12 — Top-bottom format

— Extended: one of the left- and right-eye images is stored in the form of a JPEG image, which contains the metadata required to construct a stereoscopic scene. The metadata includes information that identifies the location of the remaining image in both eyes. Figure 13 shows an example of the extended format. Although this example shows the right-eye image is contained in the same JPEG image file, the right-eye image may be referenced externally as a different JPEG image file, as per ISO/IEC 19566-5.

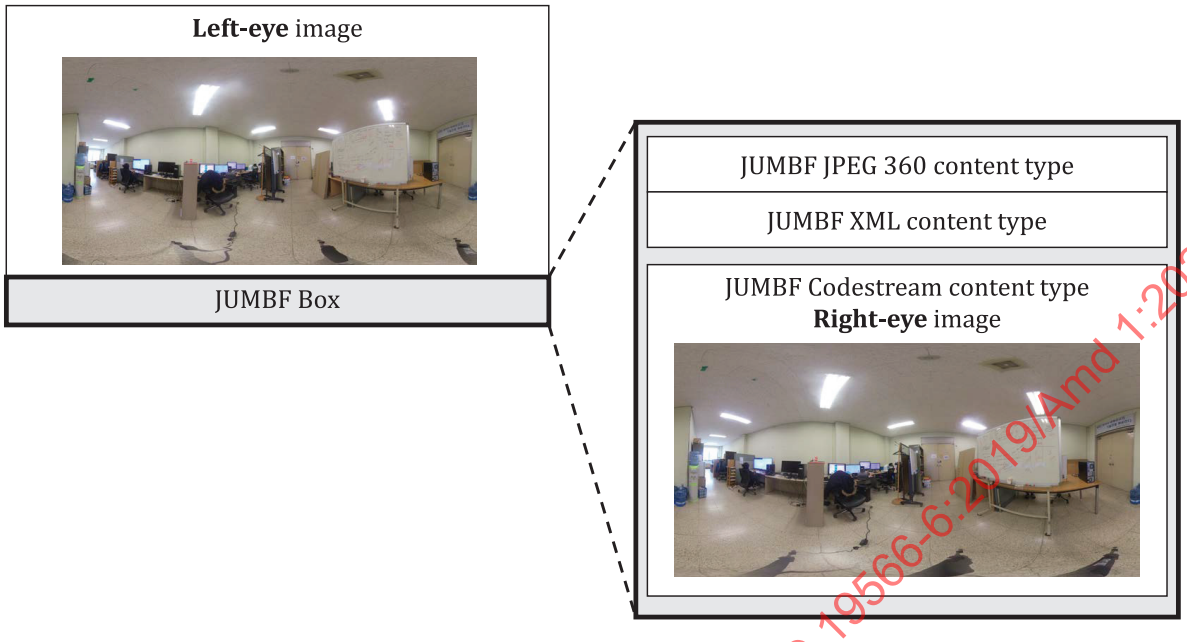
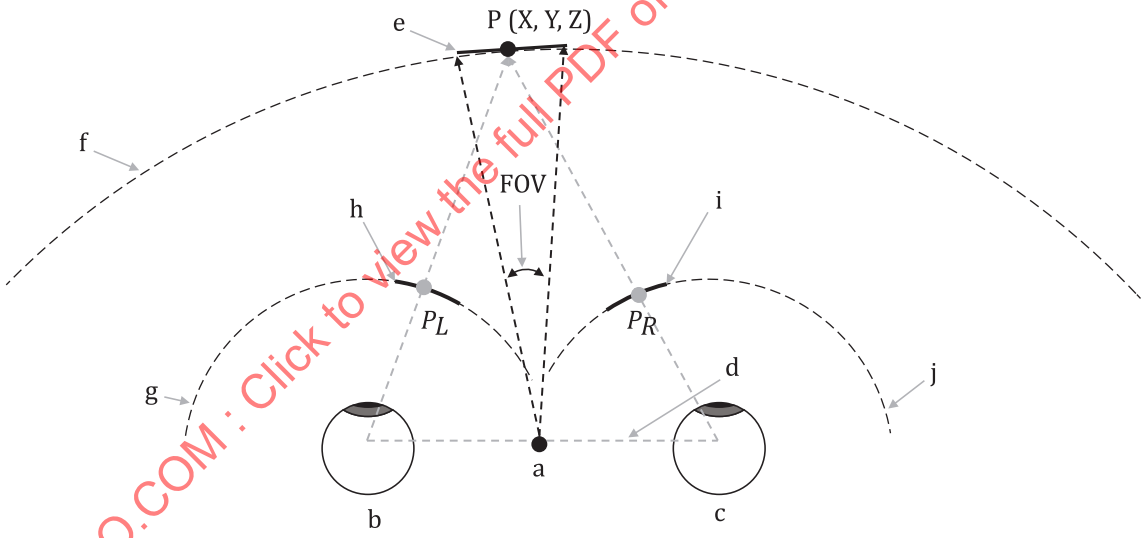


Figure 13 — Example of the extended format

5.4.3 Viewport definition in the stereo vision model



Key

a	centre of the baseline	f	sphere
b	left eye	g	left 360-degree image
c	right eye	h	viewport mapped on the left image
d	baseline	i	viewport mapped on the right image
e	viewport	j	right 360-degree image

Figure 14 — Viewport definition in the stereo vision model

Figure 14 illustrates the stereo vision model used for stereoscopic 360 images and how a viewport is organized. The viewport is a rectangular sub-region of the sphere specified by the FOV span values from the centre of the sphere that is supposed to be the centre of the two eyes in the model.

In this model, P, which is a point on the sphere with Cartesian coordinate, is mapped on to the left and right images, P_L and P_R, differently depending on the configuration of the capture devices. This

allows the human visual system to perceive the point P more realistically when left and right images are projected on both eyes, respectively. In this case, the viewports for both eyes are not necessarily defined for the left and right eyes separately but can be defined by the viewing direction from the centre of both eyes. Therefore, the viewport is specified by the FOV span values at the centre of the sphere that is considered as the centre of the two eyes in the model.

5.4.4 Compatible with a conventional 360 image viewer

An interaction modality, such as a head-mounted display, is able to present a stereoscopic scene while a conventional 360 image viewer may parse only monoscopic JPEG 360 image and present a viewport. However, when the conventional viewer loads the stereoscopic JPEG 360 image file, it is possible to extract the left- or right-eye image from the file based on the metadata embedded in the file then present it to the viewer.

Furthermore, when the stereoscopic JPEG 360 image is an extended format, the conventional viewer will display one of the left- and right-eye images without additional processing. This follows the backward compatible mechanism, as described in subclause 5.2, which allows that an image is decoded when the file is opened by conventional JPEG viewing applications.

6.1

Change the number of Figure 11 to 15.

B.2

Replace the meaning of the BoxReference in Table B.1 with the following:

Refers to a label for either a JUMBF Codestream Content Type box or a JUMBF Embedded File Content Type box, as per ISO/IEC 19566-5.

B.7

Replace the XMP expression with the following:

```
<?xpacket begin="ï»¿" id="W5M0MpCehiHzreSzNTczkc9d"?>
  <x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="XMP Core 5.5.0">
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
      <rdf:Description rdf:about=""
          xmlns:xmp="http://ns.adobe.com/xap/1.0/"
          xmlns:umf="http://ns.intel.com/umf/2.0">
        <umf:next-id>0</umf:next-id>
        <umf:schemas>
          <rdf:Bag>
            <rdf:li rdf:parseType="Resource">
              <!-- JPEG 360 Metadata -->
              <umf:schema>JPEG360Metadata</umf:schema>
                <umf:descriptors>
                  <rdf:Bag>
                    <rdf:li rdf:parseType="Resource">
                      <umf:name>JPEG360ImageMetadata</umf:name>
                      <umf:fields>
                        <rdf:Bag>
                          <rdf:li rdf:parseType="Resource">
                            <umf:name>JPEG360Version</umf:name>
                            <umf:type>integer</umf:type>
                          </rdf:li>
                          <rdf:li rdf:parseType="Resource">
                            <umf:name>MediaType</umf:name>
                            <umf:type>string</umf:type>
                          </rdf:li>
```



```
          <rdf:li rdf:parseType="Resource">
            <umf:name>ProjectionType</umf:name>
            <umf:type>string</umf:type>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <umf:name>PhiMin</umf:name>
            <umf:type>real</umf:type>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <umf:name>PhiMax</umf:name>
            <umf:type>real</umf:type>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <umf:name>ThetaMax</umf:name>
            <umf:type>real</umf:type>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <umf:name>ThetaMin</umf:name>
            <umf:type>real</umf:type>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <umf:name>PhiGravity</umf:name>
            <umf:type>real</umf:type>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <umf:name>ThetaGravity</umf:name>
            <umf:type>real</umf:type>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <umf:name>CompassPhi</umf:name>
            <umf:type>real</umf:type>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <umf:name>BoxReference</umf:name>
            <umf:type>string</umf:type>
          </rdf:li>
        </rdf:Bag>
      </umf:fields>
    </rdf:li>
    <rdf:li rdf:parseType="Resource">
      <umf:name>JPEG360ViewportMetadata</umf:name>
      <umf:fields>
        <rdf:Bag>
          <rdf:li rdf:parseType="Resource">
            <umf:name>JPEG360ViewportNumber</umf:name>
            <umf:type>integer</umf:type>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <umf:name>ViewportPhi</umf:name>
            <umf:type>real</umf:type>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <umf:name>ViewportTheta</umf:name>
            <umf:type>real</umf:type>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <umf:name>ViewportPhiFOV</umf:name>
            <umf:type>real</umf:type>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <umf:name>ViewportThetaFOV</umf:name>
            <umf:type>real</umf:type>
          </rdf:li>
          <rdf:li rdf:parseType="Resource">
            <umf:name>ViewportRoll</umf:name>
            <umf:type>real</umf:type>
          </rdf:li>
        </rdf:Bag>
      </umf:fields>
    </rdf:li>
  </rdf:Bag>
```

```
                </umf:descriptors>
              </rdf:li>
            </rdf:Bag>
          </umf:schemas>

          <umf:metadata>
          <rdf:Bag>
            <rdf:li rdf:parseType="Resource">
              <umf:schema>JPEG360Metadata</umf:schema>
              <umf:set>
              </umf:set>
            </rdf:li>
          </rdf:Bag>
        </umf:metadata>

      </rdf:Description>
    </rdf:RDF>

  </x:xmpmeta>
<?xpacket end="w"?>
```

B.9

Replace the XMP expression with the following:

```
<?xpacket begin="ï»¿" id="W5M0MpCehiHzreSzNTczkc9d"?>
  <x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="XMP Core 5.5.0">
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <rdf:Description rdf:about=""
      xmlns:xmp="http://ns.adobe.com/xap/1.0/" mlns:umf="http://ns.intel.com/umf/2.0">
      <umf:next-id>3</umf:next-id>
      <umf:schemas>
        <rdf:Bag>
        <rdf:li rdf:parseType="Resource">
        <!-- JPEG 360 Metadata -->
          <umf:schema>JPEG360Metadata</umf:schema>
          <umf:descriptors>
            <rdf:Bag>
              <rdf:li rdf:parseType="Resource">
                <umf:name>JPEG360ImageMetadata</umf:name>
                  <umf:fields>
                    <rdf:Bag>
                      <rdf:li rdf:parseType="Resource">
                        <umf:name>JPEG360Version</umf:name>
                        <umf:type>integer</umf:type>
                      </rdf:li>
                      <rdf:li rdf:parseType="Resource">
                        <umf:name>MediaType</umf:name>
                        <umf:type>string</umf:type>
                      </rdf:li>
                      <rdf:li rdf:parseType="Resource">
                        <umf:name>ProjectionType</umf:name>
                        <umf:type>string</umf:type>
                      </rdf:li>
                      <rdf:li rdf:parseType="Resource">
                        <umf:name>PhiMin</umf:name>
                        <umf:type>real</umf:type>
                      </rdf:li>
                      <rdf:li rdf:parseType="Resource">
                        <umf:name>PhiMax</umf:name>
                        <umf:type>real</umf:type>
                      </rdf:li>
                      <rdf:li rdf:parseType="Resource">
                        <umf:name>ThetaMax</umf:name>
                        <umf:type>real</umf:type>
                      </rdf:li>
                      <rdf:li rdf:parseType="Resource">
                        <umf:name>ThetaMin</umf:name>
                        <umf:type>real</umf:type>
                      </rdf:li>
```



```
            <rdf:li rdf:parseType="Resource">
              <umf:name>PhiGravity</umf:name>
              <umf:type>real</umf:type>
            </rdf:li>
            <rdf:li rdf:parseType="Resource">
              <umf:name>ThetaGravity</umf:name>
              <umf:type>real</umf:type>
            </rdf:li>
            <rdf:li rdf:parseType="Resource">
              <umf:name>CompassPhi</umf:name>
              <umf:type>real</umf:type>
            </rdf:li>
            <rdf:li rdf:parseType="Resource">
              <umf:name>BoxReference</umf:name>
              <umf:type>string</umf:type>
            </rdf:li>
          </rdf:Bag>
        </umf:fields>
      </rdf:li>
      <rdf:li rdf:parseType="Resource">
        <umf:name>JPEG360ViewportMetadata</umf:name>
        <umf:fields>
          <rdf:Bag>
            <rdf:li rdf:parseType="Resource">
              <umf:name>JPEG360ViewportNumber</umf:name>
              <umf:type>integer</umf:type>
            </rdf:li>
            <rdf:li rdf:parseType="Resource">
              <umf:name>ViewportPhi</umf:name>
              <umf:type>real</umf:type>
            </rdf:li>
            <rdf:li rdf:parseType="Resource">
              <umf:name>ViewportTheta</umf:name>
              <umf:type>real</umf:type>
            </rdf:li>
            <rdf:li rdf:parseType="Resource">
              <umf:name>ViewportPhiFOV</umf:name>
              <umf:type>real</umf:type>
            </rdf:li>
            <rdf:li rdf:parseType="Resource">
              <umf:name>ViewportThetaFOV</umf:name>
              <umf:type>real</umf:type>
            </rdf:li>
            <rdf:li rdf:parseType="Resource">
              <umf:name>ViewportRoll</umf:name>
                <umf:type>real</umf:type>
            </rdf:li>
          </rdf:Bag>
        </umf:fields>
      </rdf:li>
    </rdf:Bag>
  </umf:descriptors>
 </rdf:li>
</rdf:Bag>
</umf:schemas>

<umf:metadata>
  <rdf:Bag>
    <rdf:li rdf:parseType="Resource">
      <umf:schema>JPEG360Metadata</umf:schema>
        <umf:set>
          <rdf:Bag>
            <rdf:li rdf:parseType="Resource">
              <umf:name>JPEG360ImageMetadata</umf:name>
              <umf:set>
                <rdf:Bag>
                  <rdf:li rdf:parseType="Resource">
                    <umf:id>3</umf:id>
                    <umf:fields>
                      <rdf:Bag>
                        <rdf:li rdf:parseType="Resource">
```

```
                <rdf:value>1</rdf:value>
                <umf:name>JPEG360Version</umf:name>
              </rdf:li>
              <rdf:li rdf:parseType="Resource">
                <rdf:value> image/jpeg</rdf:value>
                <umf:name>MediaType</umf:name> </rdf:li>
              <rdf:li rdf:parseType="Resource">
                <rdf:value>Equirectangular</rdf:value>
                <umf:name>ProjectionType</umf:name> </rdf:li>
              <rdf:li rdf:parseType="Resource">
                <rdf:value>-180</rdf:value>
                <umf:name>PhiMin</umf:name> </rdf:li>
              <rdf:li rdf:parseType="Resource">
                <rdf:value>180</rdf:value>
                <umf:name>PhiMax</umf:name> </rdf:li>
              <rdf:li rdf:parseType="Resource">
                <rdf:value>90</rdf:value>
                <umf:name>ThetaMax</umf:name> </rdf:li>
              <rdf:li rdf:parseType="Resource">
                <rdf:value>-90</rdf:value>
                <umf:name>ThetaMin</umf:name> </rdf:li>
              <rdf:li rdf:parseType="Resource">
                <rdf:value>0</rdf:value>
                <umf:name>PhiGravity</umf:name> </rdf:li>
              <rdf:li rdf:parseType="Resource">
                <rdf:value>-90</rdf:value>
                <umf:name>ThetaGravity</umf:name> </rdf:li>
              <rdf:li rdf:parseType="Resource">
                <rdf:value>0</rdf:value>
                <umf:name>CompassPhi</umf:name>
              </rdf:li>
              <rdf:li rdf:parseType="Resource">
                <rdf:value>conventional</rdf:value>
                <umf:name>BoxReference</umf:name>
              </rdf:li>
            </rdf:Bag>
          </umf:fields>
        </rdf:li>
      </rdf:Bag>
    </umf:set>
  </rdf:li>
  <rdf:li rdf:parseType="Resource">
    <umf:name>JPEG360ViewportMetadata</umf:name>
    <umf:set>
      <rdf:Bag>
        <rdf:li rdf:parseType="Resource">
          <umf:id>1</umf:id>
          <umf:fields>
            <rdf:Bag>
              <rdf:li rdf:parseType="Resource">
                <rdf:value>0</rdf:value>
                <umf:name>JPEG360ViewportNumber</umf:name>
              </rdf:li>
              <rdf:li rdf:parseType="Resource">
                <rdf:value>0</rdf:value>
                <umf:name>ViewportPhi</umf:name>
              </rdf:li>
              <rdf:li rdf:parseType="Resource">
                <rdf:value>0</rdf:value>
                <umf:name>ViewportTheta</umf:name>
              </rdf:li>
              <rdf:li rdf:parseType="Resource">
                <rdf:value>100</rdf:value>
                <umf:name>ViewportPhiFOV</umf:name>
              </rdf:li>
              <rdf:li rdf:parseType="Resource">
                <rdf:value>75</rdf:value>
                <umf:name>ViewportThetaFOV</umf:name>
              </rdf:li>
              <rdf:li rdf:parseType="Resource">
                <rdf:value>0</rdf:value>
```

```
                    <umf:name>ViewportRoll</umf:name>
                  </rdf:li>
                </rdf:Bag>
              </umf:fields>
            </rdf:li>
            <rdf:li rdf:parseType="Resource">
              <umf:id>2</umf:id>
              <umf:fields>
                <rdf:Bag>
                  <rdf:li rdf:parseType="Resource">
                    <rdf:value>1</rdf:value>
                    <umf:name>JPEG360ViewportNumber</umf:name>
                  </rdf:li>
                  <rdf:li rdf:parseType="Resource">
                    <rdf:value>45</rdf:value>
                    <umf:name>ViewportPhi</umf:name>
                  </rdf:li>
                  <rdf:li rdf:parseType="Resource">
                    <rdf:value>45</rdf:value>
                    <umf:name>ViewportTheta</umf:name>
                  </rdf:li>
                  <rdf:li rdf:parseType="Resource">
                    <rdf:value>45</rdf:value>
                    <umf:name>ViewportPhiFOV</umf:name>
                  </rdf:li>
                  <rdf:li rdf:parseType="Resource">
                    <rdf:value>45</rdf:value>
                    <umf:name>ViewportThetaFOV</umf:name>
                  </rdf:li>
                  <rdf:li rdf:parseType="Resource">
                    <rdf:value>15</rdf:value>
                    <umf:name>ViewportRoll</umf:name>
                  </rdf:li>
                </rdf:Bag>
              </umf:fields>
            </rdf:li>
          </rdf:Bag>
        </umf:set>
      </rdf:li>
    </rdf:Bag>
  </umf:set>
</rdf:li>
</rdf:Bag>
</umf:metadata>
</rdf:Description>
</rdf:RDF>
</x:xmpmeta>
<?xpacket end="w"?>
```

Renumber Clause B.9 as B.11 and add new Clauses B.9 and B.10 as follows:

B.9 Versions of JPEG 360 metadata

This document provides an efficient way to exchange JPEG 360 files by defining several versions based on the depth of the features as listed in Table B.5.

Table B.5 — Supported metadata element according to the JPEG360Version

JPEG360Version	Supported metadata elements
1	This version supports monoscopic 360-degree image only.
2	This version supports monoscopic and stereoscopic 360-degree image. The StereoscopicFormat metadata element in the JPEG360ImageMetadata is an optional element, which indicates an arrangement of the left and right images for the stereoscopic 360-degree images. If the StereoscopicFormat metadata element is absent, the main image is a monoscopic image. When the StereoscopicFormat is set to 'Extended,' the BoxReference metadata element in the JPEG360ImageMetadata refers to a label for either a JUMBF Codestream Content Type box or a JUMBF Embedded File Content Type box, as per ISO/IEC 19566-5.
All other values	Reserved for ISO/IEC.

B.10 Required JPEG 360 image values for stereoscopic format

Table B.6 provides the required values of metadata for stereoscopic JPEG 360 image.

Table B.6 — Required metadata element values for stereoscopic format

JPEG 360 schema metadata elements		Default value when no metadata elements present
JPEG360Metadata		Not applicable.
JPEG360ImageMetadata	—	Not applicable.
	JPEG360Version	2
	MediaType	Shall be provided.
	ProjectionType	Equirectangular.
	StereoscopicFormat	Shall be as either 'side-by-side', 'top-bottom', or 'extended'.
	PhiMin	Shall be provided.
	PhiMax	Shall be provided.
	ThetaMax	Shall be provided.
	ThetaMin	Shall be provided.
	PhiGravity	Shall be provided.
	ThetaGravity	Shall be provided.
	CompassPhi	Shall be provided.
	BoxReference	Shall be provided. Note: 'Conventional' shall be provided when *StereoscopicFormat* is set to 'side-by-side' and 'top-bottom.' When the *StereoscopicFormat* is set to 'extended', a label shall be provided for refers to a label for either JUMBF Codestream Content Type box or a JUMBF Embedded File Content Type box, as per ISO/IEC 19566-5.
JPEG360ViewportMetadata	—	Not applicable
	JPEG360ViewportNumber	Shall be provided.
	ViewportPhi	Shall be provided.
	ViewportTheta	Shall be provided.
	ViewportPhiFOV	Shall be provided.
	ViewportThetaFOV	Shall be provided.
	ViewportRoll	Shall be provided.

Annex B

After Annex B, add the following new annex:

Annex C
(normative)

Accelerated ROIs rendering for JPEG (ISO/IEC 10918) images

C.1 General

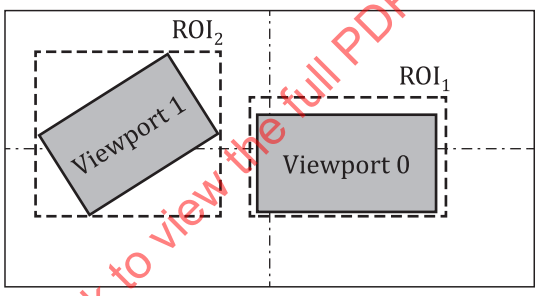
This annex defines a region of interest (ROI) representation and a mechanism to declare an order of ROIs for the progressively accelerated viewport rendering. This mechanism supports use cases defined in JPEG 360 use cases and requirements: An efficient and low latency viewport transmission and guided viewport experience.

Currently, this mechanism is defined for the monoscopic 360 image format.

C.2 ROI (region of interest) representation

C.2.1 Definition of ROI

The ROI is a region in the projected image containing one or more viewports as depicted in Figure C.1. A viewport representation is specified in subclause 5.3.



NOTE The ROI is presented by a dashed blue rectangle.

Figure C.1 — Example of the definition of ROIs

In Figure C.1, two ROIs are defined for the default viewport and one additional viewport. In this example, JPEG 360 decoder will show initially the region that is defined as a default viewport then change the scene to the next viewport if there is no a user interaction for navigation.

C.2.2 Construction of ROIs

To support the accelerated ROIs rendering, JPEG 360 encoder shall construct ROIs which contains viewports as follow steps:

— Split a 360 image into N×M squared blocks. For example, an ERP image that is encoded image using ISO/IEC 10918-1 can be divided into 16×16 sized block and 8×8 sized blocks for 4:2:0 and 4:4:4 subsampling, respectively.

— Specify ROIs as a collection of blocks with a minimum size that completely contains the viewport as described in Figure C.2.

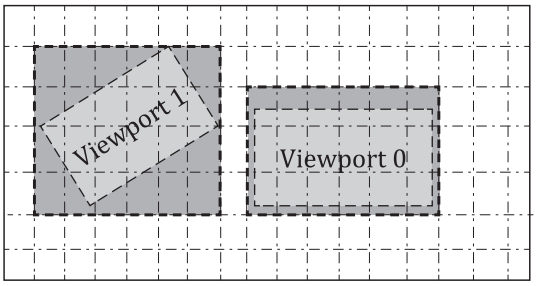


Figure C.2 — Example of the constructing ROIs

C.3 Arrangement of multiple ROIs for accelerated rendering of the viewport

C.3.1 General

This clause defines methods for arrangement of multiple ROIs in two ways: extended and centre-to-edge.

C.3.2 Extended: Backward compatible method

JPEG 360 allows that an image is decoded when the image is loaded by the conventional decoder. This backward compatible feature enables the accelerated rendering of the viewport. ROIs and remaining parts of the image are encoded as an individual image. Then the ROI image, which contains a default viewport, is placed in the conventional position and additional JUMBF boxes containing other images are embedded into the JPEG 360 file as shown in Figure C.3.

When a JPEG 360 image viewer loads the image, the viewer decodes the first ROI image and renders it quickly because the first ROI image size is smaller than the entire image size. While the viewer is showing the default viewport, additional images embedded in the JUMBF boxes are extracted, decoded, and then rendered gradually using metadata in the file.

For example, as depicted in Figures C.1 and C.2, ROI_1, ROI_2, and remaining parts of the images are encoded as an individual image. The default viewport may be decoded as quick as possible when the first ROI image is available to the decoding process. The second ROI containing a second viewport is loaded and decoded while the first viewport image is rendered to the viewer. Then, finally, after decoding the third image the whole scene will be constructed and rendered gradually. ROI_2 and Out of ROIs images can be JPEG files referenced externally, as per ISO/IEC 19566-5.

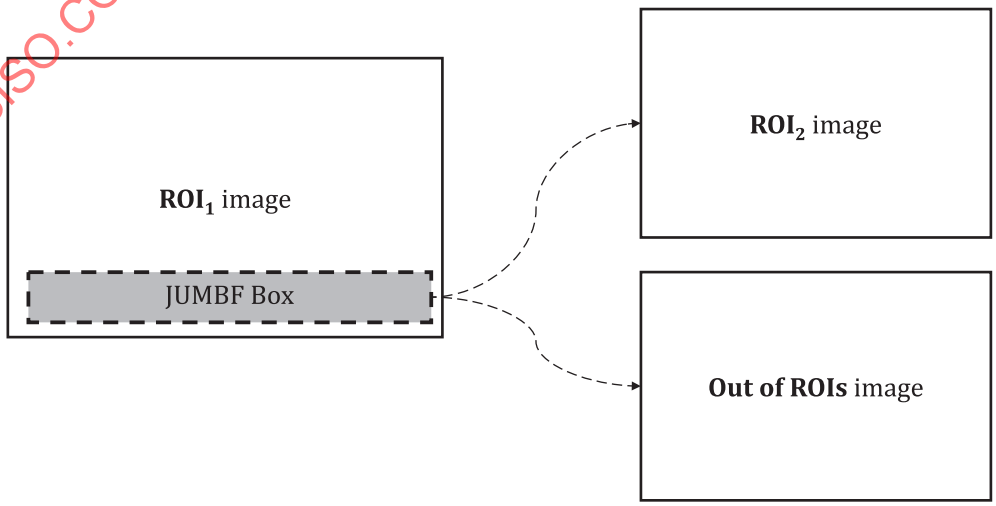


Figure C.3 — Example of the extended arrangement

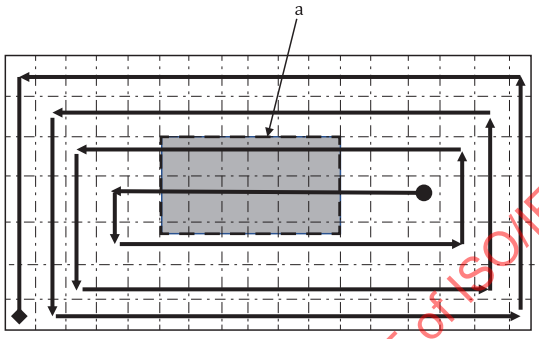
C.3.3 Centre-to-edge reordering method

C.3.3.1 *General*

JPEG 360 allows the reordering blocks in a centre-to-edge (CTE) order for accelerating the viewport rendering. The reordering is performed counter-clockwise as shown in Figure C.4. Figure C.4 and C.5 illustrate two modes of the CTE reordering: global and local. These are modes encoding the default viewport first and extending the encoding regions to the edge in a different way.

C.3.3.2 *Global CTE reordering*

This mode extends the encoding region to the edge in a spiral order globally. The order starts from the block inside of the image and ends at the edge of the image as depicted in Figure C.4. This may be a recommended mode in case no JPEG360ViewportMetadata is specified.



Key

a Default viewport.

Figure C.4 — Example of the global CTE reordering

C.3.3.3 *Local CTE reordering*

This mode extends the encoding region according to the ROI order, and CTE is performed sequentially for each ROI as illustrated in Figure C.5. After all ROIs are processed, the remaining blocks not included in the ROIs are reordered in a same order as the global mode while skipping blocks included in the ROIs.

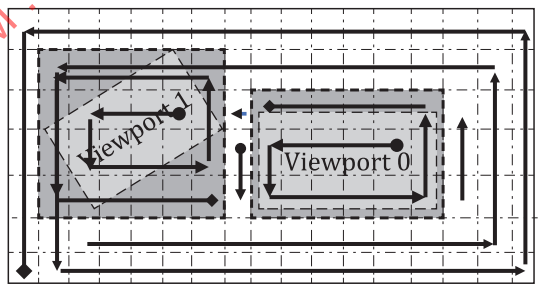


Figure C.5 — Example of the local CTE reordering

JPEG 360 viewing application should decode the CTE-ordered image data block-wisely in a conventional scan order (left-to-right and top-to-bottom), perform an inverse CTE, and construct a default viewport using metadata in the file.

All blocks belonging to the first ROI are decoded quickly and rendered to the viewer when the reconstruction of the viewport is complete. While the default viewport is rendering, consecutive blocks are decoded and inversed gradually to reconstruct the entire 360-degree image.

C.4 Metadata for accelerated ROI rendering

This clause defines metadata for the accelerated ROI rendering. JPEG360AcceleratedROI is an element that contains the following information about the ROI:

— a number for the accelerated ROI metadata;

— the size of the block;

— ROI information;

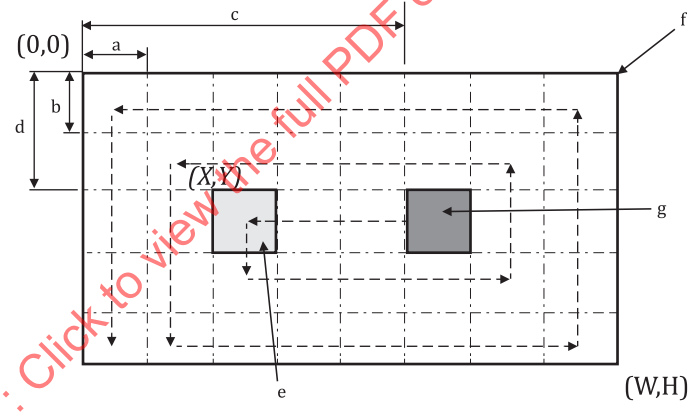
— information of the CTE reordering;

— indication of the associated viewport metadata;

— reference of the associated ROI image.

As defined in Figure C.6 and Table C.2, each ROI is defined through a separate JPEG360AcceleratedROI element. The ROI metadata is identified using a ROINumber. In general, the ROINumber is a value greater than 0. However, when the ROINumber is set to 0, the metadata is for global CTE reordering, and when there is another ROI having the number with larger than 0, it is metadata for the remaining blocks that do not belong to the ROI.

When the CTE reordering method is used, a starting block of the ROI is different based on image resolution and block size. JPEG360AcceleratedROI signals the reordering way for a ROI as shown in Figure C.6.



a) Reordering when W is greater or equal to H

Key

a Block width.

b Block height.

c Offset X.

d Offset Y.

e Turning block.

f ROI.

g Starting block.