INTERNATIONAL STANDARD

**ISO/IEC 23488**

First edition
2022-05

# Information technology — Computer graphics, image processing and environment data representation — Object/environmental representation for image-based rendering in virtual/mixed and augmented reality (VR/MAR)

*Technologies de l'information — Infographie, traitement d'images et représentation des données environnementales — Représentation d'objets/environnements pour l'habillage à partir d'images réelles dans la réalité virtuelle/mixte et augmentée (VR/MAR)*

Reference number
ISO/IEC 23488:2022(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

# Contents

Page

# 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 https://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 24, *Computer graphics, image processing and environmental data representation*.

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.

# Introduction

As virtual reality (VR) and augmented reality (AR) expand to applications in entertainment and education industries, many methods of augmenting reality to virtual space have been developed. Because of this expansion, the technology of capturing and representing objects in real environments is in high demand.

One of the proposed methods of capturing the real world is image-based representation. Image-based representation is a technique that can be used in various applications that require 3D model rendering at an arbitrary viewpoint, including virtual reality, augmented reality and video stabilization. Since image-based representation is a predominant alternative to using 3D models in the growing VR/MAR market, due to its realism, scalability, accuracy and efficiency, creating a standard for image-based representation is required.

# Information technology — Computer graphics, image processing and environment data representation — Object/environmental representation for image-based rendering in virtual/mixed and augmented reality (VR/MAR)

## 1 Scope

This document specifies an image-based representation model that represents target objects/environments using a set of images and optionally the underlying 3D model for accurate and efficient objects/environments representation at an arbitrary viewpoint. It is applicable to a wide range of graphic, virtual reality and mixed reality applications which require the method of representing a scene with various objects and environments.

This document:

— defines terms for image-based representation and 3D reconstruction techniques;

— specifies the required elements for image-based representation;

— specifies a method of representing the real world in the virtual space based on image-based representation;

— specifies how visible image patches can be integrated with the underlying 3D model for more accurate and rich objects/environments representation from arbitrary viewpoints;

— specifies how the proposed model allows multi-object representation;

— provides an XML based specification of the proposed representation model and an actual implementation example (see Annex A).

## 2 Normative references

There are no normative references in this document.

## 3 Terms and definitions and abbreviated terms

### 3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

**3.1.1**
**camera parameter**
extrinsic/external and intrinsic/internal attribute value of a (virtual or real) camera that describes the mathematical relationship between the 3D coordinates of a point in the world space and the 2D coordinates of its projection onto the image space

**3.1.2**
**image patch**
set of connected pixels of an image that has the same visibility information

Note 1 to entry: Image patch is synonymous to image.

**3.1.3**
**tessellation**
process of dividing a face into sub-faces using sub-sampled vertices to attain enhanced visibility detection

**3.1.4**
**viewpoint**
rotation and position with respect to the reference coordinate system that determines visible parts of a 3D model in a representation system

Note 1 to entry: "Viewpoint" is different from "Camera (parameters)" in that it only includes the camera position and pose, where as the "Camera" has more information such as the internal *camera parameters* (3.1.1) including the focal length, field of view, near/far planes and skew parameters.

**3.1.5**
**virtual image**
image at an arbitrary *viewpoint* (3.1.4) that is generated by collecting visible photo information from real images

**3.1.6**
**world reference frame**
coordinate reference frame used to express positions/orientations of the camera, 3D models and other objects for a given scene

## 3.2 Abbreviated terms

| | |
|---|---|
| IBR | image-based rendering |
| XML | extensible markup language |
| XSD | XML schema definition |

# 4 Domain and concepts

## 4.1 General

This clause describes the domain and key concepts. This includes the domain of image-based representation of objects/environments and overall description of its details.

## 4.2 Domain

Geometric and optionally photometric information is essential when representing target objects/environments as is done in ISO/IEC 19775-1. In this document, an alternative representation method, image-based, is pursued in which the objects/environments are represented only using the photogrammetric information (e.g. set of images and other related information) and optionally the underlying 3D geometric information.

Therefore, the image-based representation of objects/environments first requires a set of images and their photogrammetric information of the target objects/environments taken at various locations, and the 3D model represented using vertices and faces and the association between these two (if needed). Figure 1 shows the overall scene as represented a set of images (below), which have association to the 3D model (top). Note that with photogrammetric information alone, it is possible to render the target objects/environments from an arbitrary viewpoint by using and blending the sampled images around.

The association to the underlying 3D model, if it exists, can make the rendering output more accurate by helping to resolve visibility issues and registration errors.

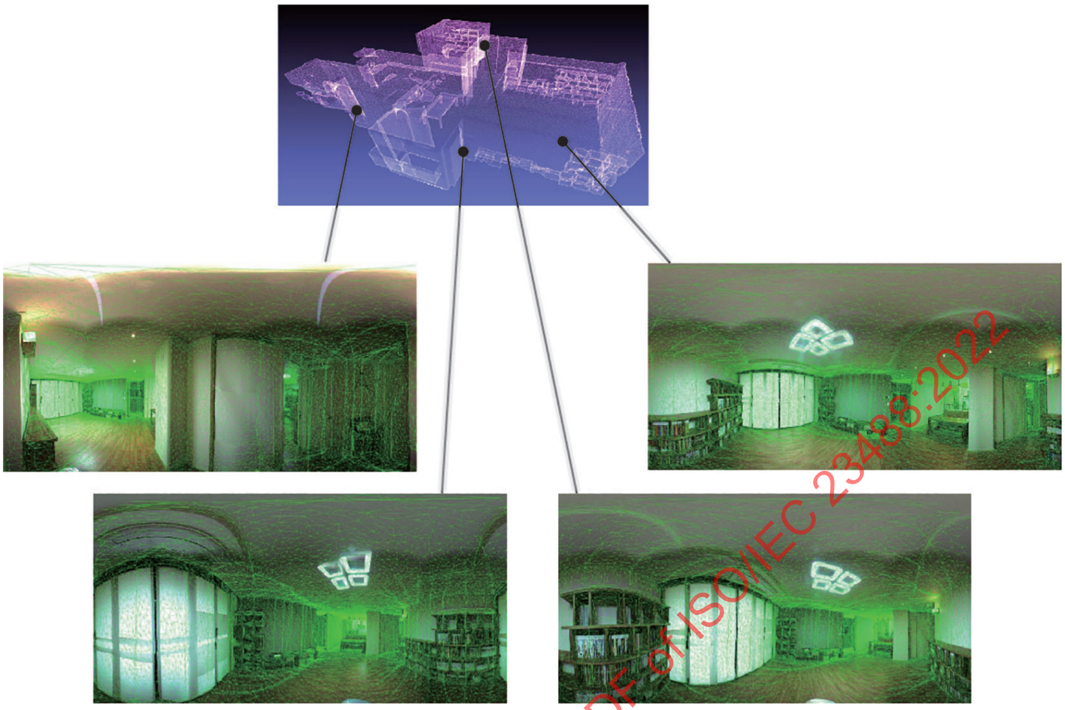


**Figure 1 — Example of a 3D model and image set for target environment representation**

To be more specific, because there are multiple blind spots in a wide and large environment that a single image cannot display, a combination of images is necessary to fully represent the whole target objects/environments without any such blind spots. Thus, to accurately associate the 3D model with the image set, each face of the 3D model is verified to be visible or not in each image of the image set. Then, the target objects/environments can be represented at an arbitrary viewpoint using only the visible image patches from different images[6],[7]. Specifically, the colour value of each face of the 3D model is determined from the corresponding visible image patches. After that, the determined colour values can be blended into a single virtual image to represent the target objects/environments at any arbitrary viewpoint.

In this regard, this document concerns the basic image-based objects/environments representation approach that uses only the images (or photogrammetric information) but also the usage and association of multiple images to the underlying 3D model for resolving the visibility of each face from where the images are taken and thereby making the rendering system more accurate and efficient. If the 3D model is used along with the image-based one, the resulting virtual or mixed reality world can be not only more photorealistic, but with the 3D depth information, more interesting interaction becomes possible.

## 4.3 Concepts

A key part of image-based representation is representing the photogrammetric information of target objects/environments which is the basis of the proposed information model. Then the next is the optional part as how to associate each face of the 3D model with visible image patches of the image set. By associating each face with visible image patches, the photometric information of the visible image patches can be accurately overlaid to each face of 3D model.

For this purpose, visibility of the 3D model's vertex/face at each image, which can be acquired using a visibility/occlusion detection algorithm like z-buffering[4], is first computed ahead of time (and added to the representation). Regarding visibility detection, the vertices/faces can be tessellated to improve visibility detection results in the case that the 3D model consists of sparse vertices.

Then, with the computed visibility information, objects/environments can be represented by connecting each face of the 3D model with the corresponding visible image patches. If a face has multiple visible image patches that can be mapped as texture, the photometric information of the face can be computed by, for example, selecting the best image patch among the candidates or by weighting the photometric information of multiple visible image patches. For the former case, to select one source image for the given face, the distance to the position where the image is captured, the area of the image, the texture quality (how well focused or blurriness) of the image can be considered. For the latter case, the aforementioned criteria can be used to derive relative weights in blending them into a virtual image texture as seen from an arbitrary viewpoint (see Figure 2).

## 4.4 Basic components

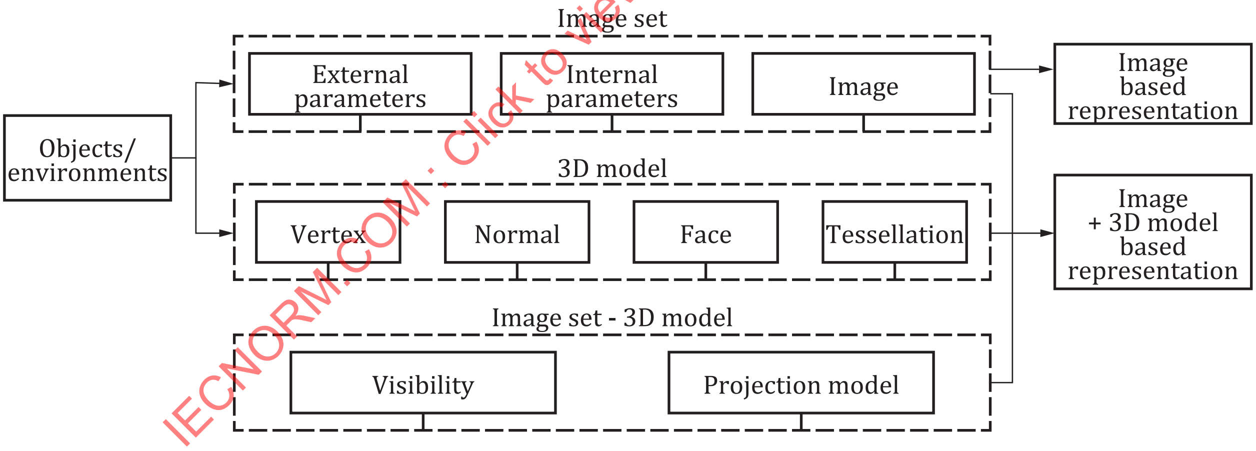
### 4.4.1 General

This subclause describes details of image-based representation. This includes explanation of the two key elements of image-based representation (image set and underlying 3D model), and their integration as in Figure 2.

### 4.4.2 Image set

#### 4.4.2.1 General

An image set consists of images capturing the photometric information of target objects/environments at different locations. The images can be taken by different entities using different hardware, allowing the usage of virtual and real images. Also, to project the images to a 3D model, it shall be augmented with the external parameters (rotation and translation) and the camera's internal parameters (focal length, resolution, aspect ratio, etc.). Using the external and internal parameters, a vertex or face of the 3D model can be projected onto the image and visibility information can be acquired, all of which are used to visualize a 3D model with multiple textures at an arbitrary viewpoint.



Visibility should be changed according to changing viewpoint and visibility of a candidate patch considered for the final image blend would be determined by a visibility/occlusion algorithm like z-buffering[4] and back-face culling.

**Figure 2 — Abstraction of image-based representation**

#### 4.4.2.2 Image

An image contains the photometric information of the target objects/environments. The photometric information can be depicted using any image format, including PNG, JPG, JPEG and TIFF. The photometric information also includes a colour model, which can be represented as RGB, HSL, or HSV. To acquire an image, there shall be no limits on the camera model, including pinhole, fisheye, cubemap, spherical and cylindrical models, as long as a projection model between the image and the 3D model can be formed. Further description of the projection model is given in 4.4.4.3.

#### 4.4.2.3 Camera external parameters

Camera external parameters are the rotation and translation of a camera when it captures an image with respect to the world reference frame. Thus, the number of camera external parameter sets is equal to the number of images used in rendering system. The camera external parameters transform the vertices of a 3D model to the camera coordinate system. Explicitly, the camera external parameters can be represented as a 3-by-4 matrix using floating points where the left 3-by-3 matrix represents a rotation matrix and the right 3-by-1 vector represents a translation[5].

#### 4.4.2.4 Camera internal parameters

Camera internal parameters geometrically describe how the target objects/environments around the camera are projected to an image. Thus, they can be used to project vertices of a 3D model, with respect to the camera coordinate system, to the image. The set of internal parameters depends on the projection model of the camera, which can be pinhole, fisheye, cubemap, spherical and cylindrical models, depending on the hardware characteristics of the camera. Each camera in the IBR capture system has a single set of internal parameters associated with one lens. Thus, for an IBR capture system with n cameras, where each camera has one or multiple lenses, there can be a total of *n* or more internal parameter sets.

### 4.4.3 3D model

#### 4.4.3.1 General

A 3D model contains the geometric information of the target environment as stated in 4.2. In this document, a 3D model is represented by a 3D mesh which consists of four main parts: vertex, normal, face, and tessellation. The vertex and normal values together mark a single point in the 3D space. These points are connected as triangles to form the face of the mesh structure. If a face is not occluded by other faces at a viewpoint, the face and the corresponding vertices are considered visible at the viewpoint. If tessellation is necessary, each face is split into sub-faces and the visibility at a viewpoint is recalculated.

#### 4.4.3.2 Vertex

A vertex refers to a point in two or three dimensional space. Among its attributes, the vertex position is written in floating points with respect to the world reference frame. A vertex can be projected to an image through external and internal camera parameters as in 4.4.4.3.

#### 4.4.3.3 Face

A face represents a polygon formed by a collection of vertices. A face requires all vertices in the polygon to be coplanar. The simplest polygon is the triangle with three vertices which are guaranteed to be coplanar. A face can be projected to an image through external and internal parameters as in 4.4.4.3.

#### 4.4.3.4 Normal

A vertex normal at a vertex of a polyhedron is a directional vector associated with a vertex, intended as a replacement to the true geometric normal of the surface. Commonly, it is computed as the normalized average of the surface normals of the faces that contain that vertex. Thus, there shall be the same number of vertex normal values as the number of vertices. This vertex normal can be used to shade a 3D model when rendering a target object/environment. It also allows compression of the object's surface. Face normal vectors perpendicular to a face can be computed using the vertex normals of the face. Multiple faces whose normal vectors are similar (relatively flat and almost coplanar) can be simplified into a single larger face with the averaged normal vector.

#### 4.4.3.5 Tessellation

A tessellation is the process of dividing a large face into a certain number of sub-faces. After the tessellation step, each sub-face is checked for visibility at every camera location to enhance visibility detection. The number of sub-faces is a non-limited integer value. Tessellation is conducted to increase the visibility detection results, but it is not a required part of the representation of target objects/ environments. Especially, tessellation of a compact 3D model with sparse vertices allows accurate visibility detection by increasing point resolution and thus results in more realistic representation. Further description of the visibility is given in 4.4.4.2.

### 4.4.4 3D model — Image set integration

#### 4.4.4.1 General

To connect the geometry of a 3D model with the photogrammetry of an image set, they need to be integrated in a systematic way. For this purpose, the visibility of each face is extracted with respect to each camera location. Thus, a face can be matched only with the appropriate images. In this process, the projection model mathematically relates a 3D model with an image.

#### 4.4.4.2 Visibility

Visibility is the property that indicates whether each face of a 3D model is visible in each image or not. Thus, each face of the 3D model has the same amount of visibility information as the number of images – i.e. whether visible from where the images were captured. Explicitly, visibility of a face (in either front or back direction) can be described with a Boolean vector of the same size as the number of images where the $n^{\text{th}}$ element indicates the visibility of the face in the $n^{\text{th}}$ image as true/false, which can be acquired using a visibility/occlusion algorithm like z-buffering[4] and back face culling. For accurate visibility with a sparse model, tessellation can be conducted before extracting visibility information as stated in 4.4.3.5.

#### 4.4.4.3 Model projection

Model projection is a mathematical procedure of how vertices of a 3D model in world reference frame are projected to an image. To do this, vertices of a 3D model are first transformed into the camera coordinate system by using the camera external parameters as stated in 4.4.2.3. Then, the transformed vertices are projected to an image through the projection model of a camera that captured the image using the camera internal parameters as stated in 4.4.2.4. As stated in 4.4.2.2, the projection models can vary as long as it projects the 3D model to the image.

### 4.4.5 XML based object model

This subclause provides a layout of the XML schema for the exemplary XML. Note the highlighted major objects.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        targetNamespace="http://www.anywebsite.com/IBM" xmlns:ibm="http://www.anywebsite.
com/IBM">

<element name="IBM">
  <complexType>
   <sequence>
    <element name="cameraMatrices">
     <complexType>
      <sequence>
       <element name="camera" type="ibm:cameraMatrixType">   </element>
     </sequence>
    </complexType>
   </element>

   <element name="imageset">
```

```
    <complexType>
     <sequence>
      <element name="image" type="ibm:imageType" minOccurs="1" maxOccurs="unbounded"> </
element>
    </sequence>
   </complexType>
 </element>

<element name="meshGroup">
 <complexType>
 <sequence>
  <element name="vertex">
   <complexType>
    <sequence>
     <element name="coordinate" type="ibm:coordinateType" minOccurs="4"
maxOccurs="unbounded"></element>
   </sequence>
  </complexType>
</element>

<element name="normal">
 <complexType>
  <sequence>
   <element name="coordinate" type="ibm:coordinateType" minOccurs="4"
maxOccurs="unbounded"></element>
  </sequence>
 </complexType>
</element>

<element name="face">
 <complexType>
  <sequence>
   <element name="index" type="ibm:coordinateType" minOccurs="1" maxOccurs="unbounded"></
element>
  </sequence>
 </complexType>
</element>

<element name="tessellation" minOccurs="0" maxOccurs="unbounded">
 <complexType>
  <sequence>

   <element name="barycentric" minOccurs="1" maxOccurs="unbounded">
    <complexType>
     <sequence>
      <element name="x" type="float"></element>
      <element name="y" type="float"></element>
    </sequence>
   </complexType>
 </element>

   <element name="visibility">
    <complexType>
     <sequence>
      <element name="visdata" minOccurs="1" maxOccurs="unbounded">
       <complexType>
        <simpleContent>
         <extension base="string">
          <attribute name="id" type="integer"></attribute>
         </extension>
        </simpleContent>
       </complexType>
      </element>
     </sequence>

     <attribute name="index" type="integer"></attribute>
    </complexType>
   </element>

  </sequence>
  <attribute name="size" type="integer"></attribute>
 </complexType>
</element>

</sequence>
</complexType>

</element>
</sequence>
</complexType>
</element>

 <complexType name="cameraMatrixType">
    <sequence>
      <element name="value" type="float" maxOccurs="16" minOccurs="16"> </element>
     </sequence>
    <attribute name="id" type="integer"></attribute>
  </complexType>

 <complexType name="imageType">
  <attribute name="id" type="integer"></attribute>
  <attribute name="path" type="string"></attribute>
  </complexType>

  <complexType name="coordinateType">
    <sequence>
      <element name="x" type="float"></element>
       <element name="y" type="float"></element>
       <element name="z" type="float"></element>
    </sequence>
   </complexType>

</schema>
```

# 5 Image-based representation usage example

## 5.1 General

This clause describes the usage example of image-based representation. This includes image-based rendering and multi-object representation.

## 5.2 Image-based rendering

Image-based rendering refers to rendering a 3D model according to a virtual camera using an image set. Note that virtual camera contains the information in addition to the virtual viewpoint (which just contains the camera position and pose) such as the internal camera parameters including the focal length, field of view, near/far planes, and skew parameters; see also 3.1.1. and 3.1.4. For IBR, image-based representation in this document can be used to extract the photometric information of each face of 3D model by providing the corresponding visible image patches. Details of the rendering process at an arbitrary viewpoint using the image-based representation are as follows:

First, visible vertex/face of a 3D model at the query viewpoint is extracted using a visibility/occlusion detection algorithm like z-buffering[4]. Second, colour candidates of each visible vertex/face are acquired from an input image set using the visibility vector stated in 4.4.4.2. Third, colour value of each visible vertex/face is calculated using colour candidates. Finally, rendering is conducted using the calculated colour values and the corresponding visible vertex/face.

When calculating the colour values from the colour candidates, several approaches can be applied. One is to select the image closest to the query viewpoint and combine colour information with the visible vertex/face at the image. The distance between the image and the query viewpoint can be calculated based on the camera external parameters of the image and the viewpoint. After that, colour information of the remaining parts is acquired from the image second closest to the query viewpoint. This blind spot texturing is recursively conducted until every visible/face at the query viewpoint has corresponding image patches. Another approach can be selecting the colour values from the image patch closest to each vertex/face, or calculating the colour values as the weighted sum of the colour candidates. Also, the acquired colour information (or image patches) from any approaches can be stitched into a virtual image to remove seams between patches.

### 5.3 Multi-object representation

Constructs proposed in this document also can use multiple 3D geometry models (multi-object representation) to render the target space as in Figure 3. In this case, each model/object can be independently textured using different sets of images. Thus, the photometric information of different models does not affect one another. A model/object can also be textured with a virtual image containing only the photometric information of that model/object. Moreover, the visibility information can be obtained by considering occlusion caused by other models/objects.

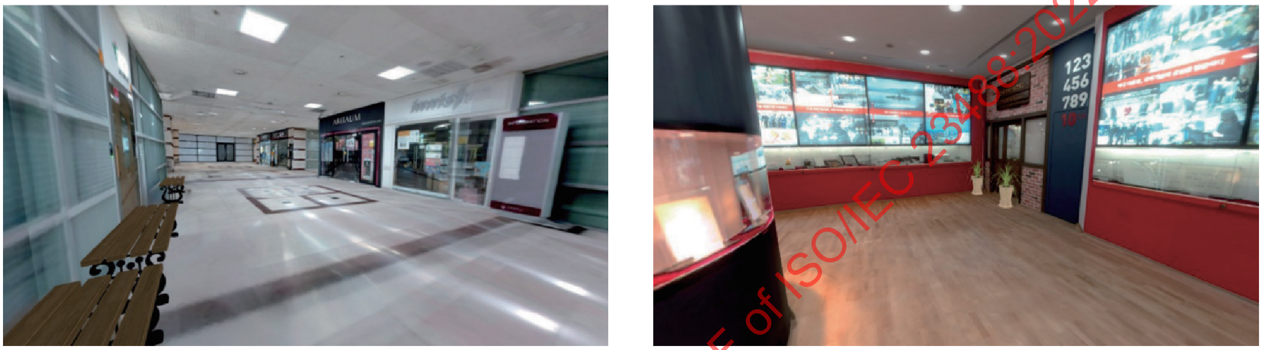


**Figure 3 — Example of multi-object representation with virtual images of objects (benches and plants) augmented to the 3D geometry model**

## 6 Conformance

### 6.1 Objective

The primary objectives of the specifications Table 1 are:

a) promoting interoperability by eliminating arbitrariness;

b) promoting uniformity in using IBR;

c) promoting consistent results.

## 6.2 Minimum requirements

**Table 1 — Conformance check list**

| | | Item | Check | Remarks |
|---|---|---|---|---|
| Model projection (required) | Camera | There is a construct for representing camera position (required). | (____) | |
| | | Camera position values form a proper view matrix with floating numbers (required). | (____) | |
| | | Intrinsic and extrinsic camera parameters are properly included (required). | (____) | |
| | Image | There is a construct for representing an image (or image URI) and a set of them (required). | (____) | |
| | | Image URIs are indexed for referral and association with the 3D data, if needed (required). | (____) | |
| | | Projection between image and 3D model is formed (optional). | (____) | |
| | Image set integration | Every image has a corresponding camera position and such information can be encoded (required). | (____) | |
| | | All faces are matched with appropriate images (optional). | (____) | |
| Object structure (optional) | Mesh | There is a construct for representing the 3D mesh (optional). | (____) | |
| | | The 3D mesh is composed of vertices, faces and normals, with coordinate values in floating numbers (optional). | (____) | |
| | Tessellation | Number of tessellations is set to an integer (optional). | (____) | |
| | | Every mesh is set for tessellation (optional). | (____) | |

# Annex A
(informative)

# Working example of the proposed information model

## A.1 General

Clause A.2 represents the exemplary data and Clause A.3 represents the XML example. For the XML document to be well-formed, XSD is listed in 4.4.5.

## A.2 Exemplary data

### A.2.1 General

This clause contains exemplary data for the XML example in Clause A.3.

### A.2.2 Camera information representation

When the camera location and view matrix information are stored, this is later used in the rendering process. Exemplary camera poses are shown as a vertical line in Figure A.1.

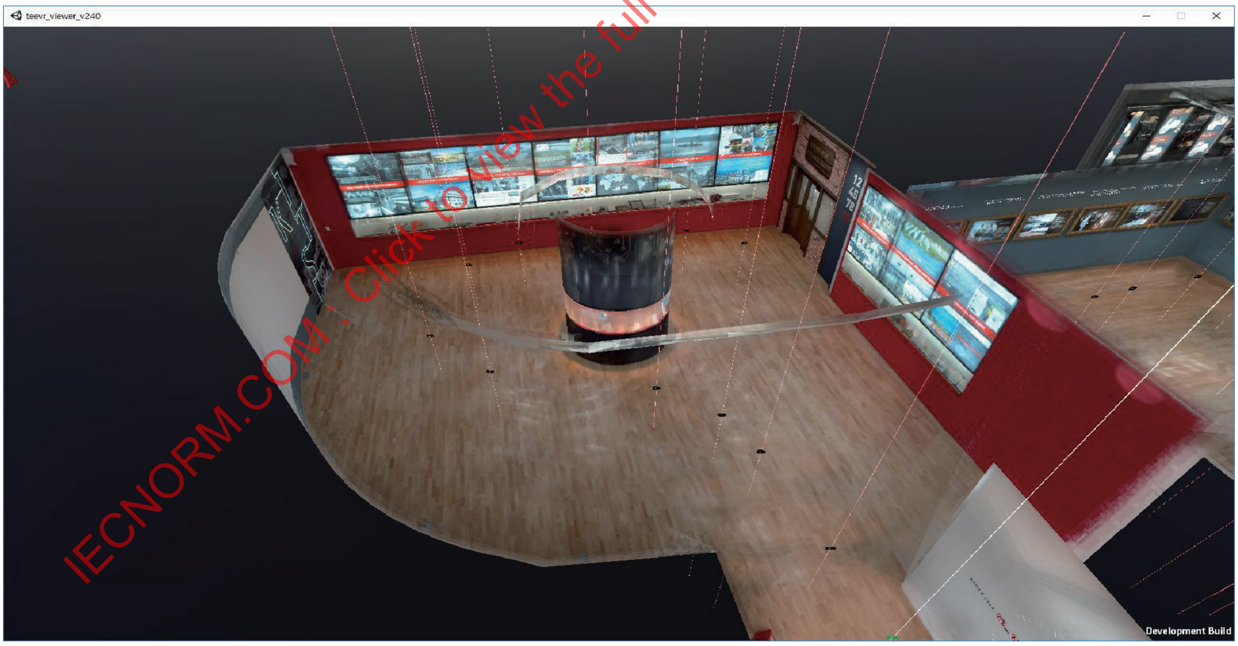


**Figure A.1 — Example of camera pose data represented in a virtual 3D model**

### A.2.3 Image information representation

For every camera pose shown in Figure A.1, a 360° image taken at that location is projected into the mesh of the object. An image taken at the exemplary pose is given in Figure A.2.