INTERNATIONAL STANDARD

ISO/IEC 23090-14

First edition
2023-06

Information technology — Coded representation of immersive media —

Part 14:
Scene description

Technologies de l'information — Représentation codée de média immersifs —

Partie 14: Description de scènes

Reference number
ISO/IEC 23090-14:2023(E)

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).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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 23090 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.

Introduction

This document defines the MPEG-I Scene Description. It provides an architecture for the MPEG-I Scene Description, a set of extensions based on ISO/IEC 12113, a set of APIs, and storage formats for scene description documents and scene description updates documents.

Information technology — Coded representation of immersive media —

Part 14: Scene description

1 Scope

This document specifies extensions to existing scene description formats in order to support MPEG media, in particular immersive media. MPEG media includes but is not limited to media encoded with MPEG codecs, media stored in MPEG containers, MPEG media and application formats as well as media provided through MPEG delivery mechanisms. Extensions include scene description format syntax and semantics and the processing model when using these extensions by a Presentation Engine. It also defines a Media Access Function (MAF) API for communication between the Presentation Engine and the Media Access Function for these extensions. While the extensions defined in this document can be applicable to other scene description formats, they are provided for ISO/IEC 12113.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 12113, *Information technology — Runtime 3D asset delivery format — Khronos glTF™ 2.0*

ISO/IEC 14496-12, *Information technology — Coding of audio-visual objects — Part 12: ISO base media file format*

ISO/IEC 21778, *Information technology — The JSON data interchange syntax*

IEEE 754-2019, *IEEE Standard for Floating-Point Arithmetic*

IETF RFC 6902, *JavaScript Object Notation (JSON) Patch*

IETF RFC 8259, *The JavaScript Object Notation (JSON) Data Interchange Format*

3 Terms, definitions, abbreviated terms, and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 12113 and the following 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
asset
3D scene described by a *scene description document* (3.1.10) together with corresponding *scene description data* (3.1.9)

3.1.2
node
element in the *scene graph* (3.1.12)

3.1.3
media access function
function that retrieves and prepares media for rendering on request by the *presentation engine* (3.1.7)

3.1.4
media pipeline
chain of media processing components to process media

3.1.5
object
node in a *scene description document* (3.1.10)

3.1.6
patch document
document that contains update instructions

Note 1 to entry: For example, update instruction can be provided as defined in RFC 6902.

3.1.7
presentation engine
engine that processes and renders the *asset* (3.1.1)

3.1.8
scene activation time
time on the media timeline at which the scene described by a *scene description document* (3.1.10) takes effect in the *presentation engine* (3.1.7)

3.1.9
scene description data
binary data that is described by *scene description document* (3.1.10)

3.1.10
scene description document
document describing a 3D scene

Note 1 to entry: For example, scene description document is containing description of node hierarchy, materials, cameras, as well as description information for meshes, animations, and other constructs.

3.1.11
scene description update
patch document (3.1.6) to a *scene description document* (3.1.10) or a *scene description document* (3.1.10)

3.1.12
scene graph
data structure used to represent *objects* (3.1.5) in a 3D scene and their hierarchical relationships

3.1.13
timed accessor
accessor defined in ISO/IEC 12113 that has an MPEG_accessor_timed extension and is used to describe access to timed data

3.1.14
timed data
timed media
media, which when decoded results in content, possibly containing internal timing values, to be presented at a given presentation time and for a certain duration

3.2 Abbreviated terms

3D	Three-Dimensional
3DoF	Three Degrees of Freedom
6DoF	Six Degrees of Freedom
API	Application Programming Interface
AR	Augmented Reality
DASH	Dynamic Adaptive Streaming over HTTP
dB	Decibel
DSR	Diffuse to Source Ratio
glTF	Graphics Language Transmission Format
HOA	Higher Order Ambisonics
ISOBMFF	ISO Base Media File Format
JSON	JavaScript Object Notation
MAF	Media Access Function
MPEG	Moving Picture Experts Group
IDL	Interface Definition Language
PCM	Pulse-Code Modulation
RT60	60 dB Reverberation Time
SDP	Session Description Protocol

3.3 Conventions

3.3.1 General

The mathematical operators used in this document are similar to those used in the C programming language. However, the results of integer division and arithmetic shift operations are defined more precisely, and additional operations are defined, such as exponentiation and real-valued division. Numbering and counting conventions generally begin from 0.

3.3.2 Arithmetic operators

+	addition
−	subtraction (as a two-argument operator) or negation (as a unary prefix operator)
*	multiplication, including matrix multiplication
/	integer division with truncation of the result toward zero. For example, 7 / 4 and −7 / −4 are truncated to 1 and −7 / 4 and 7 / −4 are truncated to −1.
÷	division in mathematical equations where no truncation or rounding is intended.

3.3.3 Logical operators

! Boolean logical "not".

3.3.4 Relational operators

\> Greater than.

\>= Greater than or equal to.

< Less than.

<= Less than or equal to.

== Equal to.

!= Not equal to.

3.3.5 Bit-wise operators

~ bit-wise "not".
When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

& bit-wise "and".
When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

| bit-wise "or".
When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

^ bit-wise "exclusive or".
When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

x >> y arithmetic right shift of a two's complement integer representation of x by y binary digits. This function is defined only for non-negative integer values of y. Bits shifted into the MSBs as a result of the right shift have a value equal to the MSB of x prior to the shift operation.

x << y arithmetic left shift of a two's complement integer representation of x by y binary digits. This function is defined only for non-negative integer values of y. Bits shifted into the LSBs as a result of the left shift have a value equal to 0.

3.3.6 Assignment operators

= assignment operator.

++ increment, i.e. x++ is equivalent to x = x + 1; when used in an array index, evaluates to the value of the variable prior to the increment operation.

-- decrement, i.e. x-- is equivalent to x = x − 1; when used in an array index, evaluates to the value of the variable prior to the decrement operation.

+= increment by amount specified, i.e. x += 3 is equivalent to x = x + 3, and x += (−3) is equivalent to x = x + (−3).

−= decrement by amount specified, i.e. x −= 3 is equivalent to x = x − 3, and x −= (−3) is equivalent to x = x − (−3).

3.3.7 Other operators

y..z range operator/notation.
This function is defined only for integer values of y and z. When z is larger than or equal to y, it defines an ordered set of values from y to z in increments of 1. Otherwise, when z is smaller than y, the output of this function is an empty set. If this operator is used within the context of a loop, it specifies that any subsequent operations defined are performed using each element of this set, unless this set is empty.

3.3.8 Order of operation precedence

When order of precedence in an expression is not indicated explicitly by use of parentheses, the following rules apply:

— Operations of a higher precedence are evaluated before any operation of a lower precedence.

— Operations of the same precedence are evaluated sequentially from left to right.

Table 1 specifies the precedence of operations from highest to lowest; a higher position in the table indicates a higher precedence.

NOTE For those operators that are also used in the C programming language, the order of precedence used in this document is the same as used in the C programming language.

Table 1 — Operation precedence from highest (at top of table) to lowest (at bottom of table)

operations (with operands x, y, and z)
"x++", "x--"
"!x", "−x" (as a unary prefix operator)
"x * y", "x / y", "x ÷ y", "x % y"
"x + y", "x − y" (as a two-argument operator)
"x << y", "x >> y"
"x < y", "x <= y", "x > y", "x >= y"
"x == y", "x != y"
"x & y"
"x | y"
"x && y"
"x || y"
"x ? y : z"
"x..y"
"x = y", "x += y", "x −= y"

3.3.9 Text description of logical operations

In the text, a statement of logical operations as would be described mathematically in the following form:


```
if( condition 0 )
   statement 0
else if( condition 1 )
   statement 1
...
else /* informative remark on remaining condition */
   statement n
```

may be described in the following manner:

... as follows / ... the following applies:

— If condition 0, statement 0

— Otherwise, if condition 1, statement 1

— ...

— Otherwise (informative remark on remaining condition), statement n

Each "If ... Otherwise, if ... Otherwise, ..." statement in the text is introduced with "... as follows" or "... the following applies" immediately followed by "If ... ". The last condition of the "If ... Otherwise, if ... Otherwise, ..." is always an "Otherwise, ...". Interleaved "If ... Otherwise, if ... Otherwise, ..." statements can be identified by matching "... as follows" or "... the following applies" with the ending "Otherwise, ...".

In the text, a statement of logical operations as would be described mathematically in the following form:


```
if( condition 0a && condition 0b )
   statement 0
else if( condition 1a || condition 1b )
   statement 1
...
else
   statement n
```


may be described in the following manner:

... as follows / ... the following applies:

— If all of the following conditions are true, statement 0:

  — condition 0a

  — condition 0b

— Otherwise, if one or more of the following conditions are true, statement 1:

  — condition 1a

  — condition 1b

— ...

— Otherwise, statement n

In the text, a statement of logical operations as would be described mathematically in the following form:

if(condition 0)

statement 0

if(condition 1)

statement 1

may be described in the following manner:

When condition 0, statement 0

When condition 1, statement 1

In addition, a “continue” statement, which is used within loops, is defined as follows:

The “continue” statement, when encountered inside a loop, jumps to the beginning of the loop for the next iteration. This results in skipping the execution of subsequent statements inside the body of the loop for the current iteration. For example:

```
for( j =0; j < N; j++ ) {
  statement 0
  if( condition 1 )
      continue
  statement 1
  statement 2
}
```

is equivalent to the following:

```
for( j =0; j < N; j++ ) {
  statement 0
  if( !condition 1 ) {
     statement 1
     statement 2
  }
}
```


4 Overview and architecture

4.1 Overview

This document enables inclusion of timed media in a scene description. This is achieved through first defining features of a scene description that describe how to get the timed media, and second how a rendering process expects the data once it is decoded. In this version of the document, the features are defined as extensions to the glTF format defined in ISO/IEC 12113, see Clause 5.

In addition to the extensions, which provide an integration of timed media with the scene description, the document describes a reference scene description architecture that includes components such as Media Access Function, Presentation Engine, Buffer Control & Management, and Pipelines. To enable cross-platform/cross-vendor interoperability, the document defines Media Access Function (MAF) API and Buffer API, see Clause 6. The MAF API provides an interface between the Media Access Function and the Presentation Engine. The Buffer API is used to allocate and control buffers for the exchange of data between Media Access Function and Presentation Engine.

Not only the timed media described by the scene description may change over the time but also the scene description itself. The document defines how such change of a scene description document is signalled to the Presentation Engine.

Finally, a scene description may be stored, delivered, or extended in a way that is consistent with MPEG formats. The document defines a number of new features that allow a carriage utilizing ISO/IEC 14496-12 and its derived specifications, see Clause 7.

4.2 Architecture

The scene description is consumed by a Presentation Engine to render a 3D scene to the viewer. The extensions defined in this document, allow for the creation of immersive experiences using timed media. The scene description extensions are designed with the goal of decoupling the Presentation Engine from the Media Access Function. Presentation Engine and Media Access Function communicate through the Media Access Function API, which allows the Presentation Engine to request timed media required for the rendering of the scene. The Media Access Function will retrieve the requested timed media and make it available in a timely manner and in a format that can be immediately processed by the Presentation Engine. For instance, a requested timed media asset may be compressed and residing in the network, so the Media Access Function will retrieve and decode the asset and pass the resulting decoded media data to the Presentation Engine for rendering. The decoded media data is passed in form of buffers from the Media Access Function to the Presentation Engine. The requests for timed media are passed through the Media Access Function API from the Presentation Engine to the Media Access Function.

Figure 1 depicts the reference architecture.

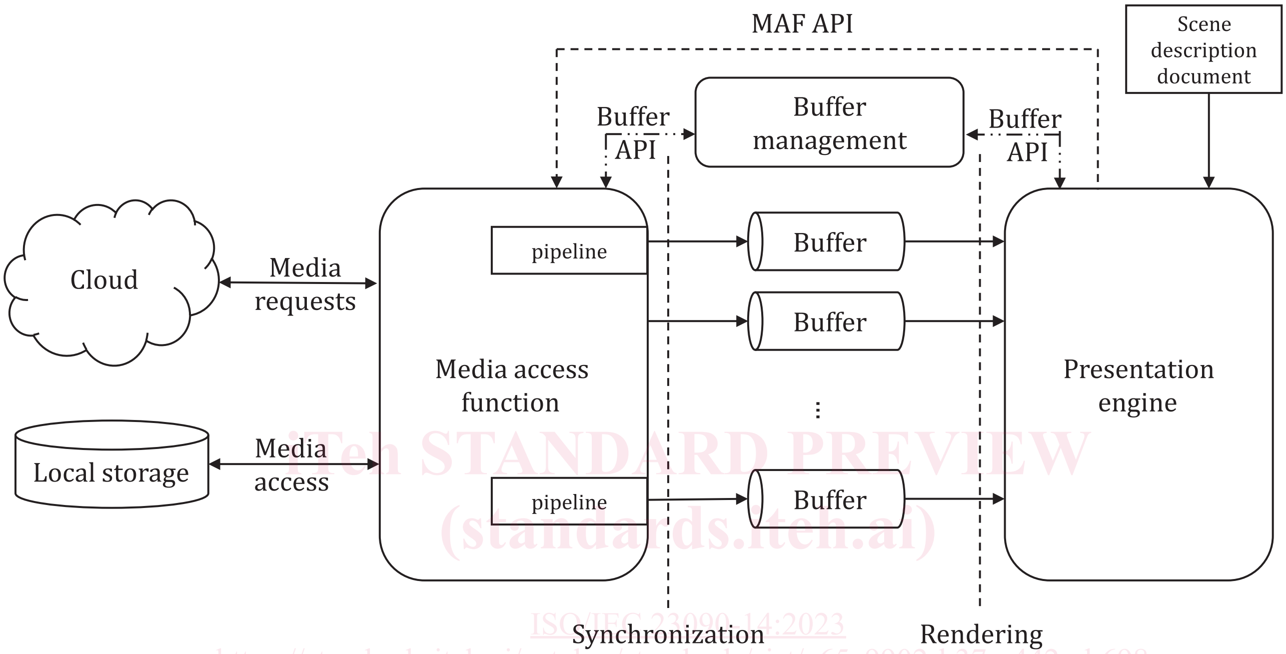

Figure 1 — Scene description reference architecture

The interfaces (MAF API, Buffer API) and extensions to ISO/IEC 12113 are within the scope of this document.

The following principles apply:

— The format of the buffers shall be provided by the scene description document and shall be passed to the MAF through the Media Access Function API

— Pipeline shall perform necessary transformations to match the buffer format and layout declared in the scene description for that buffer

— The fetching of scene description document and scene description updates may be triggered by the MAF.

Figure 1 depicts the reference architecture for scene description. The corresponding procedures are described as follows:

a) The Presentation Engine receives and parses the scene description document and following scene description updates

b) The Presentation Engine identifies timed media that needs to be presented and identifies the required presentation time

c) The Presentation Engine then uses the MAF API to request the media and provides the following information:

   1) where the MAF can find the requested media

   2) what parts of the media and at what level of detail

   3) when the requested media has to be made available

   4) in which format it wants the data and how it is passed to the Presentation Engine

d) The MAF instantiates the media fetching and decoding pipeline for the requested media at the appropriate time.

   1) It ensures that the requested media is available at the appropriate time in the appropriate buffers for access by the Presentation Engine

   2) It ensures that the media is decoded and reformatted to match the expected format by the Presentation Engine as described by the scene description document

The exchange of data (media and metadata) shall be done through buffers (circular and static buffers). The buffer management shall be controlled through the Buffer API. Each buffer should contain sufficient header information to describe its content and timing.

The information provided to the Media Access Function by the Presentation Engine allows it to

— Select the appropriate source for the media (multiple could be specified) and the MAF may select based on preferences and capabilities. Capabilities may for example be decoding capabilities or supported formats. Preferences may for example be user settings.

— For each selected source,

   i) access the media by using a media access protocol;

   ii) setup the media pipeline to provide the information in the correct buffer format.

The MAF may obtain additional information from the Presentation Engine in order to optimize the delivery, for example the required quality for each of the buffers, the exact timing information, etc.

The Media Access Function shall setup and manage the pipeline for each requested media or metadata. A pipeline takes as input one or more media or metadata tracks and outputs one or more buffers. The pipeline shall perform all the necessary processing, such as streaming, demultiplexing, decoding, decryption, and format conversion to match the expected buffer format. The final buffer or set of buffers are then used to exchange data with the Presentation Engine.

An example of pipelines setup is depicted in Figure 2 for the case of a V-PCC compressed point cloud object that is referenced in the scene description. Pipeline #1 creates four video decoders and one patch data decoder. The pipeline is also responsible for processing this data and performing 3D reconstruction based on the received information. The reconstructed data is then fed to the final buffer that is accessed by the Presentation Engine. Pipeline #2 on the other hand is not performing the 3D reconstruction process and provides decoded raw data onto the buffers, which are accessed by the Presentation Engine.