ISO/IEC JTC 1/SC 29/WG1N84033
??. ??. 2020

ISO/IEC JTC 1/SC 29/WG 1
(& ITU-T SG16)

Coding of Still Pictures

JBIG	JPEG
Joint Bi-level Image Experts Group	Joint Photographic Experts Group

TITLE: Draft text of ISO/IEC 19566-7

SOURCE: JPEG Systems Sub Group
JPEG JLINK

EDITORS: Park, Jung-Hwan (park@pjfactory.com)
Kuzma, Andy (andy.kuzma@intel.com)
Junichi Hara (waverz@jp.ricoh.com)

PROJECT: ISO/IEC 19566-7
JPEG Systems Part 7: JLINK

STATUS: WD 3.0

REQUESTED ACTION: For review

DISTRIBUTION: WG1

Contributors:	**Yusa Ko, Youngseop Kim, Takaaki Ishikawa, Frederik Temmermans , Seungcheol Choi, Kwon Oh-Jin**

INTERNATIONAL ORGANISATION FOR STANDARDISATION

INTERNATIONAL ELECTROTECHNICAL COMMISSION

Information technology:

ISO/IEC 19566-7 JPEG Systems Part 7: JPEG Media Linked Format (JLINK)

International Standard

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, the joint technical committee may propose the publication of a Technical Report of one of the following types:

— type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;

— type 3, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

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.

ISO/IEC TR 19566-7, which is a Technical Report of type 3, was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Introduction

'JPEG Linked Media Format (JLINK)' is an image file format capable of embodying multiple image types and other media elements, such as text and sound, into a single file. It also specifies how these metadata elements are contained and related within a JUMBF box to one another.

An image has been a primary vessel to convey ideas and to disseminate stories. JPEG has been at the forefront of enhancing the capability of digital images via ensuring the quality while efficiently compressing them for better publication in the web. However, knowing that a widely consumed image likely has more context attached to it, the functionality of the JLINK provides users new ways they can tell their stories within a single image file. It establishes a uniform method to relate these metadata elements enabling interactions such as navigating among images and opening overlaid text descriptions using overlaid sprites in order to convey a story.

This Recommendation | International standard contributes to the specification of system level functionalities that can provide a degree of trust while sharing image content and metadata, and simultaneously also allowing the signalling of the associated access policies.

INTERNATIONAL STANDARD
ISO/IEC 29199-2 : 200x (E)
ITU-T Rec. T.xxxx (200x E)
ITU-T RECOMMENDATION ???

INFORMATION TECHNOLOGY –
INFORMATION TECHNOLOGY – JPEG SYSTEMS PART 7:
JPEG LINKED MEDIA FORMAT (JLINK)

1 Scope

This Recommendation | International standard contributes to a system layer for JPEG standards, referred to as JPEG Systems. It gives an overview about the existing JPEG ecosystem in order to show their relation and their features.

This Recommendation | International standard:

- specifies an access policy definition;
- specifies hierarchical structure of linked media;
- specifies a container-based file format;
- provide a signalling mechanism to identify an applied access policy and data link tools
- provide guidance on implementation of an image repository with controlled access

This standard aims to provide technical solutions for resolving image container for structuring multiple types of media into a single file, including definition of metadata specification for multiple types of media. It complies with legacy technology in the domain, such as image coding technology as well as metadata standards that signal access policies, IPR conditions and others.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

ISO/IEC 19566-1: Information Technology – JPEG Systems – Part 1: Packaging of information using codestream and file formats

ISO/IEC 18477-1: Information Technology - Scalable Compression and Coding of Continuous-Tone Still Images: Scalable compression and coding of continuous-tone still images

ISO/IEC 18477-3: Information Technology - Scalable Compression and Coding of Continuous-Tone Still Images: Box file format

ISO/IEC 19566-5, Information technology — JPEG Systems — Part 5: JPEG Universal Metadata Box Format (JUMBF)

2.1 Additional references

Rec. ITU-T T.800 | ISO/IEC 15444-1: Information technology – JPEG 2000 Image Coding System – Part 1: Core coding system

ISO/IEC 14496-12: Information technology - JPEG 2000 image coding system - Part 12: ISO base media file format

Rec. ITU-T T.805 | ISO/IEC 15444-6: Information technology -- JPEG 2000 image coding system -- Part 6: Compound image file format

Rec. ITU-T T.807 | ISO/IEC 15444-8: Information technology -- JPEG 2000 image coding system: Secure JPEG 2000

Rec. ITU-T T.81 | ISO/IEC 10918-1: Information Technology – Digital Compression and Coding of Continuous Tone Still Images – Requirements and Guidelines

Rec. ITU-T T.871 | ISO/IEC 10918-5: Information Technology -- Digital compression and coding of continuous-tone still images: JPEG File Interchange Format

ISO/IEC 24800-4: Information Technology – JPSearch – Part 4: File format for metadata embedded in image data (JPEG and JPEG 2000)

ISO/TC 130, ISO 16684-1: Graphic technology – Extensible metadata platform (XMP) specification – Part 1: Data model, serialization and core properties

Exchangeable image file format for digital still cameras (Exif 2.3)

Photo Metadata IPTC Core (Specification Version 1.1)

Photo Metadata IPTC Extension (Specification Version 1.1)

Extensible Access Control Markup Language (XACML) Version 3.0

Extensible Markup Language (XML)

The PROV Data Model (PROV-DM)

The Provenance Notation (PROV-N)

The PROV Ontology (PROV-O)

The PROV XML Schema (PROV-XML)

3 Definitions, Abbreviations and Symbols

3.1 Definitions

For the purposes of this International Standard, the following definitions apply. The definitions defined in clause 3 of ISO/IEC 19566-1, ISO/IEC 19566-5 and ISO/IEC 19566-6 also apply to this International Standard.

backward compatibility: A standard is backward compatible when the new specification includes the old one.

bit stream: Partially encoded or decoded sequence of bits comprising an entropy-coded segment.

container: a structured collection of data describing the multiple images or the multiple images decoding process. See Annex A for the definition of boxes.

container-based file format: A file format whose composing elements are well defined, hierarchically structured containers.

JPEG Media Linked Format (JLINK): A file format specifies how metadata elements contained or described within a JUMBF box are related to one another. Establishing a uniform method to relate these metadata elements enables applications, such as navigating among images and opening overlaid text descriptions using overlaid sprites for the purpose of creating a story or information.

Story: Multimedia elements arranged in a tree structure.

Scene: The basic unit that composes a story is called a scene. One scene is composed of a combination of multimedia that compose a node in the story tree structure and associated actions.

Transition Effect: Animated transition between scenes.

Plot link: Link between scenes. It represents relationships such as "total and part", "outline and detail," and "before and after" between two scenes, while also representing "scene transitions" from one scene to another.

Plot link container: It contains metadata for the transition of scenes such as sprites linked to other scenes and its position, transition animation, and viewport.

Reference link: Assistive access to support efficient navigation in JLINK. Allows circular or concatenated connections between scenes.

Reference link container: Similar to transitions, such as position, viewport, sprite assignments, scene animations, etc., it has metadata about scene transitions between scenes.

Additional component: Additional elements that support buttons and animations presented to the user for navigation.

Sprite: The button or area presented to the user.

box: A portion of the file format defined by a length and unique box type. Boxes of some types may contain other boxes.
box-based file format: A file format whose composing elements are well defined, hierarchically structured boxes.

box contents: Refers to the data wrapped within the box structure. The contents of a particular box are stored within the DBox field within the box data structure.

box type: Specifies the kind of information that shall be stored with the box. The type of a particular box is stored within the TBox field within the box data structure.

byte: A group of 8 bits.

codestream: A sequence of bytes that conforms to the codestream requirements specified in Rec. ITU-T T.800 | ISO/IEC 15444-1, ISO/IEC 18477-1, ISO/IEC 1877-2 or ISO/IEC 18477-3 or is to be tested to conformance with one of the above Recommendations | International Standards.

coding: Encoding or decoding.

compression: Reduction in the number of bits used to represent source image data.

decoder: An embodiment of a decoding process.

decoding process: A process which takes as its input compressed image data and outputs a continuous-tone image.

encoder: An embodiment of an encoding process.

encoding process: A process which takes as its input a continuous-tone image and outputs compressed image data.

forward compatibility: If a new standard is forward compatible, than devices only compliant with the old version of the standard are nevertheless able to interpret the data conforming with the new standard.

Joint Photographic Experts Group; JPEG: The informal name of the committee which created this Specification. The “joint” comes from the ITU-T and ISO/IEC collaboration.

JUMBF: A box-based file format for universal metadata stored into JPEG family images; specified in ISO/IEC 19566-5.

legacy decoder: An embodiment of a decoding process conforming to Rec. ITU-T T.81|ISO/IEC 10918-1, confined to the lossy DCT process and the baseline, sequential or progressive modes, decoding at most four components to eight bits per component.

marker: A two-byte code in which the first byte is hexadecimal FF and the second byte is a value between 1 and hexadecimal FE.

marker segment: A marker together with its associated set of parameters.

metadata: Data that describes information about other data, and that are classified as descriptive, structural and administrative metadata.

null-terminated string: A character string stored as an array containing the characters and terminated with a null character ('\0', called NUL in ASCII).

pixel: A collection of sample values in the spatial image domain having all the same sample coordinates, e.g. a pixel may consist of three samples describing its red, green and blue value.

relative Earth gravity: A relative information from a direction of Earth gravity that informs for identifying positional relationships with fixed objects in those pictures when 360-degree images are taken at the same place.

XML: A markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

XMP: An ISO standard, originally created by Adobe Systems Inc., for the creation, processing and interchange of standardized and custom metadata for digital documents and data sets..

zero byte: The hexadecimal 00 byte.

3.2 Abbreviatios

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

ASCII	American Standard Code for Information Interchange
DCT	Discreate Cosine Transform
ERP	EquiRectangular Projection
FOV	Field Of View
JSON	JavaScript Object Notation
JPEG	Joint Photographic Experts Group
JUMBF	JPEG Universal Metadata Box Format
MIME	Multipurpose Internet Mail Extensions
URL	Uniform Resource Locator
XML	eXtensible Markup Language
XMP	eXtensible Metadata Platform
RDF	Resource Description Framework
W3C	World Wide Web Consortium
UMF	Universal Metadata Framework.

3.3 JLINK Foundational Concepts

In this section, we provide descriptions of the basic concepts on which JLINK is built.

Fig. 1 organization of pictures among 'a still picture,' 'a motion picture,' and 'a JLINK'.

JLINK is Using multiple still pictures as building blocks, the motion picture and JLINK arranges them in different manners to achieve their goals. While a motion picture arranges still pictures linearly to create the optical phenomenon known as persistence of vision to create an illusion of movements, JLINK creates a tree structure to show the relationship. (WIP)

The underlying logic of JLINK is best understood when expressed as a graph which composed of two categories of relational associations which are termed nodes and edges. (Informational note: See Wikipedia for more details at https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)for further explanation.) In the context of JLINK, we have relationships which are centric to image-based experiences.

Element			Description
Symbols	JLINK	Graph	
		Image	An image is defined to be the combination of a code stream and metadata as allowed by ISO/IEC 19566.
	Scene	Node	The fundamental element of a node is an associated image; there must be at least one image per node, and this is the node root. Further, a node can have associations with other media elements (e.g., images, viewports, sprites, audio, video); there is no upper limit on the number of images and media elements associated to a node.

	Link	Edge	An edge defines a relationship between elements of one node to elements of another node. Further, an edge is associated with actions. Triggering an action, either by the viewer and/or automation, results in traversing the edge to the indicated node. It is allowed that the traversal begin and end on the same node without traversing through any other nodes. It is allowed that the transversal be bidirectional or unidirectional.

		Iteration	As ISO/ISC 19566 allows for embedding images iteratively (either directly and/or using URI references), and these embedded images may themselves compose graphs, such graphs are considered sub-graphs and to the parent node.

		Graph	A JLINK graph is the description of nodes and edges expressed syntactically (versus visually). There are two types of graphs: Fully Connected Graph: all node roots are connected to all other node roots by an edge. No actions are associated with the Fully Connected Graph, but an application may choose to associate an action to allow traversing from one node root to another. All graphs have a Fully Connected Graph representation which is constructed from the set of node roots; i.e., it is not necessary to provide the Fully Connected Graph description. Directed Graph: node roots and/or node elements are connected to other node roots and/or node elements through an associated action; when the action is triggered, the edge is traversed. A Directed Graph description is required to be provided. Multiple Directed Graph descriptions are allowed. When a subgraph is present, it has a Fully Connected Graph description by default, and may also contain multiple Directed Graph descriptions. The scope of these Fully Connected Graph and Directed Graph descriptions is limited to its level of embedding in the parent node.
	Plot Link Reference Link		

4 Conventions

4.1 Conformance language

This Recommendation | International Standard consists of normative and informative text.

Normative text is that text which expresses mandatory requirements. The word "shall" is used to express mandatory requirements strictly to be followed in order to conform to this Specification and from which no deviation is permitted. A conforming implementation is one that fulfils all mandatory requirements.

Informative text is text that is potentially helpful to the user, but not indispensable and can be removed, changed or added editorially without affecting interoperability. All text in this Recommendation | International Standard is normative, with the following exceptions: the Introduction, any parts of the text that are explicitly labelled as "informative", and statements appearing with the preamble "NOTE" and behaviour described using the word "should". The word "should" is used to describe behaviour that is encouraged but is not required for conformance to this Specification.

The keywords "may" and "need not" indicate a course of action that is permissible in a conforming implementation.

The keyword "reserved" indicates a provision that is not specified at this time, shall not be used, and may be specified in the future. The keyword "forbidden" indicates "reserved" and in addition indicates that the provision will never be specified in the future.

4.2 Operators

NOTE – Many of the operators used in this Recommendation | International Standard are similar to those used in the Extensible Markup Language(XML) language.

5 General

The purpose of this clause is to give an informative overview of the elements specified in this Specification. Another purpose is to introduce many of the terms which are defined in clause 3. These terms are printed in *italics* upon first usage in this clause.

5.1 JLINK Components

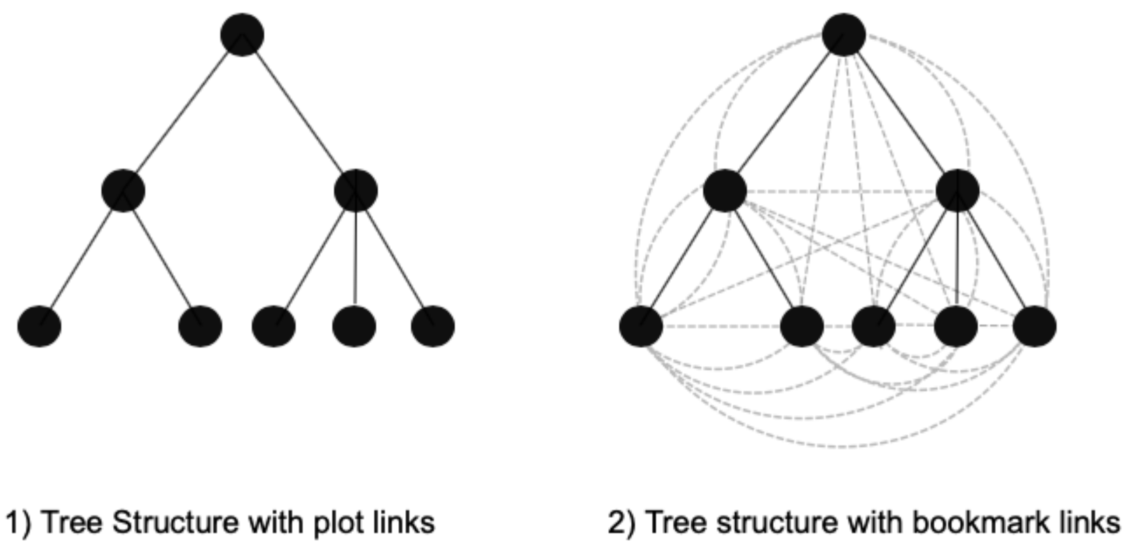


Fig. 2 Topology and linkage of JLINK

JLINK structures multiple related media into contents with one theme.

The basic unit that composes JLINK is called ‘Scene’. Scene makes a certain semantic piece by combining multimedia data such as images, text.

Scenes can be linked to other scenes according to their semantic relationship. The element that connects the scene to the scene is called ‘Link’.

Link expresses relationships between scenes and scenes through properties such as point, region, viewport, sprite, and transition effect, while used to indicate 'scene movement' (scene transition) from scene to other scene at the connected single scene.

5.2 JLINK Structure

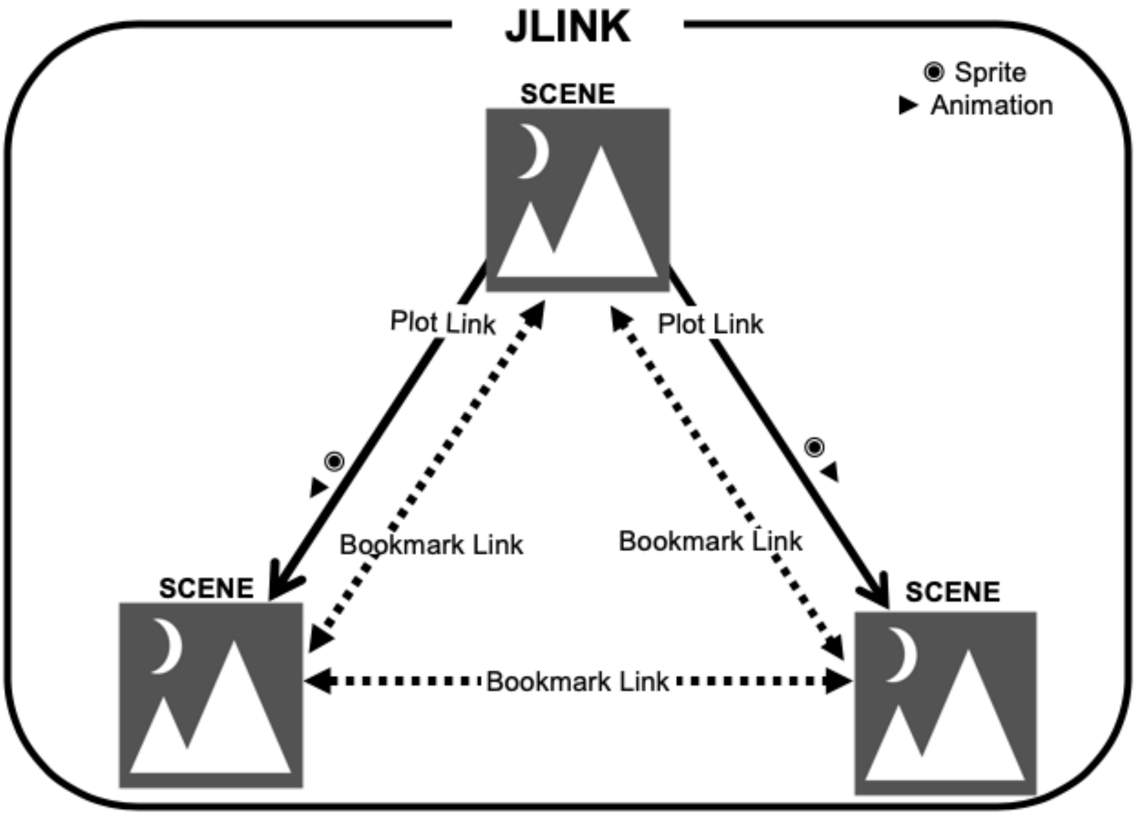
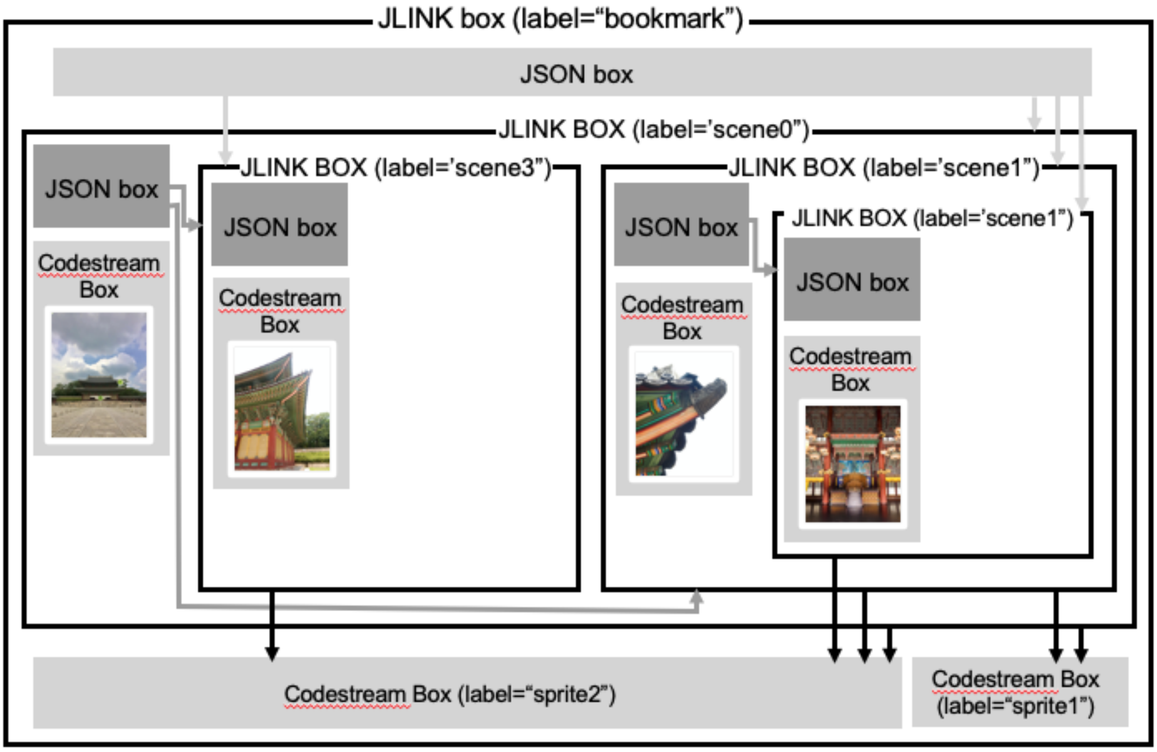
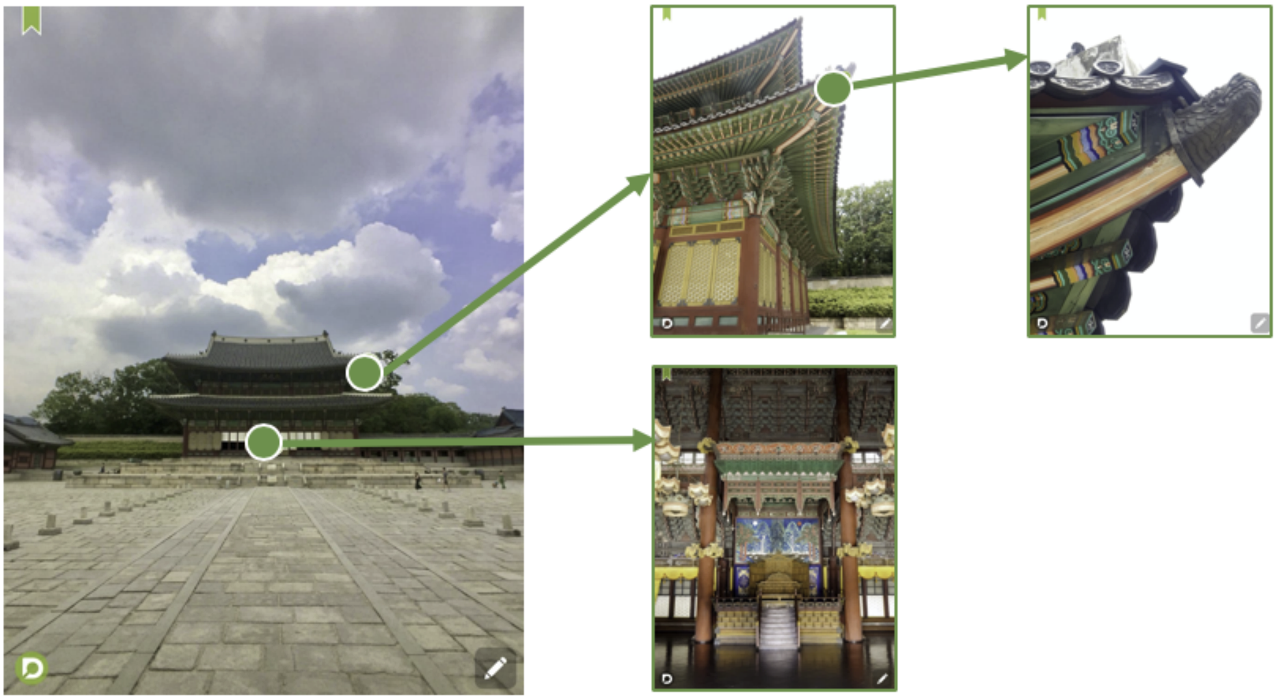


Fig. 3 Schematic diagram showing the components of a JLINK file and their relationship

JLINK forms a box that has the metadata and code stream that define the scene. The metadata of a scene possesses link information that connects to another scene, which is described in a way that points to the JLINK address to which the destination scene to be accessed belongs.

JLINK can be achieved layered structure that includes other JLINKs inside.


JLINK box (label="bookmark")
JSON box
JLINK BOX (label='scene0")
JSON box
Codestream Box
JLINK BOX (label='scene3")
JSON box
Codestream Box
JLINK BOX (label='scene1")
JSON box
Codestream Box
JLINK BOX (label='scene1")
JSON box
Codestream Box
Codestream Box (label="sprite2")
Codestream Box (label="sprite1")



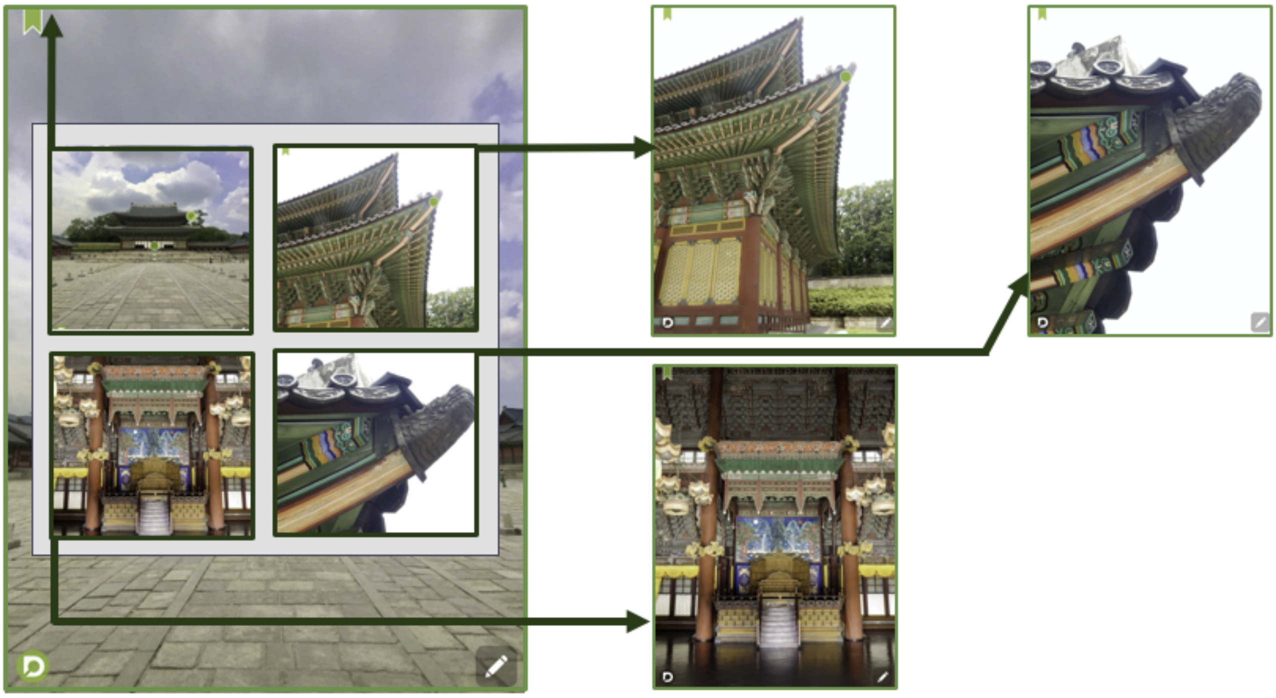


Fig. 4 Layered JLINK JUMBF box structure

TBC

5.3 Positioning of Linkage Point

5.3.1 Positioning of Linkage Point on 2D image

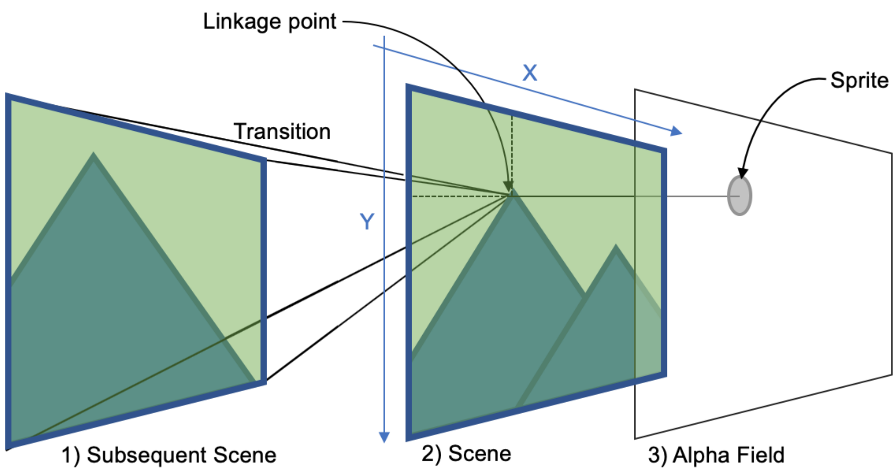


Fig. 5 Linked scenes of 2D images through linkage point

The linkage point of a 2D image is expressed by the real number of percent ratio from the left as X coordinate and the real number of percent ratio from the top as Y coordinate. (fig. 5)

5.4 Positioning of Region on 2D Image

The linkage region of a 2D image is expressed by the real number of percent ratio from the left as X coordinate, the real number of percent ratio from the top as Y coordinate, width and height in the real number percent ratio on the background picture.

5.5 Viewport on 2D Image

When navigating from one scene to another through a Plot Link or Reference Link connection, you can describe Viewport information in order to pay attention to a specific part of the latter scene to be displayed on the screen.

When the media to be presented on the screen is a 2D image, the center coordinates X and Y of the viewport are described as defined in 5.3.1 above.

In addition, the field of view span can be described in the real number of percent ratio for the X-axis and Y-axis directions with respect to the center, and it can be described the viewport rotation angle (°) in the positive direction of the X axis is 0 ° and the counterclockwise direction in the positive direction.

If the viewport is not described, the entire image without rotation is used as the viewport.

5.6 Sprite

When a scene and a scene are connected through a Plot Link or a Reference Link, Sprite information can be described in order to present to a user browsing interaction means that can move from one location of a scene to another..

Describes the image information, the width, and the height to be presented in Sprite as pixel unit

5.7 Transition Effect

When navigating from one scene to another through a Plot Link or Reference Link, it is possible to describe scene movement direction information moving from the former scene to the latter scene.

To be completed. (Is there IS for the transition effect we can refer to?)

5.8 Text for Scene

You can add a title and text for the scene.

(TBC)

6 File Position for JLINK metadata and codestreams

The JPEG standards have defined specific locations in the file structure for an image codestream. With the new usages supported by JLINK, images are included in a broader definition of metadata. It is possible to signal that the encoded image is located within JLINK Content Type box or is located in the file position of earlier standard. A simplified diagram of the file structure is shown in Figure 4.

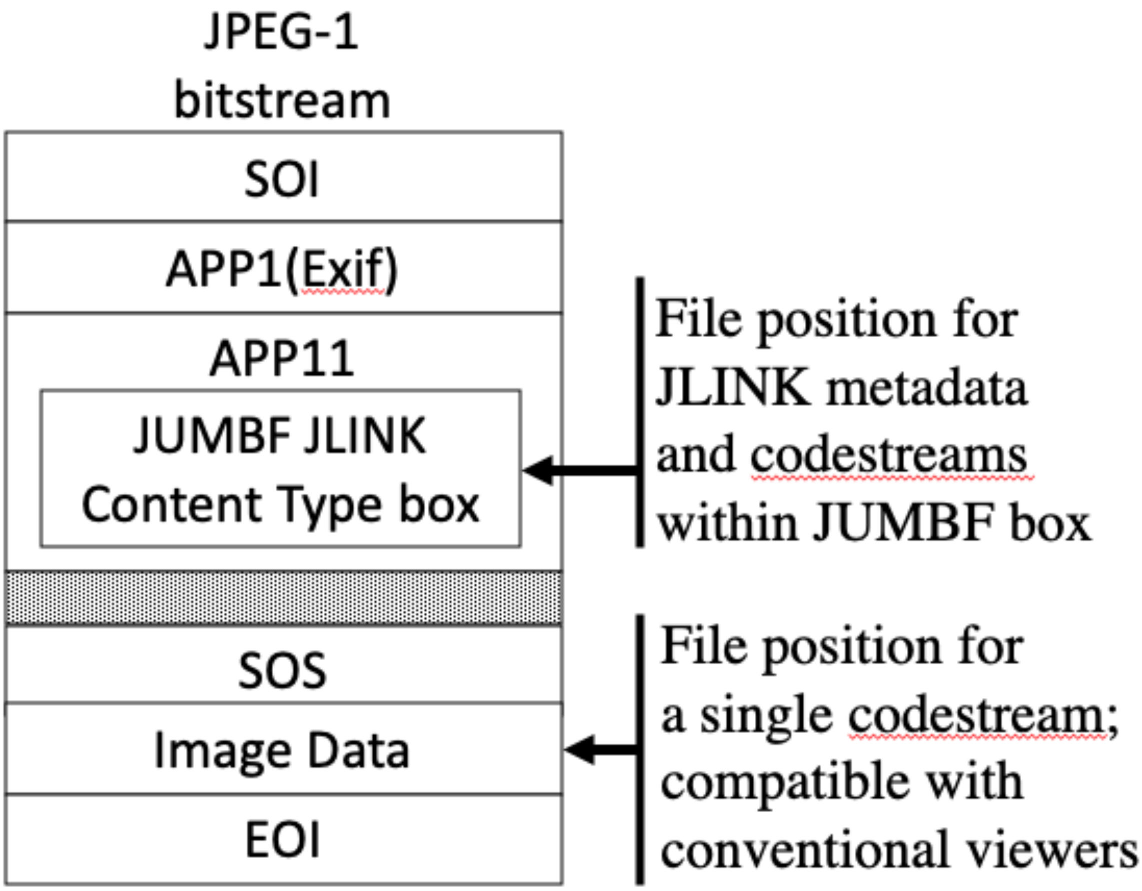


Fig. 6 File position for JLINK metadata and codestreams within JUMBF box

This "legacy" file position may be desirable so that an image is decoded when the file is opened by conventional JPEG viewing applications.

7 Structuring of JLINK Metadata

7.1 General

This Clause defines the JLINK Content Type box which is based on the JUMBF superbox defined by ISO/IEC 19566 5. The sub-box components are defined, which include the definition of an XML box, the use of nested JLINK Content Type boxes, and the use of Codestream Content Type boxes for images which are associated to scenes or sprites. An overview of the JLINK Content Type box is shown in Figure 5.

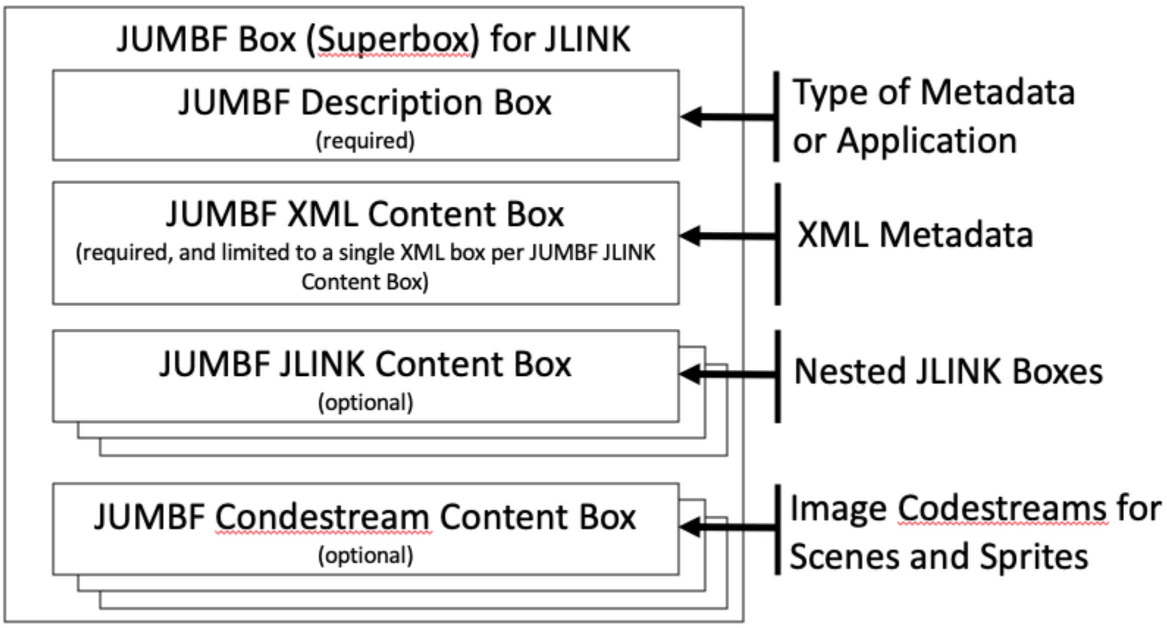


Fig. 7 Box diagram of JLINK Content Type box

7.2 Definition of JLINK Content Type boxes

With reference to ISO/IEC 19566 5, the JLINK Content Type box shall be defined in:

- Annex A, which contains the definition of the JUMBF box for JLINK ~~Story Container~~ (e.g, the structure and definitions of required content of this box),
- Annex B, which contains the definition of the JLINK ~~Story Container~~ metadata schema,

To be completed (Transition effect metadata will be added)

Annex A

JLINK Content Type JUMBF box

(This Annex forms an integral part of this Recommendation | International Standard)

A.1 General

This Annex defines the use of JUMBF elements for JLINK; the elements are defined in ISO/IEC 19566-5 JPEG Systems – Part 5: JPEG Universal Metadata Box Format.

Table A.1 JUMBF Description box: Type for JLINK

Parameter	Value
TYPE	0x[ky1] ????????-0011-0010-8000-00AA00389B71

Table A.2 JUMBF Description box: TOGGLES for JLINK

Binary mask	JLINK TOGGLE value	Toggle meaning
0000 0001	0 or 1	Value 1 for Requestable present and value 0 for not.
0000 0010	0 or 1	Value 1 for Label present and value 0 for not.
0000 0100	0 or 1	Value 1 for ID present and value 0 for not..
0000 1000	0 or 1	Value 1 for Signature present and value 0 for not.
All other values are reserved for future use.		

If Requestable of TOGGLE is set to 1 and the JLINK content metadata are represented with XML form, then the header of the response instance have 'text/XML' mime type. All content type except Story content type shall be value 1 for the ID of TOGGLE.

JUMBF Description box: recommended/default Label string for JLINK

Parameter	LABEL Value for JLINK when the Lablel value of TOGGLE is 1.
LABEL	"jlink"[2]

Description of the JLINK Content Type JUMBF box

As shown in Figure 5, JLINK metadata is contained within a JUMBF superbox. This JLINK Content Type JUMBF Box is internally composed of the following:

- **The JUMBF Descriptor Box for JLINK**
- **Exactly one XML Box which contains the JLINK Metadata as described in Annex B**
- **Optional JLINK Content Type JUMBF boxes.**
- **Optional Codestream Content Type JUMBF boxes.**

Annex B

Metadata (for JLINK)

(This Annex is an integral part of this Recommendation | International Standard)

B.1 General

The strategy behind this version of JLINK to balance the need to provide a timely standard while keeping flexibility to grow the feature sets to more fully accommodate the identified use cases. To meet this need, a basic schema descriptor and an empty metadata set are provided; which assign default values for the metadata elements.

B.2 Definition of JLINK metadata

The schema elements of the JLINK Metadata contains basic properties, described in Table B.2.1.

B.2.1 Definition of JLINK Metadata

JLINK Schema Descriptor Elements			Meaning	Data type
scene			Schema name.	string
jlinkVersion			JLINK Version Number	integer
title			Title of the scene.	string
text			Description of the scene.	string
image			URI or 32 byte ID Reference to the Codestream Content Type box.	string
links	– –		Name of subschema for link parameters.	string
	to		URI or 32 byte ID Reference to the JLINK Content Type box for destination scene.	string
	point	--	Name of subschema for point parameters.	
		x	X coordinate in percent ratio on the background picture.	real
		y	Y coordinate in percent ratio on the background picture.	real
	region	--		
		x	X coordinate in percent ratio on the background picture.	real
		y	Y coordinate in percent ratio on the background picture.	real
		width	Width in percent ratio on the background picture.	real
		height	Height in percent ratio on the background picture.	real
	sprite		URI or 32 byte ID Reference to the Codestream content box.	string
	viewportOnDestination	– –	Name of subschema for JPEG image parameters	string

	x	X coordinate of the center pixel in a viewport	real
	y	Y coordinate of the center pixel in a viewport	real
	xFOV	FOV of FOV in X direction with percent ratio on the background picture	real
	yFOV	FOV of FOV in Y direction with percent ratio on the background picture	real
	roll	Viewport rotation angle	real
transitionEffect		To be done…	

B.3 Overview of Metadata Representation

This Annex defines JLINK schema based on XMP specification which is also covered as part of ISO/IEC 16684, Graphic technology - Extensible metadata platform (XMP) specification: Data model, serialization and core properties. The data source layer implements serialization, deserialization and embedding of metadata with using XMP with the addition of new tags to express a schema descriptor, and to associate metadata elements with that schema descriptor.

Fig. 9 Logical view of the metadata

- The high-level structuring of the JLINK XML box is shown below.

XMP framing	Purpose	Top level structure and tags	Second level structure and tags
	Internal counter	umf:next-id	
	Schemas description storage	XMP arrays of schemas umf:schemas	XMP array of schema names umf:descriptors
			XMP array of fields/types umf:fields
	Metadata storage	XMP array of metadata umf:metadata	XMP array of metadata items umf:set
			XMP array of metadata field/values, references umf:fields, umf:refs

It should be understood that the top-level structure ordering is not critical as the structure below is equivalent to the structure above.

XMP framing	Purpose	Top level structure and tags	Second level structure and tags
	Internal counter	umf:next-id	
	Metadata storage	XMP array of metadata umf:metadata	XMP array of metadata items umf:set
			XMP array of metadata field/values, references umf:fields, umf:refs
	Schemas description storage	XMP arrays of schemas umf:schemas	XMP array of schema names umf:descriptors
			XMP array of fields/types umf:fields

B.4 JLINK rules/access control

Metadata for JLINK rules or access control provides the necessary information to guarantee linked image. For this purpose, the following sub-clauses specify how to:

- Express linked image rules defining the kind of linkage required and over what, to whom, and under which conditions, access is to be granted.
- Integrate linked image rules in the management of JPEG images, establishing at least an association mechanism between rules and images.
- Manage mechanisms for both metadata and content linkage, including an authorization process over the rules and a systems' architecture.

B.5 Values for JLINK metadata

The schema descriptor elements of JLINK metadata are shown in Table B.5.1.

B.5.1 Values for JLINK Metadata

JLINK Schema Descriptor			**Schema Descriptor**
scene			**<umf:schema>scene</umf:schema>**
jlinkVersion			**<umf:name>jlinkVersion</umf:name>** **<umf:type>integer</umf:type>**
title			**<umf:name>title</umf:name>** **<umf:type>string</umf:type>**
text			**<umf:name>text</umf:name>** **<umf:type>string</umf:type>**
image			**<umf:name>image</umf:name>** **<umf:type>string</umf:type>**
links	**--**		**<umf:schema>links</umf:schema>**
	to		**<umf:name>to</umf:name>** **<umf:type>string</umf:type>**
	point	**--**	**<umf:schema>point</umf:schema>**
		x	**<umf:name>x</umf:name>** **<umf:type>real</umf:type>**
		y	**<umf:name>y</umf:name>** **<umf:type> real </umf:type>**
	region	**--**	**<umf:schema>region</umf:schema>**

	x	<umf:name>x</umf:name> <umf:type> real </umf:type>
	y	<umf:name>y</umf:name> <umf:type> real </umf:type>
	width	<umf:name>width</umf:name> <umf:type> real </umf:type>
	height	<umf:name>height</umf:name> <umf:type> real </umf:type>
sprite		<umf:name>sprite</umf:name> <umf:type>string</umf:type>
viewportOnDestination	−−	<umf:schema>viewportOnDestination</umf:schema>
	x	<umf:name>x</umf:name> <umf:type>real</umf:type>
	y	<umf:name>y</umf:name> <umf:type> real </umf:type>
	xFOV	<umf:name>xFOV</umf:name> <umf:type>real</umf:type>
	yFOV	<umf:name>yFOV</umf:name> <umf:type> real </umf:type>
	roll	<umf:name>roll</umf:name> <umf:type>real</umf:type>

	transitionEffect	**To be done…**

B.6 JLINK Metadata Syntax

The JLINK schema is serialized and stored using a subset of the W3C Resource Description Framework (RDF), expressed in XML

The XMP statement to frame a Metadata Stream is:

```
<x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="XMP Core 5.4.0">
   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
       <rdf:Description rdf:about="" xmlns:umf="http://ns.intel.com/umf/2.0">
       <!- internal structure -->
       </rdf:Description>
   </rdf:RDF>
</x:xmpmeta>
```

A.1 Reserved tags

Table B.2 defines tags reserved in the JLINK XML Box.

Table B.2 — JLINK reserved XML tags

Reserved Tag	Description
umf:schemas	Tag to define start of schema descriptor. Multiple schema descriptors can be defined, each with multiple sets of metadata elements.
umf:schema	Tag to define a unique name for a schema.
umf:name	Tag to define name of schema descriptor elements, or to label sets of metadata elements.
umf:type	Tag to associate a data type to a name in the schema descriptor.
umf:id	Unique id of a metadata set.
umf:next-id	Used by implementation(s) to assign a unique id to metadata items.
umf:index	Reserved tag. Index into an image sequence that contains the first image associated with this metadata item (-1 in case of global metadata).
umf:nframes	Reserved tag. Number of sequential frames the metadata item is associated with (0 in case of global metadata).
umf:fields	Tag for description of schema fields. Fields of metadata items provide a list of value/name pairs (in case it is a structure) or an array of values (in case it is an array) or just a single value.
umf:refs	Array of references to other metadata items. Each reference is stored as metadata id.
umf:set	Tag to associate fields of metadata elements.

A.2 XMP expression of minimum self-describing schema (without metadata elements)

The JLINK XML Box contains data which is structured using XMP; the XMP expression of the schema defines its properties.

The minimal XMP expression consists of the JLINK schema descriptor and an unpopulated JLINK metadata storage, as shown below. When this minimal expression is provided, a number of default values for the JLINK description are assigned; these default values are detailed in B.8.

```
http://ns.adobe.com/xap/1.0/ <?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">
<!-- JLINK Metadata -->
<umf:schema>scene</umf:schema>
<umf:descriptors>
<rdf:Bag>
<rdf:li rdf:parseType="Resource">
<umf:name>jlinkVersion</umf:name>
<umf:type>integer</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>title</umf:name>
<umf:type>string</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>text</umf:name>
<umf:type>string</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>image</umf:name>
<umf:type>string</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>links</umf:name>
<umf:fields>
<rdf:Bag>
<rdf:li rdf:parseType="Resource">
<umf:name>to</umf:name>
<umf:type>string</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>point</umf:name>
<umf:fields>
<rdf:Bag>
<rdf:li rdf:parseType="Resource">
<umf:name>x</umf:name>
<umf:type>real</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>y</umf:name>
<umf:type>real</umf:type>
</rdf:li>
</rdf:Bag>
</umf:fields>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>region</umf:name>
<umf:fields>
<rdf:Bag>
<rdf:li rdf:parseType="Resource">
<umf:name>x</umf:name>
<umf:type>real</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>y</umf:name>
<umf:type>real</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>width</umf:name>
<umf:type>real</umf:type>
</rdf:li>
```

```
<rdf:li rdf:parseType="Resource">
<umf:name>height</umf:name>
<umf:type>real</umf:type>
</rdf:li>
</rdf:Bag>
</umf:fields>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>sprite</umf:name>
<umf:type>string</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>viewportOnDestination</umf:name>
<umf:fields>
<rdf:Bag>
<rdf:li rdf:parseType="Resource">
<umf:name>x</umf:name>
<umf:type>real</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>y</umf:name>
<umf:type>real</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>xFOV</umf:name>
<umf:type>real</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>yFOV</umf:name>
<umf:type>real</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>roll</umf:name>
<umf:type>real</umf:type>
</rdf:li>
</rdf:Bag>
</umf:fields>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>transitionEffect</umf:name>
<umf:fields>
<rdf:Bag>
<!-- TO BE COMPLETED -->
</rdf:Bag>
</umf:fields>
</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>scene</umf:schema>
<umf:set>
</umf:set>
</rdf:li>
</rdf:Bag>
```

```
</umf:metadata>

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

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

A.3 Default JLINK image values

In the XMP expression in in the previous section, only the JLINK schema was provided, but no metadata element values were assigned. In this case, the default values for the metadata elements are as shown in Table B.4.

Table B.4 — Default metadata element values

JLINK Schema Metadata Elements			Default value when no metadata elements present
scene			Not applicable.
jlinkVersion			1
title			“”
text			“”
image			“”
links	––		Not applicable.
	to		“”
	point	--	Not applicable.
		x	0
		y	0
	region	--	Not applicable.
		x	0
		y	0

	width	**0**
	height	**0**
sprite		**“”**
viewportOnDestination	**– –**	**Not applicable.**
	x	**50**
	y	**50**
	xFOV	**50**
	yFOV	**50**
	roll	**0**
transitionEffect		**To be done…**

A.4 Example XMP expression with populated metadata elements

```
http://ns.adobe.com/xap/1.0/<?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">
<!-- JLINK Metadata -->
<umf:schema>scene</umf:schema>
<umf:descriptors>
<rdf:Bag>
<rdf:li rdf:parseType="Resource">
<umf:name>jlinkVersion</umf:name>
<umf:type>integer</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>title</umf:name>
<umf:type>string</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>text</umf:name>
<umf:type>string</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>image</umf:name>
<umf:type>string</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>links</umf:name>
<umf:fields>
<rdf:Bag>
<rdf:li rdf:parseType="Resource">
<umf:name>to</umf:name>
<umf:type>string</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>point</umf:name>
<umf:fields>
<rdf:Bag>
<rdf:li rdf:parseType="Resource">
<umf:name>x</umf:name>
<umf:type>real</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>y</umf:name>
<umf:type>real</umf:type>
</rdf:li>
</rdf:Bag>
</umf:fields>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>region</umf:name>
<umf:fields>
<rdf:Bag>
<rdf:li rdf:parseType="Resource">
<umf:name>x</umf:name>
<umf:type>real</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>y</umf:name>
<umf:type>real</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>width</umf:name>
<umf:type>real</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>height</umf:name>
<umf:type>real</umf:type>
</rdf:li>
</rdf:Bag>
```

```
</umf:fields>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>sprite</umf:name>
<umf:type>string</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>viewportOnDestination</umf:name>
<umf:fields>
<rdf:Bag>
<rdf:li rdf:parseType="Resource">
<umf:name>x</umf:name>
<umf:type>real</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>y</umf:name>
<umf:type>real</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>xFOV</umf:name>
<umf:type>real</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>yFOV</umf:name>
<umf:type>real</umf:type>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>roll</umf:name>
<umf:type>real</umf:type>
</rdf:li>
</rdf:Bag>
</umf:fields>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>transitionEffect</umf:name>
<umf:fields>
<rdf:Bag>
<!-- TO BE COMPLETED -->
</rdf:Bag>
</umf:fields>
</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>scene</umf:schema>
<umf:set>
<rdf:Bag>
<rdf:li rdf:parseType="Resource">
<rdf:value>1</rdf:value>
<umf:name>jlinkVersion</umf:name>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<rdf:value>""</rdf:value>
<umf:name>title</umf:name>
```

```
</rdf:li>
<rdf:li rdf:parseType="Resource">
<rdf:value>""</rdf:value>
<umf:name>text</umf:name>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<rdf:value>""</rdf:value>
<umf:name>image</umf:name>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>links</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>""</rdf:value>
<umf:name>to</umf:name>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>point</umf:name>
<umf:set>
<rdf:Bag>
<rdf:li rdf:parseType="Resource">
<umf:id>2</umf:id>
<umf:fields>
<rdf:Bag>
<rdf:li rdf:parseType="Resource">
<rdf:value>0</rdf:value>
<umf:name>x</umf:name>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<rdf:value>0</rdf:value>
<umf:name>y</umf:name>
</rdf:li>
</rdf:Bag>
</umf:fields>
</rdf:li>
</rdf:Bag>
</umf:set>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>region</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>0</rdf:value>
<umf:name>x</umf:name>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<rdf:value>0</rdf:value>
<umf:name>y</umf:name>
</rdf:li>
```

```
<rdf:li rdf:parseType="Resource">
<rdf:value>0</rdf:value>
<umf:name>width</umf:name>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<rdf:value>0</rdf:value>
<umf:name>height</umf:name>
</rdf:li>
</rdf:Bag>
</umf:fields>
</rdf:li>
</rdf:Bag>
</umf:set>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<rdf:value>""</rdf:value>
<umf:name>sprite</umf:name>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>viewportOnDestination</umf:name>
<umf:set>
<rdf:Bag>
<rdf:li rdf:parseType="Resource">
<umf:id>4</umf:id>
<umf:fields>
<rdf:Bag>
<rdf:li rdf:parseType="Resource">
<rdf:value>50</rdf:value>
<umf:name>x</umf:name>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<rdf:value>50</rdf:value>
<umf:name>y</umf:name>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<rdf:value>50</rdf:value>
<umf:name>xFOV</umf:name>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<rdf:value>50</rdf:value>
<umf:name>yFOV</umf:name>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<rdf:value>0</rdf:value>
<umf:name>roll</umf:name>
</rdf:li>
</rdf:Bag>
</umf:fields>
</rdf:li>
</rdf:Bag>
</umf:set>
</rdf:li>
<rdf:li rdf:parseType="Resource">
<umf:name>transitionEffect</umf:name>
<umf:set>
<rdf:Bag>
<rdf:li rdf:parseType="Resource">
<umf:id>5</umf:id>
<umf:fields>
```

```
<rdf:Bag>
<!-- TO BE COMPLETED -->
</rdf:Bag>
</umf:fields>
</rdf:li>
</rdf:Bag>
</umf:set>
</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"?>
```

[ky1]How should we determine the UUID value for JLINK Content Type?

I prefer to use any text string for this LABEL.

Annex A
Backward compatibility

(This Annex is informative only and is not an integral part of this Recommendation | International Standard)

A.1 General

This Annex defines

Describes backwards compatibility.

JPSearch

The JLINK is not backwards compatible with the JPSearch Part 4 File Format. Legacy decoders that are compliant with JPSearch Part 4 will not be able to read the metadata embedded according to the JLINK specification. JPSearch Part 4 and JLINK metadata can co-exist in a single image instance. However, this is not recommended because it can lead to inconsistencies between the duplicated metadata instances. Therefore it is recommended to convert existing JPSearch Part 4 embedded metadata to the JLINK.

JPEG / JPEG XT

Legacy decoders will skip and not be aware of the added metadata. Conform decoders can read the metadata.

JPEG2000

Decoders that do not comply with the JLINK specification will skip the JLINK boxes.

Annex B
Implementation guidelines

(This Annex is informative only and is not an integral part of this Recommendation | International Standard)

B.1 General

This Annex defines describes a variety of use cases and requirements of the metadata for JLINK images (defined below) that is included within the image file needed to support these use cases. It is expected that use cases will evolve over time, and that requirements will be staged over time. The use cases bring out certain interactions with JLINK images that are considered relevant and interesting, but without being prescriptive with respect to implementation.

B.1.1 Engineering Smart Assistant

Tom is an engineer in charge of structural engineering for reinforced concrete. As there are many aged reinforced concrete constructions to be regularly inspected, he really feels that it is necessary to raise the efficiency of inspection. Tom has been storing a huge number of sample pictures regarding deterioration of concrete constructions. As a new technology trend, Tom has heard that advances in image recognition technology will become practical on mobile devices. He decided to take in the new technology to improve its effectiveness. Tom took pictures with his smart phone, being careful to capture them with high fidelity to improve the analyses. Below is a sample picture applied for a dam, but the images and analyses are similar for tunnels and bridges.

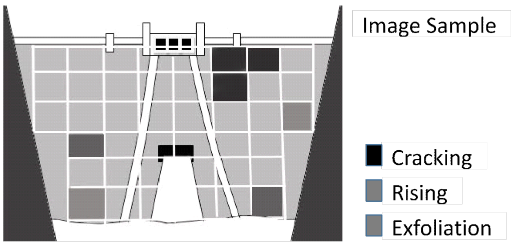


Fig. 10 Engineering Smart Assistant

B.1.2 Memento Photo Collage

Alice has returned to Sydney after being a bridesmaid at her best friend's destination-wedding in Cusco, Peru. Alice had a great time at the wedding, and really enjoyed a self-guided tour of the city using a smart guide developed by a Cusco native. Alice added a few days to visit Machu Pichu, and added a few days in Lima to explore the culture, food, nightlife, and beaches. Alice is very taken with the traditional Peruvian music, and also really liked a couple popular songs on the radio, so she downloaded those for her music collection.

On the long flight back to Sydney, Alice went through her photos and created a collage of photos and used the newly acquired music as the collage soundtrack that would play when Alice played the collage. The application she used to create the collage also allowed Alice to link to the full resolution image version of each of the photo tiles in her collage. The entire memento photo collage, including high resolution photos and soundtrack, was bundled into a single JPEG file so that it could be easily shared to her electronic photo frame and share the memento with friends; everyone could at least see the photo collage, and those with newer photo frames got to hear the soundtrack and enjoy the full-screen versions of the photos.

Fig. 11 Memento Photo Collage

B.1.3 Navigating in Omnidirectional Images

Heidi captures a 360 picture of her grandfather's house and garden. Later, when she reviews and edits the image, Heidi selects a region of interest (ROI) within the omnidirectional photo which can be viewed using a legacy JPEG viewer. Heidi also adds the default position the viewer should see when opening the 360 image. She sends this composite image to her friend Klara as an email attachment.

Klara receives Heidi's email. Klara is able to view Heidi's the ROI portion of photograph with a conventional JPEG viewer on her PC. However, her PC's photo viewer application supports the JPEG 360 photos, so when she uses that application, Klara can choose to view either the ROI image or the 360 image. Klara's phone also has a photo viewer application that supports JPEG 360 photos. Klara looks at Heidi's photo with using her smartphone., and then decides to switch to her PC to use her head mounted display (HMD) to have a more immersive and enjoyable experience which brings back memories of her life in the Alps. Klara decided that in the future, she wants to open the 360 image to a different default view, and she edits this property of the image. If she wants, Klara could change the ROI image by having the 360 photo editor replace it with her preference.

Heidi emails the URL of the omnidirectional photo to her childhood friend Peter, so Peter can access it with a 360 web-viewer using a browser on his computer. He can interact with the picture naturally without any noticeable delays because the compression technology optimizes the transmission efficiency.

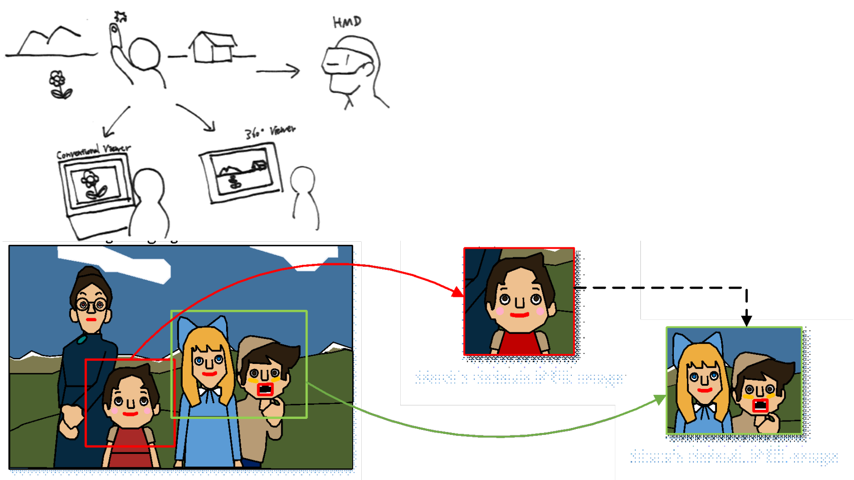


Fig. 12 Navigating in Omnidirection Images

B.1.4 Adding additional media data to omnidirectional images

Mr. Urashima Tarō captures several pictures in a domed garden in the Ryugu castle with his 360 camera and also with a high resolution DSLR camera. He captures images of the same scene in the garden with different resolutions, and he also captures videos of a waterfall and a river which include audio. After collecting several images and videos, he embeds this image and video data into a 360 JPEG file, and he makes an omnidirectional image with multiple media data.

Tarō send his omnidirectional photograph data to Ms. Oto-hime. Oto-hime checks it with her 360 viewer (e.g. HMD, handheld device, etc.). She looks in all the directions of the omnidirectional image. She enlarges a particular area of his omnidirectional photo image, and she can see some zoom-upped area in a higher resolution where high resolution photo data exist. When she picks several image areas such as waterfall area of his photo, she watches a moving picture of that part and listen to the sound of it where the video and audio data exist.

Tarō stores several enhanced omnidirectional images in the cloud. It goes fast, because the files are optimized to reduce the amount of data transfer. Momotarō, Tarō's old friend, can review the photos without waiting for downloading of large amounts of data.

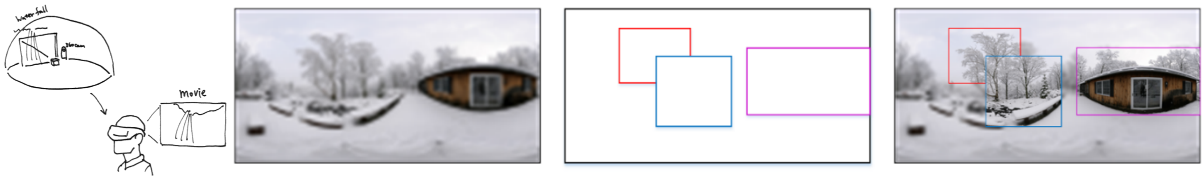


Fig. 13 Adding additional media data to omnidirectional images

B.1.5 Time-Lapse Omnidirectional Animation

Ms. Rottenmeier captures a 360 picture of an aurora scene every five minutes. Then she makes an animated omnidirectional image of aurora. She also captures a 360 picture of a Japanese garden every day during the year, which allows her to compose an animated 365 days time-lapse image of the Japanese garden. After completing her collection of several omnidirectional scenes, she puts this data on her website.

Peter browses Ms. Rottenmeier's website by looking at the still photo images, and he decides to see her aurora's animated image data. After accessing the complete image data, he can play the animated aurora omnidirectional image with his HMD and 360 viewers on his PC and his smart phone. He enjoys Rottenmeier's omnidirectional animation of aurora.

Fig. 14 Time-Lapsed Omnidirectional Animation

B.1.6 Computer-Savvy Tour Guide

Francesca earns a living by guiding walking tours through Padova; she's a native to Padova, and has been fascinated by its sights and history since she was a little girl. She offers an application that can be used by tourists for self-directed tours, and another application that can add labels and annotations to images captured by tourists after-the-fact. To create her catalogs, Francesca walked around with a 360 camera to the highlights she wants to capture, and then captures an image that includes the location data from her trekking GPS which is more reliable and accurate than the GPS in her phone.

When Francesca gets back home, she uploads her images onto her personal computer, and organizes them into catalogs for walks of different length of time, topics of interest (history, architecture, food & wine, restaurants, music, etc.); each image is tagged to at least one catalog, and often to several catalogs. Then Francesca opens each image and adds explanatory comments and image labels to call out the interesting facts about each place. The application then derives a signature for each of the tagged visual elements in the images, and using the GPS data, calculates positional data from which the tagged visual element was observed. Then Francesca adds links to her images, and links to additional online information and images for further details. Finally, the application extracts the positional data, image labels, image element signatures, and the Francesca's comments, and the additional links into a metadata set for each catalog that she offers, and copyright and authentication metadata is added. When a catalog is purchased, a customer-specific encryption is applied to the catalog; the key is provided separately.

B.1.7 Erudite Tourist

Gunther feels that wandering the city, while enjoyable, would be more enjoyable with the advice of guide, but Gunther dislikes large guided tours, and would rather spend his budget on a nice meal than getting personal tour guide. Lucky for Gunther, he noticed Francesca's self-guided tour, and he bought two of her catalogs: i) the two hour historical tour of central Padova, and ii) her latest guide to street food in Padova. Both catalogs contain very similar information, including visual signatures for the food served by the street vendors. Francesca's metadata catalogs are uploaded to his phone and his personal computer.

Gunther borrows a 360 degrees panorama camera from his friend François to use on his vacation in Italy. He straps it to his back and starts wandering Padova. The camera captures his walk by taking an image every 30 seconds, or when Gunther presses a button on a remote control in his hand in case something catches his interest. The camera also collects the time and position from the mobile phone Gunther is carrying, as well as the 6DoF and compass data from the camera sensor.

As Gunther walks, a mapping application shows him Francesca's tour path based on the GPS data from Francesca's file. He can stop anytime, take a side street, skip a sight, or stop for coffee, and then continue at his own pace.

When Gunter finishes the walking tour, he uploads the images to his personal computer. Using the sensor metadata captured as we walked around, and combining with Francesca's metadata in her catalog, the appropriate metadata is carried over from Francesca's catalog into Gunther's images, though some of it must be recomputed due to positional differences of the observations visual high lights. This merge/update of metadata become of Gunther's images. It might also include Francesca's copyright for her commentary. Gunther can decide whether or not to keep Francesca's link to additional on-line information about the sights – probably will for now since he may want to learn more after he returns home from his vacation.

B.1.8 Multiple Resolutions Omnidirectional Images

Mr. Schmitz captures several 360 pictures of the scene in different resolution with his omnidirectional camera. The camera then fuses these images into an omnidirectional image file supporting multiple resolution omnidirectional data. Mr. Schmitz zooms in and out the omnidirectional image he has just taken, and he can see that his omnidirectional picture has functionality of zooming in any direction. He stores it on his web site.

Mr. Momotarō is having a look at Mr. Schmitz's photos available online with an HMD viewer. Adolf's web site offers low resolution photo image quality when browsing through his collection of omnidirectional photographs. Momotarō picks several photos and decides to access the full data of these photos. He can also display high resolution photos with his 360 degree viewing devices.

B.1.9 Layered 360 Images

Miss Orihime tooks many 360° pictures at almost the same time in many locations, and then she puts these all 360° pictures to as single format. Mr. Kintarō take a look to Orihime's photographic image with his smartphone, and he find one house in her photo. He wanted to know what is behind the house, he use the transparent mode and select the house, and then the display application erased the selected house from the scene. After the house disappears, he can see a classic car behind it.

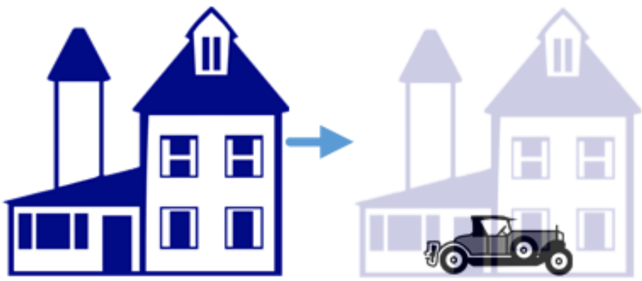


Fig. 15 Layered 360 Images

B.1.10 Linked 360 Images

Mr. Kintarō checks the 360° image rooms on some real estate's website for renting a house in the Shibuya area. He selects one picture and captures a look at the room with his HMD. He finds a door in the room. He clicks on the door, and his point of view moves to the next room which is a kitchen. He looks in several directions in the kitchen and he finds that there is a cabinet. After he clicks on the cabinet door, the cabinet door opens and he can check the cabinet inside.

Figure 3.11 Linked 360 Images

B.1.11 Guided Viewport Experience

Mr. Kitarō tooks several spherical photographs, and he decides to add information to the photograph about displaying it in his favorite way. With this information other people can see his photograph in his recommended viewing way. He also puts three thumbnails of his most favorite scenes to the captured spherical image for instant reviewing with web browser, so everyone can immediately access his favorite scenes.

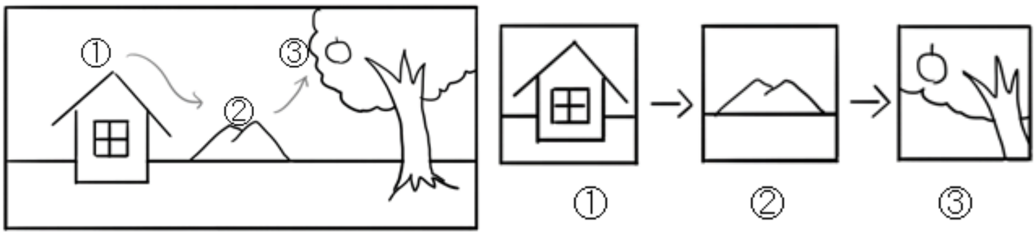


Figure 3.14 Guided Viewport Experience

B.1.12 Virtual Museum Tour with Explanatory Materials

Miss Neko-Musume goes to the web site of the National Museum to see modern art from her convenient location. Using her HMD, she goes through and see several pieces of art in the rooms of the museum via real/virtual corridor which connects the rooms. One picture catches her eyes. She wants to know what this painting art is, and she decides to display a detailed commentary of the picture. Then an explanatory note appears on the wall next to the picture, and she is able to know more about the painting art.

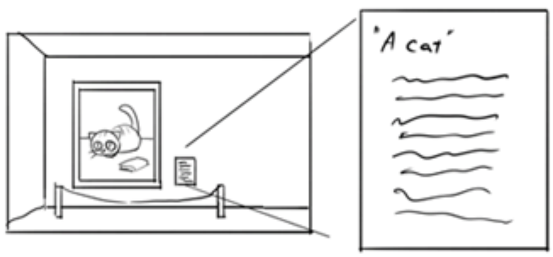


Fig. 16 Virtual Museum Tour with Explanatory Materials

B.1.13 Camera Position and Orientation

Mr. Kitarō captures several photographs using his 360° camera which stores its shooting position, direction and time. With this data he can make a relative map easily. And he is also able to see which photograph is taken at which position and when.

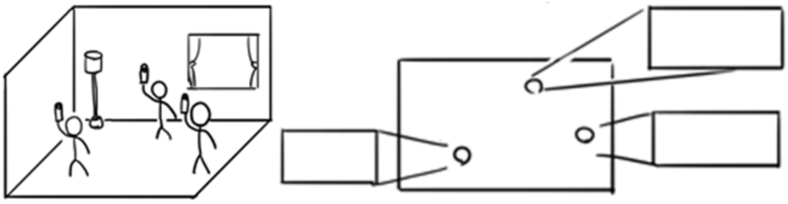


Fig. 17 Camera Position and Orientation

B.1.14 Multiple Type Sensor Images

Miss Orihime has a smart phone with dual type image sensors which can capture coloured image and infrared image. These image sensors are arranged in parallel so that the same object can be taken at the same time. Her smartphones can make wonderful photos from these images, however she saved these sensed image as one file format, and she apply other algorithms and sent it to the web service.

B.1.15 In-image Face Tagging for Photos

The photographer Jean Stringer takes a 360 degree photo for the photo department of the news agency AllNewsNow at a meeting of the heads of the 28 EU countries in Brussels. The 26 persons present at the time of the shot are seated around a table of elliptical shape and the 360 degree camera is placed in the middle of it. (Note: the illustration image does not meet this position, it is placed at the edge of the table.)

The photo editor of AllNewsNow has to add the names of the heads-of-state as metadata. As the typical viewer software of 360 degree photos opens only a limited viewport to the spherical view the photo editor wants to associate the name of a person with her or his face individually. First he draws a virtual rectangular numbered frame around the face of each head-of-state looking towards the camera in the 360 degree photo.

Figure 3.20.1 Captured Photo with Face Position Indications

Then he adds the name to the Person Shown metadata field of each numbered frame.

AllNewsNow Photo/Video Metadata Editor's Desk

Frame	Metadata of that frame
#1	Person shown: Ms Aname
#2	Person shown: Mr Cname
#3	Person shown: Mr Hname
#4	Person shown: Mr Kname
#5	Person shown: Mr Lname
#6	Person shown: Mr Mname
#6	Person shown: Mr Tname
#8	Person shown: Mr Uname
#9	Person shown: Mr Zname
#10	

Figure 3.20.2 Association of Person's Name to Frame Number

The information of each frame is transformed into an individual set of XMP metadata and all sets are embedded into the JPEG photo file.

A client of AllNewsNow launches a 360-degree-photo-viewer software to have a look at this JPEG photo. While moving the viewport across the sphere she can click on the M key or touch the screen of the tablet or smartphone and the viewer software shows the names next to the faces which are currently shown:

View in one direction …

Figure 3.20.3 FOV in One Direction

… view in another direction:

Figure 3.20.4 FOV in A Second Direction

… view in another direction:

Figure 3.20.5 FOV in A Third Direction

This definitely helps to distinguish which faces out of the many heads-of-state around the table are currently shown in the viewport and this improves the acceptance by consumers, and this improves the business value of 360-degree photos with metadata for different regions inside it.

AllNewsNow highly appreciates a very similar way of editing metadata for frames inside a 360-degree video.

IPTC considers it as key technical requirement for metadata associated with virtual frames to have a standardized reference point or reference coordinates system for a 360-degree spherical view. This is the origin for the definition of other points in this sphere relative to it. These points can be used to define such virtual rectangles.

B.1.16 Point to Point Move

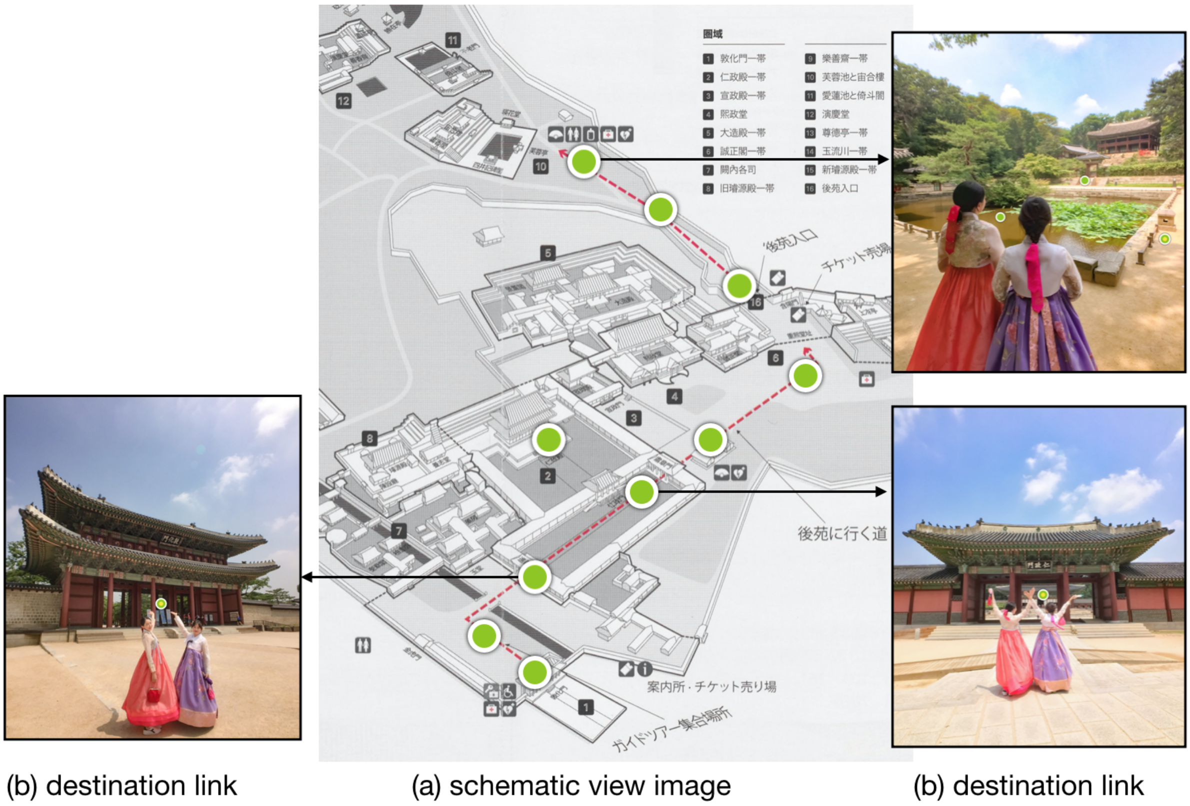


Fig. 18 Poin to Point move in JLINK

When viewing the linked images of Changdeokgung Palace, Mrs. Park follows the links as if she is taking a walk in the palace. However, browsing via links forces her to move sequentially, *i.e.*, a picture at a time, makes it timeconsuming to view an image located at the end of the link. Mrs. Park uses 'point-to-point move' (Fig. 3.22-a) to skip the images in between by choosing the 'destination link' (Fig. 3.22-b) from the 'schematic view.' While the 'schematic view' in Mrs. Park's case is a map (Fig. 3.22), it can be a floor plan of a house, tree structure of the linked images, *etc.*

B.1.17 Bookmarks as Reference Links

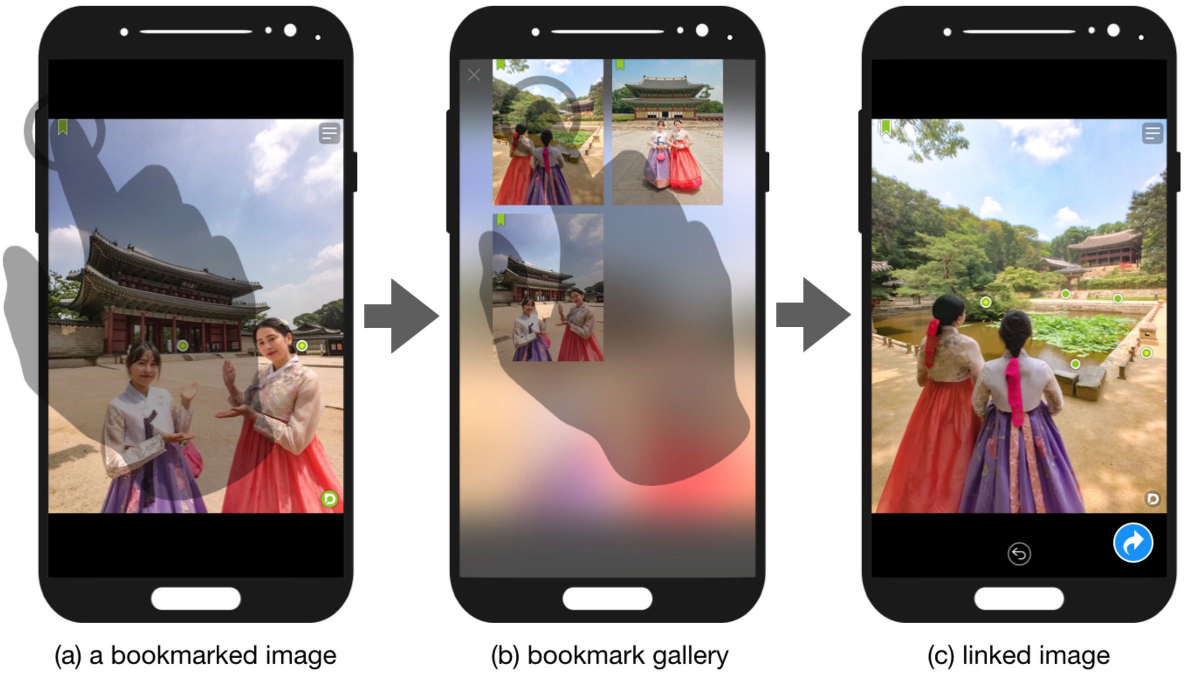


Fig. 19 Usage of Bookmarks as Reference Links

Mr. Park created a linked image of Changdeokgung Palace using 2D and 360-degree images. The linked image is comprised of over a hundred photographs, covering the 57.2 ha of the area filled with historic buildings. When sending the linked image to Mrs. Park, bookmarks were added to accentuate the pictures that Mr. Park particularly wants her to see.

With the received linked image, Mrs. Park is able to perceive Mr. Park's intention when the bookmark is shown in the picture (fig. 3.21-a), view the bookmarked pictures instantaneously (fig. 3.21-b) from the 'bookmark viewer', and jump to the desired image from the bookmark viewer (fig. 3.21-c). She also can add bookmarks to the 2D and 360 degree photos and that are interesting to her.

Requirements

• To specify the relative location of tagged image elements a standardized origin is needed.

Annex C

JLINK Requirements

This section lists the set of metadata functional requirements determined by the JPEG Committee

C.1.1 Metadata Functional Requirements

Metadata Requirements and Description		Applicability to JLINK
Store multiple images within a single file	**The standard shall support the ability to declare and discover the association to multiple images within the same file.** **The current JPEG file definition allows for only a single image to be contained within it, whereas for JPEG 360, there is need to keep multiple images within the same file. An example are the images from individual cameras which are stitched together to form a single complete spherical (or semi-spherical) image. Another example is a conventional (flat) higher-resolution image (or a pair of such images for the stereoscopic case) which is linked to a region of pixels on a 360 image.**	**Yes, partial for more sophisticated associations.**
Projection type	**The standard shall support the ability to declare and discover the projection type for images contained in the file. It must also support declaration and discovery of the spans of the image within the projection.** **Several types of representations (projections) are used to map an omnidirectional image to a 2D planar surface (e.g, equirectangular, cubic, pyramidal, dodecahedron, etc). The projection mapping needs to be known with sufficient detail for it to be possible to render an omnidirectional image. This can include geometry description (mesh) and texture coordinates.** **Images may be only a portion of a projection. E.g., a 180 image is a hemisphere. Further, during editing, an image can become an even smaller portion of the projection, as might be the case in a photo collage onto a spherical projection.**	**No**
File format of the additional images	**The standard shall support the ability to declare and discover the file format used by the additional images contained in the file.**	**No**

	Systems-level information about how and where an additional image is stored in the metadata of a JPEG360 file is required to properly extract and parse the additional image. For example, an additional image may be stored using JPEG-1 or JPEG XT file formatting, or in another form.	
Coding format	The standard shall support the ability to declare and discover the image coding format used to compress the images contained in the file. The additional images being stored in the JPEG 360 file are likely to have been compressed in order to reduce the overall file size. In order to use the compressed images, we need to know which compression has been applied. These may already be included in the definitions of storage format.	No
Pixel format	The standard shall support the ability to declare and discover the pixel format of the images contained in the file. Further details of the compression, such as pixel component interleave ordering, pixel color space, and color gamut, are needed to reproduce the uncompressed images. These may already be included in the definitions of the compression codec and storage format.	No
Viewport position upon opening image	The standard should support the ability to declare and discover the initial position in the projection rendering which is viewed when the image is opened. The ability to set the viewport position allows for the primary subject of the image to be seen first when the image is rendered.	No
Associate a region of pixels to an interactive action	The standard should support the ability to declare and discover the association of a region of pixels with an action, such as a mouse or controller, to create interactivity with image elements. This feature allows for interactions by the user with elements seen in the image. For example, putting cursor over a person's face could bring up an overlay of that person's name, or clicking on a door seen in the image could cause the image to be changed to an image taken in the adjacent room of a house.	Yes

Camera pose relative to Earth reference	**The standard should support the ability to declare and discover the pose of the camera relative to a local reference frame.** **Cameras are increasingly provided with a variety of sensors, such as a gravity sensor and Eath's magnetic field, from which the camera's overall orientation can be determined. This information can be helpful with image stabilization, or setting the nominal image horizon.**	**No**
Location data	**The standard should support the ability to declare and discover a location that is associated to an image, or a set of images, or a set of locations to a set of images.** **The notion of location occurs at many abstraction levels, ranging from a set of longitude, latitude, altitude coordinates to building address, neighborhood, city, region, country, continent, planet, solar system, galaxy, etc.** **Location is often combined with other sensor data, such as pose, to further refine the oberved scene.**	**Yes**
Date & Time	**The standard should support the ability to declare and discover a timestamp that is associated to an image, or a set of images, or a set of timestamps associate to a set of images.**	**Yes**
Camera sensor data	**The standard should support the the ability to declare and discover associated sensor data.** **Many cameras have built-in or access to nearby sensors to monitor acceleration, velocity, temperature, underwater depth, humidity, heart rate, respiration rate, blood pressure, GPS, etc.**	**Yes**
Relative sensor arrangement for multi-sensor capture	**The standard should support the ability to declare and discover relative sensors positions a suitable coordinate system, including a 2D grid array.** **Assuming the availability of depth information, this information should allow to transform the image of one camera/sensor to the viewing position of another camera/sensor. This allows for stitching multiple images**	**No**

	into a larger image, such as a spherical image, or partial spherical image, cube-map image, etc. under possible consideration of occuring parallax. **Examples of multisensor cameras include Google VR180), and spherical (Ricoh Theta series). It is also possible to use a time-series of captures using a typical 2D camera which can be stitched together to create panoramas to various projections.**	
Exposure time	**The standard should support the ability to declare and discover the exposure time that is associated to each image captured within the same file.**	**No**
Embed video clips	**The standard shall support the ability the ability to declare and discover multiple videos within the same file.** **360 cameras typically can capture high resolution still and lower resolution video images. The viewer's experience and understanding of the image can be enhanced if it's possible to see a relevant video clip that provides more context for the image.**	**No**
Embed audio-only clips	**The standard should support the ability to declare and discover an audio clip associated to an image or a set of images within the same file.** **A brief clip of sounds captured concurrent with the image capture can be used to enhance the experience of the image; e.g., the trickling of water, birdsong, and spoken narrative to explain the image content.**	**Yes**
Director's view for scripted experience	**The standard should support the ability to declare and discover scripted experience of the 360 image within the same file.** **The viewport of a spherical image is usually smaller than the total viewable area, and several items of interest may fall outside the initial viewport setting. A script that guides the viewport through the interesting portions of the image (a director's cut) can increase the viewer's enjoyment and understanding of the image.**	**Yes**
Lens parameters	**The standard should support the ability to declare and discover the optical system parameters that describe the image formation onto the sensor within the same file.**	**No**

	In order that the images can be stitched and properly mapped to the desired projection requires knowledge of the distortions, focal length, modulation transfer function, and other optics information.	
Embed stereo 360 images	**The standard should support the ability to declare and discover the association to stereo images within the same file.**	**No**
Embed stereo 360 video clips	**The standard should support the ability to declare and discover the association to stereo video clips within the same file.**	**Yes**
Support JPEG Systems Part 4 for Privacy and Security	**The standard should support the ability to declare and discover the metadata needed to support JPEG Systems Part 4 – Privacy & Security.**	**No**
Default render mode wrt available images and display devices.	**The standard should support the ability to declare and discover the preferred characteristics of the display device.** **If the image was created to be experienced on a head mounted device (HMD), and both a 2D monitor and HMD are connected as display devices, the image should be preferentially shown on the HMD.**	**Maybe**
Stitching software & version	**The standard may support the ability to declare and discover the relevant information about the origin and version of software which was used to generate the image projections, people tags, object tags, etc.** **Stitching algorithms are in rapid transition. If the component images are available, and an improved stitching algorithm is available, the stitched image can be updated with a better quality version.**	**Perhaps**
Link to external data sources for the additional images	**The standard may support the ability to declare and discover links to images which are stored in a separate file (local or remote).** **For sharing images through email, or for uploading to photosharing sites, it may be desirable to reduce the overall size of the file by removing some of the embedded files, and replacing them with links (e.g., URIs) so to represent the deleted images. Likewise, to protect anonymity, it might be**	**Yes, rich links.**

	helpful to remove face taggings, but to keep a reference as to where those tags are available.	
Embed HTML5 scripts	**The standard may support the ability to declare and discover the HTML5 scripts.** **When using a browser to view images, customing the experience through HTML5 code improve the experience, for example, to create a scrapbook feel.**	**Yes.**
Support encrypted data	**The standard may support the ability to declare and discover the presence of encrypted data.** **Hardware and software vendors may want to protect metadata they have added to the images, and to make it accessible under certain restrictions. End users may want to protect sensitive metadata from unintended uses.**	**No**
Embed spatial audio clips	**The standard may support the ability to declare and discover the presence of spatial audio.** **Spatial audio, aka 3d audio, clips give a more immersive experience and can be used as another way to guide users through a 360 image.**	**Yes**
Support for medical images	**The standard may support the ability to declare and discover the presence of an atypical image in the same file.**	**Yes**
Support authenticated data	**The standard may support the ability to declare and discover the authenticity of the images and/or metadata in the file.**	**Perhaps**
Dynamic Viewport -- programmatic	**The standard may support the ability to declare and discover the presence of a dynamically generated script for the viewport behavior.** **Similar to the director's cut, except this script is external to the image file. At the time the image is opened, the metadata provides an API through which the viewport behavior is scripted.**	**Yes**
In-image data tags	**The standard may support the ability to declare and discover the metadata which is derived from the images within the same file.** **The meeting attendance list can be constructed by finding and identifying the faces seen in a 360 image taken from the**	**Yes**

	middle of a conference room table; this list can be included within the file for future reference.	
Accelerated ROI rendering	**The standard may support the ability to declare an order of ROIs for quickly rendering ROIs of the images.**	**No**

C.1.2 Metadata Constructional Requirements

All these requirements remain the same for Part 6 or Part 7.

Metadata Requirements and Description	
Platform independence	**The standard shall be usable on a wide variety of hardware and operating systems.**
Compatible to JPEG-1 compliant decoder	**The standard shall be compatible with JPEG-1 compliant decoder in that method used to add the metadata does not degrade the expected operation of a JPEG-1 compliant decoder.** **When using a JPEG-1 compliant decoder to read a file with the new metadata, the JPEG-1 compliant decoder is able to render the expected image, and it ignores the new metadata.**
Compatible to JPEG XT compliant decoder	**The standard shall be compatible with JPEG XT compliant decoder in that method used to add the metadata does not degrade the expected operation of a JPEG XT compliant decoder.** **When using a JPEG XT compliant decoder to read a file with the new metadata, the JPEG XT compliant decoder is able to render the expected image, and it ignores the new metadata.**
Compatible to be used within JPEG Box File Formats	**The standard shall be compatible and fully functional with the JPEG Box File structure.** **The proposal may also work with files structured differently from the JPEG Box File Format, but this is not mandatory.**
Compliant to JPEG Systems guidelines ISO/IEC 19566-1	**The standard should be compliant to the guidelines in ISO/IEC 19566-1.**
Compatbility with existing JPEG-1 metadata functionality	**The standard should provide for compatibility with the functionality provided by existing metadata.**
Use existing ISO metadata formats.	**The standard should use existing standards with respect to the presentation of metadata information such as XMP.**
Compatibility with long term metadata standard roadmap	**The standard should be support the addition of metadata required to support new usages.**
Metadata contained within a BOX, and be capable to reference (meta)data in different BOXes.	**The standard should support metadata being contained within a JPEG Systems compliant BOX format, and should have the ability to refer to (meta)data contained in other BOXes.**

	Metadata will be contained within a well-known BOX. This metadata can refer to data and metadata contained in other BOXes within the same file, or BOXes in other files. It is also possible that when referring to an image, that referenced image may also contain metadata; i.e., recursive metadata.
Robust to a variety representations of the metadata.	**The standard should provide robust with respect to formats and data type differences to represent the same information.**
Usable with JPEG Box File Formats AND non-JPEG file formats	**The standard may work also be compatible to non-BOX format files.**
Non-Box Image File Directory (IFD) container	**The standard may work also be compatible to these particular non-BOX format files without regard to other non-BOX format files.**