INTERNATIONAL STANDARD

ISO/IEC 29199-2

Fourth edition
2020-05

Information technology — JPEG XR image coding system —

Part 2:
Image coding specification

Technologies de l'information — Système de codage d'image JPEG XR —

Partie 2: Spécification de codage d'image

Reference number
ISO/IEC 29199-2:2020(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).

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

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

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

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*, in collaboration with ITU-T. The technically identical text is published as Rec. ITU-T T.832.

This fourth edition cancels and replaces the third edition (ISO/IEC 29199-2:2012), which has been technically revised. It also incorporates the Amendment ISO/IEC 29199-2:2012/Amd.1:2017.

The main changes compared to the previous edition include:

- the specification of additional colour type identifiers;
- the specification of an alternative file storage format based on ISO/IEC 23008-12 for the storage and interchange of JPEG XR coded images and image sequences;
- the specification of media type identifiers for use for use in various internet protocols.

A list of all parts in the ISO/IEC 29199 series can be found on the ISO website.

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.

Introduction

This document specifies requirements and implementation guidelines for the compressed representation of digital images for storage and interchange in a form referred to as JPEG XR. The JPEG XR design provides a practical coding technology for a broad range of applications with excellent compression capability and important additional functionalities. An input image is typically operated on by an encoder to create a JPEG XR coded image. The decoder then operates on the coded image to produce an output image that is either an exact or approximate reconstruction of the input image.

The primary intended application of JPEG XR is the representation of continuous-tone still images such as photographic images. The manner of representation of the compressed image data and the associated decoding process are specified. These processes and representations are generic, that is, they are applicable to a broad range of applications using compressed colour and grayscale images in communications and computer systems and within embedded applications, including mobile devices.

As of 2008, the most widely used digital photography format is a nominal implementation of the first JPEG coding format as specified in ITU-T Recommendation T.81 | ISO/IEC 10918-1. This encoding uses a bit depth of 8 for each of three channels, resulting in 256 representable values per channel (a total of 16 777 216 representable colour values).

More demanding applications may require a bit depth of 16, providing 65 536 representable values for each channel, and resulting in over $2.8 * 10^{14}$ colour values. Additional scenarios may necessitate even greater bit depths and sample representation formats. When memory or processing power is at a premium, as few as five or six bits per channel may be used.

The JPEG XR specification enables greater effective use of compressed imagery with this broadened diversity of application requirements. JPEG XR supports a wide range of colour encoding formats including monochrome, RGB, CMYK and n-component encodings using a variety of unsigned integer, fixed point, and floating point decoded numerical representations with a variety of bit depths. The primary goal is to provide a compressed format specification appropriate for a wide range of applications while keeping the implementation requirements for encoders and decoders simple. A special focus of the design is support for emerging high dynamic range (HDR) imagery applications.

JPEG XR combines the benefits of optimized image quality and compression efficiency together with low-complexity encoding and decoding implementation requirements. It also provides an extensive set of additional functionalities, including:

- high compression capability;
- low computational and memory resource requirements;
- lossless and lossy compression;
- image tile segmentation for random access and large image formats;
- support for low-complexity compressed-domain image manipulations;
- support for embedded thumbnail images and progressive resolution refinement;
- embedded codestream scalability for both image resolution and fidelity;
- alpha plane support;
- bit-exact decoder results for fixed and floating point image formats.

Important detailed design properties include:

- high performance, embedded system friendly compression;
- small memory footprint;
- integer-only operations with no divides;
- a signal processing structure that is highly amenable to parallel processing;
- use of the same signal processing operations for both lossless and lossy compression operation;
- support for a wide range of decoded sample formats (many of which support high dynamic range imagery):
  - monochrome, RGB, CMYK or n-component image representation;
  - 8- or 16-bit unsigned integer;
  - 16- or 32-bit fixed point;
  - 16- or 32-bit floating point;
  - several packed bit formats;
  - 1-bit per sample monochrome;
  - 5- or 10-bit per sample RGB;
  - radiance RGBE.

The algorithm uses a reversible hierarchical lifting-based lapped biorthogonal transform. The transform has lossless image representation capability and requires only a small number of integer processing operations for both encoding and decoding. The processing is based on 16 × 16 macroblocks in the transform domain, which may or may not affect overlapping areas in the spatial domain (with the overlapping property selected under the control of the encoder). The design provides encoding and decoding with a minimal memory footprint suitable for embedded implementations.

The algorithm provides native support for both RGB and CMYK colour types by converting these colour formats to an internal luma-dominant format through the use of a reversible colour transform. In addition, YUV, monochrome and arbitrary n-channel colour formats are supported.

The transforms employed are reversible; both lossless and lossy operations are supported using the same algorithm. Using the same algorithm for both types of operation simplifies implementation, which is especially important for embedded applications.

A wide range of numerical encodings at multiple bit depths are supported: 8-bit and 16-bit formats, as well as additional specialized packed bit formats, are supported for both lossy and lossless compression. (32-bit formats are supported using lossy compression.) Up to 24 bits are retained through the various transforms. While only integer arithmetic is used for internal processing, lossless and lossy coding are supported for floating point and fixed point image data – as well as for integer image formats.

The main body of this document specifies the syntax and semantics of JPEG XR coded images and the associated decoding process that produces an output image from a coded image. Annex A specifies a tag-based file storage format for storage and interchange of such coded images. Annex B specifies profiles and levels, which determine conformance requirements for classes of encoders and decoders. Aspects of colour imagery representations and colour management are discussed in Annex C. The typical expected encoding process is described in Annex D. Annex E contains a media type specification for images encoded according to the tag-based format specified in Annex A for use in various internet protocols. Annex F specifies an alternative file storage format based on

ISO/IEC 23008-12 and associated media type specifications for the storage and interchange of JPEG XR coded images and image sequences. Annexes A, B, E, and F are an integral part of this document and contain normative specifications.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of patents.

ISO and IEC take no position concerning the evidence, validity and scope of these patent rights.

The holders of these patent rights have assured ISO and IEC that they are willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patent rights are registered with ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Information technology — JPEG XR image coding system —

Part 2:
Image coding specification

1 Scope

This document specifies a coding format, referred to as JPEG XR, which is designed primarily for continuous-tone photographic content.

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.

Rec. ITU-T T.833 | ISO/IEC 29199-3, *Information technology — JPEG XR image coding system — Part 3: Motion JPEG XR*

ISO/IEC/IEEE 60559, *Information technology — Microprocessor systems — Floating-Point arithmetic*

ISO/IEC 10646:2017, *Information technology — Universal coded character set (UCS)*

ISO/IEC 23008-12:2017, *Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 12: Image file format*

3 Terms and definitions

For the purposes of this document, the terms, definitions and abbreviated terms specified in ISO/IEC 23008-12 and the following apply.

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

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

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

NOTE For the avoidance of doubt, in case of ambiguities, the definitions in this document take precedence over the definitions of ISO/IEC 23008-12 except in regard to the file format specified in Annex F. A sample in the context of ISO/IEC 23008-12 and in Annex F is "all the data associated with a single time". In Annex G, this is meant as all data associated with one coded image, not "element in a two-dimensional image array that comprises an image plane".

3.1
adaptive coefficient normalization
parsing sub-process where *transform coefficients* (3.75) are dynamically partitioned into a *VLC*-coded (3.77) part and a *fixed-length coded* (3.28) part, in a manner designed to control (i.e., "normalize") bits used to represent the VLC-coded part

Note 1 to entry: The fixed-length coded part of DC coefficients and low-pass coefficients is called FLC refinement and the fixed-length coded part of high-pass coefficients is called flexbits.

3.2
adaptive inverse scanning
parsing sub-process where the *zigzag scan order* (3.80) associated with a set of *transform coefficients* (3.75) is dynamically modified, based on the statistics of previously-parsed transform coefficients

3.3
adaptive VLC
parsing sub-process where the code table associated with *VLC* (3.77) parsing of a particular syntax element is switched, among a finite set of fixed tables, based on the statistics of previously-parsed instances of this syntax element

3.4
alpha image plane
optional secondary *image plane* (3.36) associated with an image of the same dimensions as the *luma* (3.45) component of the *primary image plane* (3.57)

Note 1 to entry: The alpha image plane has one component, a luma component.

3.5
block
m×n array of *samples* (3.64), or an m×n array of *transform coefficients* (3.75)

3.6
block index
integer in the range 0 to 15 identifying, by its position in *raster scan order* (3.61), a particular 4×4 *block* (3.5) within a partition of a 16×16 block into 16 4×4 blocks

3.7
byte
sequence of 8 bits

3.8
byte-aligned
bit in a *codestream* (3.13) where its position is an integer multiple of 8 bits from the beginning of the codestream, where the first bit in the codestream is at position 0

3.9
chroma
component (3.14) of the *primary image plane* (3.57) with non-zero index, or the *transform coefficients* (3.75) and sample values associated with this component

3.10
coded block pattern high-pass
syntax element indicating the *coded block status* (3.12), i.e. the presence or absence of non-zero *high-pass coefficients* (3.34), for each of the *blocks* (3.5) in the *macroblock* (3.46)

3.11
coded block pattern low-pass
syntax element indicating the presence or absence of non-zero *low-pass coefficients* (3.44) in the *macroblock* (3.46)

3.12
coded block status
indication of the presence or absence of non-zero *transform coefficients* (3.75) in that *block* (3.5)

3.13
codestream
sequence of bits contained in a sequence of *bytes* (3.7) from which syntax elements are parsed

Note 1 to entry: The most significant bit of the first byte is the first bit of the codestream, the next most significant bit of the first byte is the second bit of the codestream, and so on, to the least significant bit of the first byte (which is the eighth bit of the codestream), followed by the most significant bit of the second byte (which is the ninth bit of the codestream), and so on, up to and including the least significant bit of the last byte of the sequence of bytes (which is the last bit of the codestream).

3.14
component
array of samples associated with an *image plane* (3.36)

3.15
context
possible value of a specific instance of a *context variable* (3.16)

3.16
context variable
variable used in the *parsing process* (3.54) to select which data structure is to be used for the *adaptive VLC* (3.3) parsing of a given syntax element

3.17
DC coefficient
first subset when the *transform coefficients* (3.75), that are contained in a specific *macroblock* (3.46) and a specific *component* (3.14), are partitioned into 3 subsets

3.18
DC-LP array
array of all *DC* (3.17) and *low-pass* (3.44) *transform coefficients* (3.75), for all *macroblocks* (3.46) associated with a specific *component* (3.14)

3.19
decoder
embodiment of a *parsing process* (3.54) and *decoding process* (3.20)

3.20
decoding process
process of computing output sample values from the parsed syntax elements of the *codestream* (3.13)

3.21
dequantization
process of rescaling the quantized *transform coefficients* (3.75) after their value has been parsed from the *codestream* (3.13) and before they are presented to the *inverse transform process* (3.41)

3.22
discriminant
one of DiscrimVal1 or DiscrimVal2, which are the two member variables of an instance of the *adaptive VLC* (3.3) data structure

Note 1 to entry: The adaptive VLC data structure is specified in subclause 5.5.5.

3.23
encoder
embodiment of an *encoding process* (3.24)

3.24
encoding process
process of converting source sample values into a *codestream* (3.13)

3.25
extended image
image (3.35) produced by the *decoding process* (3.20) prior to *windowing* (3.79)

Note 1 to entry: The extended image has a *luma* (3.45) array that is an integer multiple of 16 in width and height.

3.26
file
finite-length sequence of *bytes* (3.7) that is accessible to a *decoder* (3.19) in a manner such that the decoder can obtain access to the data at specified positions within the sequence of bytes

EXAMPLE Access to the data can be achieved by storing the entire sequence of bytes in random access memory or by performing "position seek" operations to specified positions within the sequence of bytes.

3.27
file format
specified structure for the content of a *file* (3.26)

3.28
fixed-length code
FLC
code which assigns a finite set of allowable bit patterns to a specific set of values, where each bit pattern has the same length

3.29
FLC refinement
fixed-length coded (3.28) part of a *DC coefficient* (3.17) or *low-pass coefficient* (3.44) that is parsed using adaptive fixed-length codes

3.30
flexbits
fixed-length coded (3.28) part of the *high-pass coefficient* (3.34) information which is parsed using adaptive fixed-length codes

3.31
frequency band
one of three subsets of the *transform coefficients* (3.75) for an *image* (3.35), which are separately parsed: *DC coefficients* (3.17), *low-pass coefficients* (3.44) and *high-pass coefficients* (3.34)

3.32
frequency mode
codestream (3.13) structure mode where the *DC* (3.17), *low-pass* (3.44), *high-pass* (3.34) and *flexbits* (3.30) *frequency bands* (3.31) for each *tile* (3.72) are grouped separately

3.33
hard tiles
codestream (3.13) structure mode where the overlap operators are not applied across tile boundaries; instead, boundary overlap operators are applied at tile boundaries

3.34
high-pass coefficient
third subset, when the *transform coefficients* (3.75) that are contained in a specific *macroblock* (3.46) and a specific *component* (3.14) are partitioned into 3 subsets

3.35
image
result of the *decoding process* (3.20) consisting of a *primary image plane* (3.57) and an optional *alpha image plane* (3.4)

3.36
image plane
collective term for a grouping of the *components* (3.14) of the *image* (3.35)

3.37
initial level value
one of two values used to compute the *VLC*-coded (3.77) part of a *transform coefficient* (3.75)

3.38
interleaved alpha image plane
alpha image plane (3.4) that is coded in a *codestream* (3.13) in an interleaved manner together with the *primary image plane* (3.57)

3.39
internal colour format
colour format associated with the spatial-domain samples obtained through the *inverse transform process* (3.40) and the *sample reconstruction process* (3.65), and distinguished from the *output colour format* (3.49) associated with the *output formatting process* (3.50)

3.40
inverse core transform
ICT
two steps of the *inverse transform process* (3.40) that involve processing of *transform coefficients* (3.75) associated with each *macroblock* (3.46) independently, with no *overlap filtering* (3.53)

3.41
inverse transform process
part of the *decoding process* (3.20) by which a set of *dequantized* (3.21) *transform coefficients* (3.75) are converted into spatial-domain values

3.42
inverse scanning
process of reordering an ordered set of parsed syntax elements from the *codestream* (3.13) to form an array of *transform coefficients* (3.75) associated with a specific *component* (3.14) and *macroblock* (3.46)

3.43
little-endian form
ordering of the *bytes* (3.7) that represent a numerical value as an integer number of bytes in which the bytes representing the number are in ascending order of significance

Note 1 to entry: With the least significant byte first, followed by the next least significant byte, etc.

3.44
low-pass coefficient
second subset, when the *transform coefficients* (3.75) that are contained in a specific *macroblock* (3.46) and a specific *component* (3.14) are partitioned into 3 subsets

3.45
luma
component (3.14) of an *image plane* (3.36) with index zero, and the *transform coefficients* (3.75) and sample values associated with this component

Note 1 to entry: Although this term is commonly associated with a signal that conveys perceptual brightness information, as used in this document the term is primarily an identifier of a particular array of samples or transform coefficients for an *image* (3.35).

3.46
macroblock
collection of *transform coefficients* (3.75) or samples, across all *components* (3.14), that have the same indices *i* and *j* with respect to a *macroblock partition* (3.47)

3.47
macroblock partition
partitioning of each *component* (3.14), into 16×16, 8×8, or 16×8 *blocks* (3.5), depending on the *internal colour format* (3.39)

3.48
output bit depth
representation, including the number of bits and the interpretation of the bit pattern, used for the sample values of the output *image* (3.35) that are the result of the *decoding process* (3.20)

3.49
output colour format
colour format associated with the output *image* (3.35) that is the result of the *decoding process* (3.20)

3.50
output formatting process
process of converting the arrays of samples (that are the result of the *sample reconstruction process* (3.65)) into the output samples that constitute the output of the *decoding process* (3.20)

Note 1 to entry: This specifies a conversion (if necessary) into the appropriate *output colour format* (3.49) and *output bit depth* (3.48).

3.51
output image height
height of the sub-array of the *luma* (3.45) *component* (3.14) of the *primary image plane* (3.57) that is output by the *decoding process* (3.20)

3.52
output image width
width of the sub-array, of the *luma* (3.45) *component* (3.14) of the *primary image plane* (3.57) that is output by the *decoding process* (3.20)

3.53
overlap filtering
steps of the *inverse transform process* (3.41) that involve processing of *transform coefficients* (3.75) across adjacent *blocks* (3.5) and *macroblocks* (3.46)

Note 1 to entry: When overlap filtering is applied, it is applied across macroblock boundaries as well as block boundaries. When the *codestream* (3.13) uses *soft tiles* (3.67), the overlap filtering is also applied across *tile* (3.72) boundaries. Otherwise, overlap filtering does not occur across tile boundaries.

3.54
parsing process
process of extracting bit sequences from the *codestream* (3.13), converting these bit sequences to syntax element values, and setting the values of global variables for use in the *decoding process* (3.20)

3.55
prediction
process of computing an estimate of the sample value or data element that is currently being decoded

3.56
prediction residual
difference between the result of the *prediction* (3.55) process invoked for a sample or data element, and its intended value

