INTERNATIONAL STANDARD

ISO/IEC 29199-2

First edition
2009-08-15

JPEG XR image coding system —

Part 2:
Image coding specification

Système de codage d'image JPEG XR —

Partie 2: Spécification de codage d'image

Reference number
ISO/IEC 29199-2:2009(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. 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.

ISO/IEC 29199-2 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.

This part of ISO/IEC 29199 is technically aligned with ITU-T Rec. T.832 but is not published as identical text.

ISO/IEC 29199 consists of the following parts, under the general title *JPEG XR image coding system*:

— *Part 2: Image coding specification*

— *Part 3: Motion JPEG XR*

— *Part 4: Conformance testing*

— *Part 5: Reference software*

System architecture will form the subject of a future Part 1.

Introduction

This International Standard 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 color 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 (at total of 16 777 216 representable color values).

More demanding applications may require a bit depth of 16, providing 65 536 representable values for each channel, and resulting in over $2.8 * 10^{14}$ color 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 color 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 color types by converting these color formats to an internal luma-dominant format through the use of a reversible color transform. In addition, YUV, monochrome and arbitrary n-channel color 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 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 part of ISO/IEC 29199 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 color imagery representations and color management are discussed in Annex C. The typical expected encoding process is described in Annex D.

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 companies listed in Annex E.

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

JPEG XR image coding system —

Part 2: Image coding specification

1 Scope

This Part of ISO/IEC 29199 specifies a coding format, referred to as JPEG XR, which is designed primarily for continuous-tone photographic content.

2 Normative references

Normative references having a scope that is limited to the use of the file format specified in Annex A are listed in A.2.

3 Terms and definitions

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

NOTE Definitions of terms having a scope that is limited to the use of the **file format** specified in Annex A are listed in A.2.

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

NOTE 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** associated with a set of **transform coefficients** 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** 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 associated with an image, of the same dimensions as the luma component of the primary image plane

NOTE The alpha image plane has one component, a luma component.

3.5
block
m×n array of **samples**, or an m×n array of **transform coefficients**

3.6
block index
integer in the range 0 to 15 identifying, by its position in **raster scan order**, a particular 4×4 **block** 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** is **byte-aligned** if 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 of the **primary image plane** with non-zero index, or the **transform coefficients** and sample values associated with this **component**

3.10
coded block pattern high-pass
coded block pattern high-pass is a **syntax element** indicating the **coded block status**, i.e. the presence or absence of non-zero **high-pass transform coefficients**, for each of the **blocks** in the **macroblock**

3.11
coded block pattern low-pass
coded block pattern low-pass is a **syntax element** indicating the presence or absence of non-zero **low-pass transform coefficients** in the **macroblock**

3.12
coded block status
coded block status is an indication of the presence or absence of non-zero **transform coefficients** in that **block**

3.13
codestream
sequence of bits contained in a sequence of **bytes** from which syntax elements are parsed, such that 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**

3.15
context
possible value of a specific instance of a **context variable**

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

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

3.18
DC-LP array
array of all DC and low-pass **transform coefficients**, for all **macroblocks** associated with a specific **component**

3.19
decoder
embodiment of a **parsing process** and **decoding process**

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

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

3.22
discriminant
collective term for one of DiscrimVal1 or DiscrimVal2, which are the two member variables of an instance of the **adaptive VLC** data structure specified in subclause 5.5.5.

3.23
encoder
embodiment of an **encoding process**

3.24
encoding process
process of converting source sample values into a **codestream** conforming to this specification

3.25
extended image
image produced by the **decoding process** prior to **windowing**

NOTE The **extended image** has a **luma** array that is an integer multiple of 16 in width and height.

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

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

3.29
flexbits
fixed length coded part of the **high-pass coefficient** information which is parsed using adaptive fixed-length codes

3.30
frequency band
collective term for one of the following three subsets of the **transform coefficients** for an **image**, which are separately parsed: **DC coefficients**, **low-pass coefficients**, and **high-pass coefficients**

3.31
frequency mode
codestream structure mode where the DC, low-pass, high-pass and **flexbits frequency bands** for each **tile** are grouped separately

3.32
hard tiles
codestream structure mode where the overlap operators are not applied across tile boundaries; instead, boundary overlap operators are applied at tile boundaries

3.33
high-pass coefficient
third subset, when the **transform coefficients** that are contained in a specific **macroblock** and a specific **component** are partitioned into 3 subsets

3.34
image
result of the **decoding process**, consisting of a **primary image plane** and an optional **alpha image plane**

3.35
image plane
collective term for a grouping of the **components** of the **image**

3.36
initial level value
one of two values used to compute the **VLC**-coded part of a **transform coefficient**

3.37
internal color format
color format associated with the spatial-domain samples obtained through the **inverse transform process** and the **sample reconstruction process**, and distinguished from the **output color format** associated with the **output formatting process**

3.38
inverse core transform (ICT)
two steps of the **inverse transform process** that involve processing of **transform coefficients** associated with each **macroblock** independently, with no **overlap filtering**

3.39
inverse transform process
part of the **decoding process** by which a set of **dequantized transform coefficients** are converted into spatial-domain values

3.40
inverse scanning
process of reordering an ordered set of parsed **syntax elements** from the **codestream** to form an array of **transform coefficients** associated with a specific **component** and **macroblock**

3.41
little-endian form
ordering of the **bytes** that represent a numerical value as an integer number of **bytes** in which the **bytes** representing the number are in ascending order of significance, i.e. with the least significant **byte** first, followed by the next least significant **byte**, etc.

3.42
low-pass coefficient
second subset, when the **transform coefficients** that are contained in a specific **macroblock** and a specific **component** are partitioned into 3 subsets

3.43
luma
component of an **image plane** with index zero, and the **transform coefficients** and sample values associated with this **component**

NOTE Although this term is commonly associated with a signal that conveys perceptual brightness information, as used in this International Standard the term is primarily an identifier of a particular array of samples or **transform coefficients** for an **image**.

3.44
macroblock
collection of **transform coefficients** or samples, across all **components**, that have the same indices *i* and *j* with respect to a **macroblock partition**

3.45
macroblock partition
partitioning of each **component**, into 16×16, 8×8, or 16×8 **blocks**, depending on the **internal color format**

3.46
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** that are the result of the **decoding process**

3.47
output color format
color format associated with the output **image** that is the result of the **decoding process**

3.48
output formatting process
process of converting the arrays of samples (that are the result of the **sample reconstruction process**) into the output samples that constitute the output of the **decoding process**

NOTE This specifies a conversion (if necessary) into the appropriate **output color format** and **output bit depth**.

3.49
output image height
height of the sub-array of the **luma component** of the **primary image plane** that is output by the **decoding process**

3.50
output image width
width of the sub-array, of the **luma component** of the **primary image plane** that is output by the **decoding process**

3.51
overlap filtering
steps of the **inverse transform process** that involve processing of **transform coefficients** across adjacent **blocks** and **macroblocks**

NOTE When **overlap filtering** is applied, it is applied across **macroblock** boundaries as well as **block** boundaries. When the **codestream** uses **soft tiles**, the **overlap filtering** is also applied across **tile** boundaries. Otherwise, **overlap filtering** does not occur across **tile** boundaries.

3.52
parsing process
process of extracting bit sequences from the **codestream**, converting these bit sequences to syntax element values, and setting the values of global variables for use in the **decoding process**

3.53
prediction
process of computing an estimate of the sample value or data element that is currently being decoded

3.54
prediction residual
difference between the result of the **prediction** process invoked for a sample or data element, and its intended value

3.55
primary image plane
image plane that consists of all **image components** that are not a part of the **alpha image plane**

3.56
QP index
integer, which for a particular **frequency band** and **macroblock** specifies the index into the table of **quantization parameters** available for this **frequency band** and **tile**

NOTE The **QP index** thereby selects, for this **macroblock**, the **quantization parameter** used for the **dequantization** of the **transform coefficients** in the specific **frequency** band.

3.57
QP set
set of **quantization parameters** associated with a particular **frequency band**, corresponding to the **luma** and **chroma components**

3.58
quantization parameter
value used to compute the scaling factor for the **dequantization** of a **transform coefficient**, before the **inverse transform process** is applied

3.59
raster scan order
scan order in which a two-dimensional array of values is scanned row-wise from left to right, and the rows are scanned from the top row to the bottom

3.60
refinement
process of modifying a predicted or partially-computed **transform coefficient**

3.61
run
number of zero valued coefficient levels that precede a non-zero valued coefficient level in the **zigzag scan order** during the **inverse scanning** process

3.62
sample reconstruction process
process of converting dequantized **transform coefficients** into samples of the **image**

3.63
soft tiles
codestream structure mode where the overlap operators are applied across tile boundaries

3.64
spatial co-location
sub-arrays of samples are **spatially co-located** across **components** when they correspond to the same spatial region of the decoded **image**

NOTE The **macroblock partition** of the **image** ensures that the *i*-th **macroblock** horizontally and *j*-th **macroblock** vertically across all **components** are **spatially co-located**.

3.65
spatial mode
codestream structure mode where the **DC**, **low-pass**, **high-pass** and **flexbits frequency bands** for each specific **macroblock** are grouped together

3.66
spatial transformation
element in the **codestream** indicating the preferred final displayed orientation of the decoded **image**, as specified in subclause 8.3.8

NOTE The **spatial transformation** is only a suggestion, and **decoder** conformance is checked only for the decoded **image** prior to the application of this transformation (i.e. for orientation 0).

3.67
start code
bit pattern that specifies the beginning of a **tile packet** or other distinguished, contiguous set of syntax elements in the **codestream**

3.68
tile
collection of **macroblocks** that have the same indices *i* and *j* with respect to a **tile partition**

NOTE Each **tile** corresponds to the **macroblocks** for a rectangular region of the **image**.

3.69
tile packet
contiguous subset of the **codestream**, which contains the coded **syntax elements** associated with a specific **tile**

3.70
tile partition
partition of the **image** into rectangular arrays of **macroblocks**, as specified in subclause 6.4

3.71
transform coefficients
values, associated with each specific **macroblock** and specific **component**, that — after **dequantization** — form the input arrays into the **inverse transform process**

3.72
variable length code (VLC)
code which assigns a finite set of allowable bit patterns, where each bit pattern is potentially of a different length, to a specific set of values

3.73
VLC refinement
one of two values used to compute the **VLC**-coded part of a **transform coefficient**

NOTE The number of bits required to specify the **VLC-refinement** is dependent on the value of the **initial level value**. The **VLC refinement** is added to the **initial level value** to produce the **VLC**-coded part of the transform coefficient.

3.74
windowing
selection of **spatially co-located** sub-arrays of the **components** of all present **image planes** associated with an **image** that are output by the **decoding process**

3.75
zigzag scan order
adaptive ordering for the **inverse scanning** process, which assigns array indices to each subsequent **transform coefficient** parsed from the **codestream**