INTERNATIONAL STANDARD

ISO/IEC 21122-5

First edition
2020-10

Information technology — JPEG XS low-latency lightweight image coding system —

Part 5:
Reference software

Technologies de l'information — Système de codage d'images léger à faible latence JPEG XS —

Partie 5: Logiciel de référence

Reference number
ISO/IEC 21122-5:2020(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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

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

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

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO/IEC 21122 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 is part of a series of standards for a low-latency lightweight image coding system, denoted JPEG XS.

In many use cases during production or transmission of a movie, limiting the latency and the recompression loss is a more important aspect than the compression efficiency. The JPEG XS coding system offers compression and recompression of image sequences with very moderate computational resources while remaining robust under multiple compression and decompression cycles and mixing of content sources, e.g. embedding of subtitles, overlays or logos. Typical target compression ratios ensuring visually lossless quality are in the range of 2:1 to 10:1, depending on the nature of the source material. The end-to-end latency can be confined to a fraction of a frame, typically between a small number of lines down to below a single line.

This document provides the reference software of the ISO/IEC 21122 series. It has been successfully compiled and tested on Linux®[1)] and Windows™[1)] operating systems at the time of writing.

1) Linux® and Windows™ are examples of suitable products available commercially. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC of these products.

Information technology — JPEG XS low-latency lightweight image coding system —

Part 5: Reference software

1 Scope

This document contains the reference software of the ISO/IEC 21122 series. It acts as a guideline for implementation and as a reference for conformance testing.

2 Normative references

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

ISO/IEC 21122-1, *Information technology — JPEG XS low-latency lightweight image coding system — Part 1: Core coding system*

ISO/IEC 21122-2, *Information technology — JPEG XS low-latency lightweight image coding system — Part 2: Profiles and buffer models*

ISO/IEC 21122-4, *Information technology — JPEG XS low-latency lightweight image coding system — Part 4: Conformance testing*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 21122-1, ISO/IEC 21122-2, ISO/IEC 21122-4 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/

3.2 Abbreviated terms

MSE mean square error

PSNR peak signal to noise ratio

4 Conventions

4.1 Operators

NOTE Many of the operators used in this document are similar to those used in the C programming language.

+	addition
−	subtraction (as a binary operator) or negation (as a unary prefix operator)
×	multiplication
/	division without truncation or rounding

4.2 Typesetting

Regular face fonts as this text describe informative text that provides instructions, comments or details for the reader.

```
Monospaced text as this paragraph indicates program input or output as necessary to either
run the software, or as generated by the software on the console.
```

5 Reference software

5.1 Purpose

The purpose of this document is to provide the following.

— Provide a reference decoder software capable of decoding codestreams that conform to ISO/IEC 21122-1, or to ISO/IEC 21122-1 and ISO/IEC 21122-2.

— Provide a sample encoder software capable of producing codestreams that conform to ISO/IEC 21122-1 and ISO/IEC 21122-2.

The use of the reference software is not required for making an implementation of an encoder or decoder in conformance to any of the parts of the ISO/IEC 21122 series. Requirements established in ISO/IEC 21122-1 and ISO/IEC 21122-2 take precedence over the behaviour of the reference software.

The reference software decoder generates the output of the decoding process specified by ISO/IEC 21122-1. It therefore allows conformance testing by means of comparing the output of a candidate implementation by that of the reference software specified in this document. Conformance testing is specified in ISO/IEC 21122-4. The reference software decoder can also be used to validate the syntactic correctness of a codestream by attempting to decode a candidate codestream with the reference software.

5.2 Examples of use

Some examples of use for the reference decoder software implementations are as follows:

— As an illustration of how to perform the decoding processes specified in ISO/IEC 21122-1.

— As the starting basis for the implementation of a decoder that conforms to ISO/IEC 21122-1.

— For testing the conformance of a decoder implementation to ISO/IEC 21122-1 with the procedures specified in ISO/IEC 21122-4. Details on reference testing can be found in ISO/IEC 21122-4.

— For (non-exhaustive) testing of the conformance of a codestream (or file) to the constraints specified in ISO/IEC 21122-1.

The lack of detection of any conformance violation by any reference software implementation should not be considered as a definite proof that the codestream under testing conforms to all constraints required for it be conforming to ISO/IEC 21122-1. In fact, ISO/IEC 21122-4 defines additional testing procedures that can be used to test a codestream for conformance to ISO/IEC 21122-1 and ISO/IEC 21122-2, such as for example testing the conformance to the buffer models specified there.

Some examples of use for a sample encoder software are as follows:

— As an illustration of how to implement an encoding process that produces codestreams that are conforming to ISO/IEC 21122-1 and ISO/IEC 21122-2.

— As starting point for an implementation of an encoder that conforms to ISO/IEC 21122-1 and ISO/IEC 21122-2.

— As a means of generating codestreams conforming to ISO/IEC 21122-1 and ISO/IEC 21122-2 for testing purposes.

— As a means of demonstrating and evaluating examples of the quality that can be achieved by an encoding process that conforms to ISO/IEC 21122-1 and ISO/IEC 21122-2.

However, no guarantee of the quality that will be achieved by an encoder is provided by its conformance to ISO/IEC 21122-1 as the conformance is only defined in terms of specific constraints imposed on the syntax of the generated codestream. In particular, while sample encoder software implementations may suffice to provide some illustrative examples of which quality can be achieved within the ISO/IEC 21122 series, they provide neither an assurance of minimum guaranteed image encoding quality nor maximum achievable image encoding quality.

Similarly, the computation resource characteristics in terms of program or data memory usage, execution speed, etc. of sample software encoder or decoder implementations should not be construed as a representative of the typical, minimal or maximal computational resource characteristics to be exhibited by implementations of ISO/IEC 21122-1.

5.3 Layout of this document

Annex A describes unpacking, configuration and compiling the reference software. Annex B describes the file formats in which the sample encoder accepts images and into which the reference decoder reconstructs images. Annex C describes its use as encoder or decoder. In addition to the main text, the reference software is available at https://standards.iso.org/iso-iec/21122/-5/ed-1/en.

6 Copyright, licensing and intellectual property

These software modules were originally developed by the parties indicated in the file LICENSE within the package forming a part of this document, in the course of development of ISO/IEC 21122-5. ISO/IEC draws the attention of the users of these software modules to the licence terms and conditions specified in the same LICENSE file. Those intending to use these software modules in hardware or software products are advised that their use may infringe existing patents. In particular, the original developers of these software modules and their respective companies, the editors and their companies, and ISO/IEC have disclaimed liability for any proposed use of these software modules or modifications thereof.

Annex A
(informative)

Unpacking and compiling the reference software

A.1 Unpacking the software

The reference software is available at https://standards.iso.org/iso-iec/21122/-5/ed-1/en in the form of a ZIP archive named `reference.zip`. Unpacking a ZIP file is system specific. Under POSIX®[2)] compliant operating systems, open a command line window and enter

```
unzip reference.zip
```

This will unpack all components of the software into the current directory. Under a Windows™ operating system, double-clicking on the ZIP archive will open it. The complete contents of the ZIP archive should then be copied to a convenient target directory for compilation.

A.2 Prerequisites for building the software

A.2.1 Prerequisites for building the software under a POSIX® compliant system

Under a POSIX® compliant system, the following additional components need to be installed to be able to compile the reference software:

— The GNU gcc compiler collection, version 4.4.3 or later;

— The GNU make utility version 4.1 or later.

A.2.2 Prerequisites for building the software under Windows™

Under the Windows™ operating system, the following components are necessary:

— The Visual Studio™[2)] compiler version 15.0 or later.

A.3 Compiling the software

A.3.1 Compiling the software under a POSIX® compliant operating system

To compile the reference software under a POSIX® compliant operating system, open a command line and change the current directory to the directory into which the software was unpacked. Then enter the following command:

```
make -C build
```

This will create the reference software binary in either build/bin/linux64 or build/bin/linux32, depending on the native bit depth of the processor of the system on which the build command has been issued.

2) POSIX® and Visual Studio™ are examples of suitable products available commercially. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC of these products.

A.3.2 Compiling the software under Windows™

Open the Visual Studio™ solution file that is found in the directory build/vs_2015/jxs_reference.sln relative to the unpacking location of the archive.

Annex B
(informative)

File formats read and written by the reference software

B.1 General

This annex describes the file format of the images the reference software is able to encode, or into which the reference software will decode images. The reference software supports three file formats: ppm for 4:4:4 sampled content for bit depths between 8 and 16 bits, yuv for 4:2:2 content, and pgx for all other content. Clause B.2 describes the format of ppm files, Clause B.3 the format of yuv files. The format of pgx files is specified in ISO/IEC 21122-4.

B.2 ppm format

B.2.1 ppm capabilities

The ppm format is capable of carrying images of bit depths between 1 and 16 bits per component, with 3 components per sample. While ppm files are specified to describe images in the sRGB colour space, the reference software is agnostic to any colour space specification and will only encode and decode raw sample values.

A ppm file consists of a header that describes the dimensions and bit depth of the image, concatenated with the binary sample values contained in the image in raster scan order, left to right, top to bottom, with 1 or 2 bytes per sample.

B.2.2 Description of the ppm header

The ppm header consists of the following components:

— The two ASCII (see Reference [2]) characters P6 identifying this file as ppm file.

— An arbitrary number of white space characters, i.e. blank (ASCII 32 = 0x20), tab (ASCII 9 = 0x09) line feed (ASCII 10=0x0a) or carriage return (ASCII 12=0x0c).

— An image width in the number of sample positions, formatted as ASCII decimals.

— White space.

— An image height in the number of sample positions, formatted as ASCII decimals.

— White space.

— A maximum sample value formatted as ASCII decimals. This maximum sample value is a number of the form 2^P-1 where P is the bit precision of the samples.

— A **single** white space character.

B.2.3 Description of the ppm body

Following the ppm header, the W×H sample values follow, in raster scan order, left to right, top to bottom. For a sample precision P≤8, each sample value is encoded in binary, in 3 bytes, first the red, then the green, then the blue colour coordinates relative to colour primaries that are of no relevance for the reference software. This implies that most significant bits not required to represent sample values with P bits precision are zero.

For sample precisions P>8, also W×H binary encoded sample values follow, though each sample value is represented in 6 bytes. Most significant bits not required to represent sample values with P bits precision are zero. First the 8 most significant bits of the red colour coordinate, followed by the least 8 significant bits of the red colour coordinate, followed by the 8 most significant bits of the green colour coordinate, followed by the 8 least significant bits of the green colour coordinate, followed by the 8 most significant bits of the blue colour coordinate, then the 8 least significant bits of the blue colour coordinate. This format is also denoted as big endian.

Sample precisions P>16 cannot be represented by ISO/IEC 21122-1 and are thus irrelevant for the purposes of this document.

B.3 yuv format description

The yuv format is an unframed format, i.e. the data does not include any indication of the dimensions and bit depths of the image. Such side information needs to be given to the sample encoder as command line parameters and needs to be available for converting yuv into other formats for visual inspection of such images. yuv files consist of an even width of (luminance) samples.

A yuv file of an image of width W and height H of a sample precision P≤8 consists of the following:

— W×H bytes, encoding the sample values of the luminance channel Y, in raster scan order left to right, top to bottom, one byte per sample value.

— W/2×H bytes, encoding the sample values of the U chrominance channel, in raster scan order, left to right, top to bottom, one byte per sample value.

— W/2×H bytes, encoding the sample values of the V chrominance channel, in raster scan order, left to right, top to bottom, one byte per sample value.

— Most significant bits not required to represent sample values with P bits precision are zero.

A yuv file of an image of width W and height H of a sample precision P>8 consists of the following:

— W×H×2 bytes of luminance data, where each luminance sample is encoded in two bytes, with the 8 least significant bits first, followed by the 8 most significant bits, i.e. 2 bytes per sample.

— W×H bytes of chrominance data for the U channel, where each chrominance sample is encoded in two bytes, with the 8 least significant bits first, followed by the 8 most significant bits, i.e. 2 bytes per sample.

— W×H bytes of chrominance data for the U channel, where each chrominance sample is encoded in two bytes, with the 8 least significant bits first, followed by the 8 most significant bits, i.e. 2 bytes per sample.

— Most significant bits not required to represent sample values with P bits precision are zero.

Sample precisions P>16 cannot be represented by ISO/IEC 21122-1 and are thus irrelevant for the purpose of this document.

NOTE The difftest_ng tool that is supplied as an electronic attachment to ISO/IEC 21122-4 can be used to convert such images to other formats. To this end, difftest_ng uses a format indicator. The format indicator for yuv encoded files of precision P=8 of an image of dimension W×H reads

```
WxHx3:[8=0]:[8=1]/2x1:[8=2]/2x1
```

where the parameters W and H are substituted by the image width and height. The format indicator of an image of dimensions W×H of a bit precision P≥8, with Q=16-P reads as follows:

```
WxHx3:[Q-],[P-=0]:[Q-]/2x1,[P-=1]/2x1:[Q-],[P-=2]/2x1
```

where the parameters W, H, P and Q are substituted accordingly by the width, height, precision and 16−P. Images can be converted from pgx to yuv with the following command line:

```
difftest_ng --convert out.yuv@<spec> out.pgx -
```

where `<spec>` is one of the format indicators above. Conversion from yuv into pgx is accomplished by

```
difftest_ng --convert out.pgx out.yuv@<spec> -
```

Annex C
(informative)

Using the reference software

C.1 General

This annex provides guidelines how to use the sample software encoder and the reference software decoder. The sample software encoder can compress `ppm`, `pgx` or `yuv` files into a JPEG XS codestreams, and the reference decoder can reconstruct JPEG XS codestreams into `ppm`, `pgx` or `yuv` files, depending on the sampling format encoded in the codestream.

C.2 Running the sample software encoder

The sample software encoder is run on the command line as follows:

```
jxs_encoder <command line options> <inputfile> <outputfile>
```

where `<inputfile>` is a file name either ending on `.ppm`,`.pgx` or `.yuv`. If the file name ending is .ppm, a ppm encoded source image is assumed, if the file name ends on `.yuv`, a `yuv` encoded source file is assumed. If the file name ends on `.pgx`, a `pgx` encoded source file is assumed. The `ppm`, `pgx` and `yuv` formats are specified in Annex B.

The available command line options are as follows (parameter values are indicated in angle brackets):

`-p <profile>` Specifies the profile in which the source image will be encoded. The following profile indicators are available:

- 3 main profile;
- 4 high profile;
- 5 light-subline profile;
- 6 light profile.

`-b <bitrate>` Specifies the target bitrate in bits per pixel. Fractional numbers are possible.

`-s <bytesize>` Specifies the target file size in bytes. This command line option is mutually exclusive to the –b command line option. The encoder requires either the –b or the –s command line option, but not both.

`-o <mode>` Specifies the rate allocation mode. The following modes are available:

- `psnr` PNSR optimal encoding, i.e. minimize the mean square error;
- `visual` optimize for best visual performance.

`-w <width>` Specifies the width of the source image in luminance samples. This argument is only required for `yuv` input files.

`-h <height>` Specifies the height of the image in luminance samples. This argument is only required for `yuv` input files.

`-d <bitprecision>` Specifies the sample precision in bits for `yuv` input files. Not required for `pgx` or `ppm` input.

-f <frst_idx> With this option, the encoder is instructed to compress image sequences rather than a single image. The image source file names are generated by concatenating the sequence index to the input file name. This option specifies the index of the first image in the image sequence. The output file names are generated similarly by concatenating the sequence number to the output file name.

-n <seq_cnt> For image sequence compression (see the -f option above), this parameter specifies the number of images in the image sequence.

-C <config> Specifies the name of a configuration file to be used. The following config files are shipped with the reference software:

jpeg_xs_high.ini	Configures the encoder in the high profile with 5 horizontal and 2 vertical wavelet decomposition levels, 1 column and uniform quantization.
jpeg_xs_main.ini	Configures the encoder in the main profile with 5 horizontal and 1 vertical wavelet decomposition level, 1 column and uniform quantization.
jpeg_xs_light.ini	Configures the encoder in the light profile with 5 horizontal and 1 vertical wavelet decomposition level, 1 column and deadzone quantization.
jpeg_xs_light_subline.ini	Configures the encoder in the light sub-line profile with 5 horizontal and 0 vertical wavelet decomposition levels, 2 columns and uniform quantization.

-v Enables additional output for debugging.

C.3 Running the reference software decoder

The reference software encoder is run on the command line as follows:

```
jxs_decoder <command line options> <inputfile> <outputfile>
```

where <inputfile> is a JPEG XS encoded input file whose sample values are decoded into the output. The output file format is inferred from the appendix of the output file name. If the output file name ends on .ppm, a ppm encoded output image (see Annex B) is generated. If the output file name ends on .yuv, a yuv encoded output image is created. If the output file name ends on .pgx, a pgx encoded output file is generated.

The following command line options are supported by the decoder: