INTERNATIONAL STANDARD

ISO/IEC 10918-7

First edition
2019-12

Information technology — Digital compression and coding of continuous-tone still images —

Part 7: Reference software

Technologies de l'information — Compression numérique et codage des images fixes à modelé continu —

Partie 7: Logiciel de référence

Reference number
ISO/IEC 10918-7:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

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 ITU-T as ITU-T T.873 (05/2019) and drafted in accordance with its editorial rules. It was assigned to 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 10918 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.

Table of Contents

Electronic attachment with two reference implementations of Rec. ITU-T T.81 | ISO/IEC 10918-1.

Introduction

The ITU-T T.80-series | ISO/IEC 10918-series establishes guidelines and specifies requirements for coding of continuous-tone still images known under the name JPEG. Rec. ITU-T T.81 | ISO/IEC 10918-1 specifies the codestream format and the decoding process. Rec. ITU-T T.81 | ISO/IEC 10918-1 is designed primarily for compression of continuous-tone photographic content.

This Recommendation | International Standard provides reference software for Rec. ITU-T T.81 | ISO/IEC 10918-1. The software has been successfully compiled and tested on Linux and Windows operating systems and conforms to the decoder requirements set forth in Rec. ITU-T T.83 | ISO/IEC 10918-2. It has also been tested for conformance to Rec. ITU-T T.86 | ISO/IEC 10918-4 and ISO/IEC 18477-4.

Instructions for unpacking and building the software are found in Annexes A, and C. Instructions for its use are listed in Annexes B and D.

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

Information technology – Digital compression and coding of continuous-tone still images: Reference software

1 Scope

This Specification provides reference software for the coding technology specified in Recommendation ITU-T T.81 | ISO/IEC 10918-1. While the reference implementations also provide an encoder, conformance testing of their encoding process is beyond the scope of this Specification.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies. At the time of publication, the editions indicated in dated references 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.

– Recommendation ITU-T T.81 (latest) | ISO/IEC 10918-1 (latest), *Information technology – Digital compression and coding of continuous-tone still images: Requirements and guidelines*.

3 Definitions

For the purposes of this Specification, the terms and definitions specified in Rec. ITU-T T.81 | ISO/IEC 10918-1 and the following apply.

3.1 **codestream; JPEG file**: Sequence of bytes.

3.2 **pgx format; portable graphics format**: Image format describing integer-based continuous-tone images.

NOTE – For the purposes of this Specification, the image format is as specified in Rec. ITU-T T.803 | ISO/IEC 15444-4.

3.3 **pnm format; portable any map format**: Image format describing integer-based continuous-tone images of either one or three components consisting of a header determining image dimensions and sample precision and component-interleaved image samples encoded as 8-bit or 16-bit big-endian integers.

NOTE – For a specification of the pnm format, see Bourke (1997).

3.4 **R′G′B′**: Colour space that describes a colour by three gamma-corrected coordinates relative to three colour primaries.

3.5 **upsampling**: Procedure that increases the spatial or temporal sampling rate of a time-discretely sampled signal.

3.6 **Y′C$_B$C$_R$**: Colour space that describes a colour by 1 luma coordinate and 2 chroma coordinates derived from a gamma-corrected R′G′B′ colour space by a linear transformation.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply:

DCT Discrete Cosine Transform

DNL Define Number of Lines

IDCT Inverse Discrete Cosine Transform

MCU Minimum Coded Unit

POSIX Portable Operating System Interface

5 Conventions

Text in Times New Roman provides instructions, comments or details for the reader.

Text in `Courier New` indicates program input or output as necessary to either run the software or as generated by the software on the console.

6 Reference software

6.1 Purpose

For the purposes of this Recommendation | International Standard, the codestream shall conform to Rec. ITU-T T.81 | ISO/IEC 10918-1.

The purpose of this Specification is to provide:

- reference decoder software capable of decoding codestreams that conform to Rec. ITU-T T.81 | ISO/IEC 10918-1;
- sample encoder software capable of producing codestreams that conform to Rec. ITU-T T.81 | ISO/IEC 10918-1.

The use of the reference software is not required to implement an encoder or decoder in conformance to Rec. ITU-T T.81 | ISO/IEC 10918-1. Requirements established in Rec. ITU-T T.81 | ISO/IEC 10918-1 take precedence over the behaviour of the reference software.

6.2 Examples of use

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

- as an illustration of how to perform the decoding processes specified in Rec. ITU-T T.81 | ISO/IEC 10918-1;
- as the starting basis for the implementation of a decoder that conforms to Rec. ITU-T T.81 | ISO/IEC 10918-1;
- for (non-exhaustive) testing of the conformance of a codestream (or file) to the constraints specified in Rec. ITU-T T.81 | ISO/IEC 10918-1.

NOTE 1 – The lack of detection of any conformance violation by any reference software implementation cannot be considered as a definitive proof that the codestream under test conforms to Rec. ITU-T T.81 | ISO/IEC 10918-1.

Some examples of use for reference encoder software are as:

- an illustration of how to implement an encoding process that produces codestreams that conform to Rec. ITU-T T.81 | ISO/IEC 10918-1;
- a starting point for an implementation of an encoder that conforms to Rec. ITU-T T.81 | ISO/IEC 10918-1;
- a means of generating codestreams conforming to Rec. ITU-T T.81 | ISO/IEC 10918-1 for testing purposes;
- a means of demonstrating and evaluating examples of the quality that can be achieved by an encoding process that conforms to Rec. ITU-T T.81 | ISO/IEC 10918-1.

NOTE 2 – No guarantee of the quality that will be achieved by an encoder is provided by its conformance to Rec. ITU-T T.81 | ISO/IEC 10918-1, as the conformance is only defined in terms of specific constraints imposed on the syntax of the generated codestream and maximum tolerable errors of the discrete cosine transform (DCT) coefficients after reconstruction. In particular, while sample encoder software implementations could suffice to provide some illustrative examples of which quality can be achieved within Rec. ITU-T T.81 | ISO/IEC 10918-1, they provide neither an assurance of minimum guaranteed image encoding quality nor maximum achievable image encoding quality.

NOTE 3 – The computation resource characteristics in terms of program or data memory usage, execution speed, etc. of sample software encoder or decoder implementations cannot be construed as representative of the typical, minimal or maximal computational resource characteristics to be exhibited by implementations of some parts of Rec. ITU-T T.81 | ISO/IEC 10918-1.

6.3 General

The reference software implementations for Rec. ITU-T T.81 | ISO/IEC 10918-1 are available from ISO at https://standards.iso.org/iso-iec/10918/-7/ed-1/en and also from ITU at https://www.itu.int/rec/T-REC-T.873/en. Each of the two zip archives contains one reference software implementation.

– The file "reference1.zip" contains a reference implementation for all processes of Rec. ITU-T T.81 | ISO/IEC 10918-1. Unpacking and compilation of this software is explained in Annex A. Guidance on how to use this software is given in Annex B.

– The file "reference2.zip" contains a reference implementation for the baseline and extended Huffman and arithmetic coding DCT processes of Rec. ITU-T T.81 | ISO/IEC 10918-1. This software does not implement the lossless and hierarchical processes of Rec. ITU-T T.81 | ISO/IEC 10918-1. Unpacking and compilation of this software is explained in Annex C. Guidance on how to use this software is given in Annex D.

Annex A

Unpacking and compiling reference software A

(This annex does not form an integral part of this Recommendation | International Standard.)

Source code of the reference software implementation is provided in a zip archive available at https://standards.iso.org/iso-iec/10918/-7/ed-1/en, and https://www.itu.int/rec/T-REC-T.873/en. Unpacking a zip file is operating system specific. Under portable operating system interface (POSIX) compliant operating systems, open a command line window and enter

```
unzip reference1.zip
```

This will unpack all components of the software into the current directory.

To compile the software, follow these steps:

– for POSIX compliant operating systems, go to the directory into which the electronic attachment was unpacked, then enter on the command line

```
./configure
make
```

This assumes that a POSIX compliant shell is available, and the GNU compiler (make, gcc compiler and linker) are installed on the system. The reference software will then be built in the current directory, and a binary named "`jpeg`" will be created.

– For Microsoft Windows, the Visual Studio VS2010 or VS2013 compiler suite provides another option for building the software. A VS2010 solution file allowing loading and compiling the project can be found in the directory "vs10.0/jpeg". A solution file for VS2013 can be found in the directory "vs12.0/jpeg".

The compiler will generate a command line tool without any graphical interface that compresses images represented in the pnm format into JPEG and expands JPEG images into pnm or pgx files. pnm-files represent one component integer grey-scale or three-component integer colour data, whereas pgx files describe non-upsampled N-component raw sample data bare any colour interpretation.

For the purpose of testing for conformance to Rec. ITU-T T.81 | ISO/IEC 10918-1, the pgx output is preferable as it includes neither a transformation from $Y'C_BC_R$ to $R'G'B'$ and nor an upsampling process. These processes are not formally part of Rec. ITU-T T.81 | ISO/IEC 10918-1, but have been included in later standards. such as Rec. ITU-T T.871 | ISO/IEC 10918-5, also known as JFIF, or ISO/IEC 18477-1.

Annex B

Using reference software A

(This annex does not form an integral part of this Recommendation | International Standard.)

B.1 General

This annex describes the usage of the software prepared and compiled with the instructions given in Annex A.

The software can both compress images to JPEG files and expand such files back to pnm or pgx. The encoder and decoder are both contained in the same executable, named "jpeg". If a quality parameter (see clause B.2) is included on the command line, or the command line parameters request lossless coding, the executable encodes images. If only an input and an output file are provided, the software attempts to expand the input file to the output file.

The encoder requires one input image encoded in the pnm format that is to be compressed. For lossy compression, the -q parameter defines a quality parameter between 0 and 100. The `-p` parameter will enable lossless compression.

The command line parameters for the encoders are as follows:

```
jpeg [options] source target
```

where `options` is a set of command line options, all starting with a hyphen, source is the source image to be compressed, and the target is the output file to which the encoder writes the output codestream. Command line switches control the operations of the software. They are listed in clauses B.2 to B.8.

Decoding from a JPEG file to a ppm or pfm file requires only the input and output file name:

```
jpeg [options] source target
```

The only command line options recognized for decoding are `-c` and `-U`, which are described in subclause B.3.

B.2 Encoder options defining the quality of the image

The following options control the quality of the encoded images.

`-q quality` selects the encoding mode and defines the quality of the base image

B.3 Options controlling the colour space

The following options control how out of gamut colours and gamut extension is handled:

`-c` inserts a marker into the codestream that signals that the transformation from $Y'C_BC_R$ to $R'G'B'$ is disabled by the decoder

NOTE – Colour space interpretation and specifically the usage of this marker is beyond the scope of Rec. ITU-T T.81 | ISO/IEC 10918-1 and is further specified in Rec. ITU-T T.871 | ISO/IEC 10918-5 and ISO/IEC 18447-1.

B.4 Options controlling the scan generation and entropy coding

The following parameters control the entropy coding and generation of scan patterns for JPEG scans. Especially the `-h` parameter (optimize Huffman tables) is recommended, and for most configurations even necessary as Rec. ITU-T T.81 | ISO/IEC 10918-1 does not provide for all modes examples for tables the encoder could choose.

`-h` optimize the Huffman tables

`-v` use progressive instead of sequential encoding

`-p` use the lossless (predictive) mode of Rec. ITU-T T.81 | ISO/IEC 10918-1

`-y N` use hierarchical coding with `N` hierarchical layers. If `N` is 0, a two-scan hierarchical process is initiated of which the first scan is a DCT process and the second hierarchical scan is a lossless coding process

`-a` selects arithmetic coding instead of Huffman coding. Can be combined with `-v`, `-p` and `-y`

B.5 Options controlling the quantizer

The following parameters control the quantizer:

`-qt n` defines the quantization table

The following tables are defined:

- `n` = 0 the example tables from Rec. ITU-T T.81 | ISO/IEC 10918-1 (default)
- `n` = 1 a completely flat table that should be PSNR-optimal
- `n` = 2 a MS-SSIM (see Wang *et al.* (2003)) optimized table
- `n` = 3 the table suggested by ImageMagick
- `n` = 4 an HSV-PSNR optimized table
- `n` = 5 the table from Klein *et al*. (1992)
- `n` = 6 the table from Watson *et al*. (1997)
- `n` = 7 the table from Ahumada and Peterson (1993)
- `n` = 8 the table from Peterson *et al*. (1993)

`-dz` enables a deadzone quantizer which replaces the default uniform quantizer

B.6 Options controlling the subsampling of components

The following parameters control the subsampling of components. Unlike other implementations, this software does not specify the subsampling by means of the minimum coded unit (MCU) sizes, but directly by giving the subsampling factors. By default, the software uses 4:4:4 sampling. 4:2:0 chroma subsampling is, for example, defined by the parameters `-s 1x1,2x2,2x2,` 4:2:2 subsampling by `-s 1x1,2x1,2x1`.

`-s WxH,...` define subsampling factors for all components

NOTE – The arguments to `-s` are not MCU sizes.

B.7 Miscellaneous options

The following options control miscellaneous encoder options.

`-z mcus` defines the restart interval size, if the argument is 0, restart markers are disabled

`-n` signals the image height by a define number of lines (DNL) marker instead of writing the image height into the frame header

B.8 Decoder options

The following options control the decoding process.

`-c` disables the transformation from $Y'C_BC_R$ to $R'G'B'$

NOTE – This transformation is not part of Rec. ITU-T T.81 | ISO/IEC 10918-1 but has been specified in Rec. ITU-T T.871 | ISO/IEC 10918-5.

`-U` writes the output in pgx format rather than pnm format and disables upsampling of components and the transformation from $Y'C_BC_R$ to $R'G'B'$

Neither upsampling nor this transformation are formally part of the Rec. ITU-T T.81 | ISO/IEC 10918-1 process and hence should be disabled for the purpose of conformance testing with Rec. ITU-T T.83 | ISO/IEC 10918-2.

Annex C

Unpacking and compiling reference software B

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

The source code of the software is provided in a zip archive available at https://standards.iso.org/iso-iec/10918/-7/ed-1/en and https://www.itu.int/rec/T-REC-T.873/en. Unpacking a zip file is operating system specific. Under POSIX compliant operating systems, open a command line window and enter

```
unzip reference2.zip
```

This will unpack all components of the software into the current directory.

To compile the software, follow these steps:

– In order to compile a command line tool that supports the 8 bit baseline, sequential and progressive DCT coding modes on POSIX compliant operating systems, go to the directory into which the electronic attachment was unpacked, then enter on the command line

```
./configure
make
```

The following tools are required for successful building:

– a POSIX compliant shell
– autoconf 2.56 or later
– automake 1.7
– libtool 1.4 or later
– The GNU compiler (make, gcc compiler and linker)
– The NASM assembler (NASM Development Team (Internet)) or Yasm assembler (Yasm (Internet))

The reference software will then be built in the current directory consisting of two binaries, the encoder `"cjpeg"` and the decoder `"djpeg"`, and a compliance-testing enabled decoder `"djpegtopgx"`. The binaries created this way will be only capable of encoding and decoding to the 8 bit sequential or progressive DCT modes.

– In order to compile binaries that support the 12 bit sequential and progressive Huffman-coded DCT modes, enter the following sequence of commands on the command line:

```
./configure --with-12-bit
make
```

NOTE – The supplied software cannot be built in such a way that both 8 bit and 12 bit modes are supported simultaneously by the same executable.

– For Microsoft Windows, the following tools are required to build the software:
  – The Visual Studio VS2008, VS2010 or VS2013 compiler suite
  – CMake 2.8.11 (Kitware (Internet-1))

The following steps need to be performed to compile the software:

– Generate a Visual Studio project file for building the tool, which can be done either through the CMake graphical user interface or from the command line. For the latter, go to the directory into which sources have been unpacked, then enter

```
cmake -G "Visual Studio 10 2010".
```

where the argument behind `-G` needs to be adapted to the target version of Visual Studio. For a list of supported platforms, see Kitware (Internet-2).

The command in the previous paragraph will generate suitable project files in the current directory that can be opened directly from the Visual Studio IDE. Details of the building procedure can be found in the file BUILDING.md in the source directory of the software.

- Open the Visual Studio project file generated by the step in the previous paragraph from the graphical user interface and rebuild the software from the menu.
- The compiler will generate two command line tools: "`cjpeg`" will compress images represented in pnm (Picture AnyMap) into JPEG; and "`djpeg`" expands JPEG and JPEG XT part 1 images into pnm-files.

Annex D

Using reference software B

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

D.1 General

This annex describes the usage of the software prepared and compiled with the instructions given in Annex C.

The compilation process results in three executables. The "cjpeg" executable compresses pnm files into JPEG files, the "djpeg" file expands JPEG files into pnm files and the "djpegtopgx" files expands JPEG files into pgx files. Depending on the build process, the executables will either support the 8 bit or the 12 bit DCT modes of Rec. ITU-T T.81 | ISO/IEC 10918-1. There is no integrated version of the software that supports both the 8 bit and the 12 bit modes from the same executable.

The encoder "cjpeg" requires a single input image encoded in the pnm format and compresses its input stream to a JPEG file on its output stream, with the –quality option controlling approximately the quality and, through that, the byte size of the generated stream.

```
cjpeg [options] <source >target
```

where options is a set of command line options, all starting with a hyphen, source is the source image to compressed, and the target is the output file to which the encoder writes its output. Command line switches control the operations of the software.

Decoding from a JPEG file to a ppm file requires the "djpeg" decoder to be called as follows:

```
djpeg [options] <source >target
```

NOTE – In this command line, the angle brackets are part of the input/output redirection of the operating system. Without them, "djpeg" reads from the input stream and writes to the output stream.

Decoding from a JPEG file to a pgx file requires the "djpegtopgx" decoder to be called as follows:

```
djpegtopgx source target
```

The djpegtopgx decoder does not take any command line options beyond the compressed stream and output file.

D.2 Encoder options defining the quality of the base and full image

The following options of the cjpeg command control the quality of the encoded images:

-q quality — defines the quality of the image, where the quality parameter is a number between 0 and 100, where 0 indicates the lowest and 100 the highest quality

D.3 Encoder options controlling the colour space

The following options of the cjpeg command control the colour space:

-grayscale — creates monochrome JPEG files

-rgb — Includes a marker in the codestream that instructs the decoder to disable the $Y'C_BC_R$ to $R'G'B'$ transformation, and the encoder encodes the source image data directly in $R'G'B'$ space without a conversion to $Y'C_BC_R$. This marker and the corresponding transformation between $R'G'B'$ and $Y'C_BC_R$ is not part of Rec ITU-T T.81 | ISO/IEC 10918-1, but has been specified in Rec. ITU-T T.871 | ISO/IEC 10918-5 and ISO/IEC 18477-1.

D.4 Encoder options controlling the scan generation and entropy coding

The following command line parameters of the cjpeg command control the entropy coding process.

`-optimize`	optimizes the Huffman tables instead of using the example Huffman tables from Rec. ITU-T T.81 | ISO/IEC 10918-1
`-progressive`	use the progressive rather than sequential process for encoding data
`-arithmetic`	use arithmetic coding rather than Huffman coding. Can be combined with `-progressive`
`-baseline`	enforce that quantization parameters follow the more restrictive baseline Huffman process rather than the (extended) sequential process

D.5 Encoder options controlling the DCT implementation

The following `cjpeg` command line parameters control the choice of the DCT implementation.

`-dct int`	use an integer-based DCT process
`-dct fast`	use a fast, less accurate DCT process

NOTE – While this DCT process conforms to Rec. ITU-T T.83 | ISO/IEC 10918-2, it is not sufficiently accurate to pass the conformance requirements of ISO/IEC 18477-1.

`-dct float`	use a floating-point based DCT process

D.6 Encoder options controlling the subsampling of components

The following `cjpeg` command line parameters control the subsampling of components.

`-sample HxV,[HxV,[...]]`	Set the component MCU sizes. If only a single MCU size is given, this applies to the luma component only and all other MCU sizes are 1x1. The default is 4:2:0 subsampling, which is identical to the command line option `-sample 2x2`.

Unlike reference software A, the parameters define the MCU sizes, not the subsampling factors derived from them. For 4:4:4 sampling, set all MCU sizes to `1x1`. Since chroma MCU sizes default to 1x1, this is identical to the command line parameter `-sample 1x1`. For 4:2:0 subsampling, the MCU sizes of the first component are 2x2, and 1x1 for the second and third component. Due to the default MCU sizes for the chroma components, the command line argument `-sample 2x2` is sufficient to enable 4:2:0 subsampling. Similarly, 4:2:2 subsampling is requested with `-sample 2x1`.

D.7 Miscellaneous encoder options

`-targa`	use the targa format instead of the pnm image format for the input images
`-restart N`	insert every `N` MCUs a restart marker into the codestream
`-smooth N`	smooth dithered input prior to compression where `N` is a smoothing strength between 1 and 100
`-maxmemory N`	Specify how much memory the encoder may allocate at most, where `N` is given in kilobytes
`-outfile name`	write to the given output file name instead of the standard output of the command
`-memdst`	write to memory instead of a file (useful for benchmarking)
`-verbose`	generate debug output
`-debug`	generate debug output, identical to `-verbose`
`-version`	print version information and exit immediately
`-qtables name`	read the quantization table from a given file rather than using the example tables from Rec. ITU-T T.81 | ISO/IEC 10918-1
`-qslots N[N,[...]]`	define the quantization table slots per component

`-scans name` read a scan description from a given file instead of using the pre-defined scan definition for the progressive mode.

D.8 Decoder options controlling the choice of the inverse discrete cosine transform

The following options impact the selection of the inverse discrete cosine transform (IDCT) process within the `djpeg` binary. For the purpose of reference testing, the default option should be used as otherwise the precision of the IDCT may not be precise enough.

`-dct int` use an integer based IDCT transformation (default)

`-dct fast` use a faster, less precise IDCT implementation

NOTE – While the IDCT process selected by this option conforms to Rec. ITU-T T.83 | ISO/IEC 10918-2, the precision offered by this process is not sufficient to ensure conformance to ISO/IEC 18477-1.

`-dct float` use a floating-point based IDCT implementation

D.9 Decoder options selecting the output file format

The following set of options control the selection of the file format of the output. Note that the `djpeg` executable includes a conversion from $Y'C_BC_R$ to R′G′B′ and an upsampling step and is hence unsuitable for the purpose of conformance testing with Rec. ITU-T T.83 | ISO/IEC 10918-2. For this purpose, use the `djpegtopgx` binary instead.

`-bmp` use the Windows bmp format to represent the output image

`-gif` write the reconstructed image in the gif format

NOTE – The gif format is restricted to palette mapped images and is hence unsuitable for a faithful representation of continuous-tone images.

`-os2` use an OS/2 compatible bmp format to represent the reconstructed image

`-pnm` write the reconstructed image in the portable any map format (default)

`-targa` Write the reconstructed image in the targa format

D.10 Decoder options controlling the rendering of the output image

The following set of options trigger additional colour reduction and rendering operations that imply a lossy conversion process that is not part of Rec. ITU-T T.81 | ISO/IEC 10918-1, Rec. ITU-T T.871 | ISO/IEC 10918-5 nor ISO/IEC 18477-1. The operations enabled by the following set of options are therefore unsuitable for reference testing as they introduce additional distortion.

`-colors N` reduce the output to a palette-mapped image using at most `N` distinct colours

`-map FILE` Remap the image to the palette recorded in the given file

`-onepass` use a fast, one-pass colour quantization process instead of a more precise two-pass quantization

`-fast` combines various options to speed up the colour reproduction. This is equivalent to `-dct fast -dither ordered -colors 216 -nosmooth`

`-grayscale` enforce monochromatic output even if the JPEG codestream includes more than one component

`-rgb` write the output file with three components, even if the JPEG codestream contains only a single component

`-rgb565` reduce the bit precision of R′G′B′ output to 5 bits red, 6 bits green, and 5 bits blue

`-dither fs` for reducing the number of colours, use dithering with the Floyd-Steinberg algorithm

`-dither none` disable the use of any dithering for colour reduction