

First edition
1999-12-01

Corrected and reprinted
2000-09-15

Information technology — Lossless and near-lossless compression of continuous-tone still images: Baseline

*Technologies de l'information — Compression «lossless» proche de
«lossless» des images fixes à ton continu: Ligne de base*

Reference number
ISO/IEC 14495-1:1999(E)



PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

© ISO/IEC 1999

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 734 10 79
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

CONTENTS

	<i>Page</i>
1 Scope	1
2 Normative references.....	1
2.1 Identical Recommendations International Standards.....	1
2.2 Additional references	1
3 Definitions, abbreviations, symbols and conventions	2
3.1 Definitions.....	2
3.2 Abbreviations	4
3.3 Symbols.....	5
4 General description.....	7
4.1 Purpose	7
4.2 Coding principles	7
4.3 Source image	8
4.4 Encoding process.....	9
4.5 Decoding process	10
4.6 Coding of multiple component images.....	10
4.7 Compressed image data.....	10
4.8 Interchange format	10
5 Interchange format requirements.....	11
6 Encoder requirements	11
7 Decoder requirements.....	11
8 Conformance testing.....	12
8.1 Purpose	12
8.2 Encoder conformance tests.....	12
8.3 Decoder conformance tests	12
Annex A – Encoding procedures for a single component	16
A.1 Coding parameters and compressed image data.....	16
A.2 Initialisations and conventions	16
A.3 Context determination	18
A.4 Prediction	19
A.5 Prediction error encoding	21
A.6 Update variables	22
A.7 Run mode	23
A.8 Flow of encoding procedures	26
Annex B – Multi-component images	28
B.1 Introduction	28
B.2 Line interleaved mode.....	28
B.3 Sample interleaved mode	29
B.4 Minimum Coded Unit (MCU).....	30
Annex C – Compressed data format.....	31
C.1 General aspects of the compressed data format specification	31
C.2 General JPEG-LS coding syntax	31
C.3 Abbreviated format for compressed image data.....	37
C.4 Abbreviated format for table-specification data.....	37

	<i>Page</i>
Annex D – Control procedures	38
D.1 Control procedure for encoding an image	38
D.2 Control procedure for encoding a frame	38
D.3 Control procedure for encoding a scan.....	38
D.4 Control procedure for encoding a restart interval.....	40
D.5 Control procedure for encoding a Minimum Coded Unit (MCU).....	40
Annex E – Conformance Tests	42
E.1 Test images.....	42
Annex F – Decoding procedures	45
F.1 Process flow	45
Annex G – Description of the coding process	47
G.1 Context modelling.....	47
G.2 Encoding in the regular coding mode.....	48
G.3 Encoding in the run mode	50
Annex H – Examples and guidelines	51
H.1 Introduction	51
H.2 Example of how bits are output in the bit stream	51
H.3 Detailed coding example	51
H.4 Example image data	57
H.5 Use of SPIFF with JPEG-LS compressed image data	63
Annex I – Bibliography	65
Annex J – Patents	66

Included files and programs:

- JPEG-LS reference implementation
- JPEG-LS conformance testing image set
- Auxiliary programs and examples.

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

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

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 14495 may be the subject of patent rights other than those identified in annex J. ISO and IEC shall not be held responsible for identifying any and all such patent rights.

International Standard ISO/IEC 14495-1 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 identical text is published as ITU-T Recommendation T.87.

ISO/IEC 14495 consists of the following parts, under the general title *Information technology — Lossless and near-lossless compression of continuous-tone still images*:

- *Part 1: Baseline*
- *Part 2: Extensions*

Annexes A to E form a normative part of this part of ISO/IEC 14495. Annexes F to J are for information only. Annex J is only applicable to ISO/IEC 14495-1.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – LOSSLESS AND NEAR-LOSSLESS COMPRESSION OF CONTINUOUS-TONE STILL IMAGES – BASELINE

1 Scope

This Recommendation | International Standard defines a set of lossless (bit-preserving) and nearly lossless (where the error for each reconstructed sample is bounded by a pre-defined value) compression methods for coding continuous-tone, gray-scale, or colour digital still images.

This Recommendation | International Standard

- specifies a process for converting source image data to compressed image data;
- specifies processes for converting compressed image data to reconstructed image data;
- specifies coded representations for compressed image data;
- provides guidance on how to implement these processes in practice.

2 Normative references

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

2.1 Identical Recommendations | International Standards

- CCITT Recommendation T.81 (1992) | ISO/IEC 10918-1:1994, *Information technology – Digital compression and coding of continuous-tone still images: Requirements and guidelines*.
- ITU-T Recommendation T.83 (1994) | ISO/IEC 10918-2:1995, *Information technology – Digital compression and coding of continuous-tone still images: Compliance testing*.
- ITU-T Recommendation T.84 (1996) | ISO/IEC 10918-3:1997, *Information technology – Digital compression and coding of continuous-tone still images: Extensions*.
- ITU-T Recommendation T.84/Amd.1¹ | ISO/IEC 10918-3/Amd.1¹, *Information technology – Digital compression and coding of continuous-tone still images: Extensions – Amendment 1*.

2.2 Additional references

- ISO/IEC 646:1991, *Information technology – ISO 7-bit coded character set for information interchange*.
- ISO 5807:1985, *Information processing – Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts*.
- ISO/IEC 9899:1990, *Programming languages – C*.

¹ Currently at the stage of draft.

3 Definitions, abbreviations, symbols and conventions

3.1 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

- 3.1.1 $\lfloor i \rfloor$, **floor**: Indicates the largest integer not exceeding i .
- 3.1.2 $\lceil i \rceil$, **ceiling**: Indicates the smallest integer not exceeded by i .
- 3.1.3 **abs(i)**: The absolute value of i : $-i$ if $i < 0$, i otherwise.
- 3.1.4 **abbreviated format**: A representation of compressed image data which is missing some or all of the mapping table specifications required for decoding, or a representation of mapping tables without frame headers, scan headers, and coded image data.
- 3.1.5 **application environment**: The standards for data representation, communication, or storage, which have been established for a particular application.
- 3.1.6 **bias**: Deviation from zero of accumulated prediction errors.
- 3.1.7 **bit stream**: Partially encoded or decoded sequence of bits.
- 3.1.8 **causal template**: A set of fixed relative positions of samples (with respect to the current sample being coded) which have been previously coded according to a pre-specified scan sequence.
- 3.1.9 **coded image data segment**: The coded representation of one restart interval.
- 3.1.10 **coding**: Encoding or decoding.
- 3.1.11 **coding parameters**: Integers used to specify the encoding process.
- 3.1.12 **(coding) process**: A general term for referring to an encoding process, a decoding process, or both.
- 3.1.13 **colour image**: A continuous-tone image that has more than one component.
- 3.1.14 **columns**: Samples per line in a component.
- 3.1.15 **component**: One of the two-dimensional arrays which comprise an image.
- 3.1.16 **compressed data**: Either compressed image data or parameter specification data or both.
- 3.1.17 **compressed image (data)**: A coded representation of an image, as specified in this Recommendation | International Standard.
- 3.1.18 **compression**: Reduction in the number of bits used to represent source image data.
- 3.1.19 **context**: Function of samples in the causal template used to determine the coding of the present sample.
- 3.1.20 **context modelling**: Procedure estimating a probability distribution of prediction error from the context.
- 3.1.21 **continuous-tone image**: An image whose components have more than one bit per sample.
- 3.1.22 **decoder**: An embodiment of a decoding process and a sample transformation process.
- 3.1.23 **decoding process**: A process which takes as its input compressed image data and outputs a reconstructed image.
- 3.1.24 **encoder**: An embodiment of an encoding process.
- 3.1.25 **encoding process**: A process which takes as its input a source image and outputs compressed image data.
- 3.1.26 **frame**: A group of one or more scans through the data of one or more of the components in an image.
- 3.1.27 **frame header**: A marker segment that contains a start-of-frame marker and associated frame parameters that are coded at the beginning of a frame.
- 3.1.28 **Golomb coding**: A special case of Huffman coding matched to geometric distributions.
- 3.1.29 **(local) gradient**: Either a vector of differences between values of samples in the causal template, or each difference separately.

- 3.1.30 gray-scale image:** A continuous-tone image that contains only one component.
- 3.1.31 horizontal sampling factor:** The relative number of horizontal samples of a particular component with respect to the number of horizontal samples in the other components.
- 3.1.32 Huffman encoding:** A prefix coding procedure which assigns a variable length code to each input symbol, so that the total code length is minimised.
- 3.1.33 image:** A set of two-dimensional arrays of integer data.
- 3.1.34 image data:** Either source image data or reconstructed image data.
- 3.1.35 interchange format:** The representation of compressed image data for exchange between application environments.
- 3.1.36 interleaved:** The descriptive term applied to the repetitive multiplexing of groups of data from each component in a scan in a specified order.
- 3.1.37 JPEG-LS:** Used to refer globally to the encoding and decoding processes in this Recommendation | International Standard and their embodiment in applications.
- 3.1.38 JPEG-LS preset coding parameters:** Coding parameters for which a normative set of default values is specified.
- 3.1.39 JPEG-LS preset parameters:** A coding parameter or a mapping table specified in an LSE marker segment.
- 3.1.40 line interleaved:** The mode of operation in which the interleaved entities are lines.
- 3.1.41 lossless:** A descriptive term for the encoding and decoding processes in which the output of the decoding process is identical to the input to the encoding process.
- 3.1.42 lossless coding:** The mode of operation which refers to any one of the coding processes defined in this Recommendation | Standard in which all of the procedures are lossless.
- 3.1.43 lossy:** A descriptive term for encoding and decoding processes which are not lossless.
- 3.1.44 mapping table:** A table used in a sampling mapping procedure.
- 3.1.45 marker:** A two-byte code in which the first byte is hexadecimal FF (X'FF') and the second byte is a value between 1 and hexadecimal FE (X'FE')
- 3.1.46 marker segment:** A marker and associated set of parameters.
- 3.1.47 max(i,j):** The largest of i or j : i if $i > j$, j otherwise.
- 3.1.48 min(i,j):** The smallest of i or j : i if $i < j$, j otherwise.
- 3.1.49 minimum coded unit:** The smallest group of samples that is coded.
- 3.1.50 near-lossless:** A description term for lossy encoding and decoding processes and procedures in which the output of the decoding process is such that each reconstructed image sample differs from the corresponding one in the input to the encoding process by not more than a pre-specified value.
- 3.1.51 near-lossless coding:** The mode of operation which refers to any one of the encoding process, decoding process, or both, defined in this Recommendation | International Standard in which some of the procedures are near-lossless.
- 3.1.52 non-interleaved:** The descriptive term applied to the data processing sequence when the scan has only one component.
- 3.1.53 parameter specification data:** The coded representation of the parameters used in the encoder and decoder.
- 3.1.54 point transform:** Scaling of a sample.
- 3.1.55 precision:** Number of bits allocated to a particular sample.
- 3.1.56 predicted sample value:** The output from the predictor.
- 3.1.57 prediction correction:** The procedure that compensates for systematic biases in prediction.
- 3.1.58 prediction error:** Difference between the current sample and the predicted sample value.

- 3.1.59 predictor:** The procedure that computes a predicted sample value from previously reconstructed samples.
- 3.1.60 procedure:** A set of steps which accomplishes one of the tasks which comprise an encoding or decoding process.
- 3.1.61 reconstructed image (data):** An image which is the output of a decoding process or a sample transformation process.
- 3.1.62 reconstructed sample:** The sample value reconstructed by the decoder. This equals the original sample value in lossless coding, or differs from the original sample value by at most **NEAR** in magnitude in near-lossless coding.
- 3.1.63 regular mode:** Mode of operation when coding samples while not in the run mode.
- 3.1.64 restart interval:** The integer number of MCUs processed as an independent sequence within a scan.
- 3.1.65 restart marker:** The marker that separates two restart intervals in a scan.
- 3.1.66 run:** A sequence of consecutive samples whose values are identical for lossless coding, or are within the limits required for near-lossless coding, and which is contained in the current image line.
- 3.1.67 run length:** Number of samples in a run.
- 3.1.68 run mode:** Mode of operation while coding runs.
- 3.1.69 run interruption sample:** The sample following the last sample in a run when the run terminates before the end of line.
- 3.1.70 sample:** One element in the two-dimensional array which comprises a component.
- 3.1.71 sample interleaved:** The mode of operation in which the interleaved entities are samples.
- 3.1.72 sample mapping procedure:** A procedure that maps each sample value output by a decoding process to a reconstructed sample value by means of mapping tables.
- 3.1.73 sample transformation process:** A sample mapping procedure followed by an inverse point transform.
- 3.1.74 sample value:** A non-negative integer indicating the image information in an image sample.
- 3.1.75 scan:** A single pass through the data for one or more of the components in the image.
- 3.1.76 scan header:** A marker segment that contains a start-of-scan marker and associated scan parameters that are coded at the beginning of a scan.
- 3.1.77 source image (data):** An image used as input to an encoder.
- 3.1.78 unary code:** The unary code of a non-negative integer number n is composed of n zero bits followed by a one bit.
- 3.1.79 vertical sampling factor:** The relative number of vertical samples of a particular component with respect to the number of vertical samples in the other components in the frame.

3.2 Abbreviations

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

BPS	Bits Per Sample
JPEG	Joint Photographic Experts Group – The joint ISO/ITU committee responsible for developing standards for continuous-tone still picture coding. It also refers to the standards produced by this committee: CCITT Rec. T.81 ISO/IEC 10918-1, ITU-T Rec. T.83 ISO/IEC 10918-2, and ITU-T Rec. T.84 ISO/IEC 10918-3.
LSB	Least Significant Bit
MCU	Minimum Coded Unit
MSB	Most Significant Bit
PGM	Portable Grey Map
PPM	Portable Pix Map
SPIFF	Still Picture Interchange File Format

3.3 Symbols

The following symbols used in this Recommendation | International Standard are listed below. A convention is used that parameters which are fixed in value during the encoding of a scan are indicated in **boldface** capital letters, and variables which change in value during the encoding of a scan are indicated in *italicised* letters.

a, b, c, d	positions of samples in the causal template
$A[0..366]$	367 counters storing accumulated prediction error magnitudes
Ah	ignored field in scan header
Al	successive approximation bit position, low
AppendToBitStream()	a function in the C programming language
APP_n	marker reserved for application segments
$B[0..364]$	365 counters for storing bias values
BASIC_T1, BASIC_T2, BASIC_T3	basic default threshold values
bpp	number of bits needed to represent MAXVAL (not less than 2)
C	SPIFF parameter
$C[0..364]$	365 counters storing prediction correction values
C_i	component identifier
CLAMP	out-of-range value clamping function
COM	comment marker
ComputeRx()	a function in the C programming language
$D1, D2, D3$	local gradients
DNL	define-number-of-lines marker
DRI	define restart interval marker
<i>EMErrval</i>	<i>Errval</i> mapped to non-negative integers in run interruption mode
ENTRIES	number of yet unspecified mapping table entries
EOI	end-of-image marker
<i>EOLine</i>	end of line indicator, used in run mode
<i>Errval</i>	prediction error (quantized or unquantized, before and after modulo reduction)
g	the order of a Golomb code
GetNextSample()	a function in the C programming language
$G(k)$	Golomb code function
<i>glimit</i>	number of bits to which the length of a Golomb code word is limited
H_i	horizontal sampling factor for the <i>i</i> th component
H_{max}	largest horizontal sampling factor
i, j, n	integers
$i..j$	the set of integers between <i>i</i> and <i>j</i> , including <i>i</i> and <i>j</i> .
ILV	indication of the interleave mode used for the scan
I_x	the value of the current sample in the input image
$J[0..31]$	32 variables indicating order of run-length codes
JPG	marker reserved for JPEG extensions
JPG_n	marker reserved for JPEG extensions
k	Golomb coding variable for regular mode
$LG(k, glimit)$	limited length Golomb code function
LIMIT	the value of <i>glimit</i> for a sample encoded in regular mode
LI	JPEG-LS preset parameters marker segment length specifier
LSE	JPEG-LS preset parameters marker

<i>m</i>	modulo counter for restart marker
<i>MErrval</i>	<i>Errval</i> mapped to non-negative integers in regular mode
<i>map</i>	auxiliary variable for error mapping at run interruption
MAXTAB, MAXTABX	maximum index to a mapping table
MAXVAL	maximum possible image sample value over all components of a scan
MAX_C	maximum allowed value of <i>C</i> [0..364], equal to 127
MIN_C	minimum allowed value of <i>C</i> [0..364], equal to –128
ModRange()	a function in the C programming language
<i>N</i> [0..366]	367 counters for frequency of occurrence of each context
Nb	number of samples in an MCU
Nf	number of components in a frame
<i>Nn</i> [365..366]	2 counters for negative prediction error for run interruption
NEAR	difference bound for near-lossless coding
Ns	number of components in a scan
$\wp$	probability distribution
P	sample precision
Pt	point transform parameter
<i>Px</i>	predicted value for the current sample
<i>Q</i>	context, determined from <i>Q1</i> , <i>Q2</i> , <i>Q3</i>
<i>Q1</i> , <i>Q2</i> , <i>Q3</i>	region numbers to quantize local gradients
qbpp	number of bits needed to represent a mapped error value
<i>Qi</i>	one of the three quantized region numbers <i>Q1</i> , <i>Q2</i> , <i>Q3</i>
Quantize()	a function in the C programming language
<i>Ra</i> , <i>Rb</i> , <i>Rc</i> , <i>Rd</i>	reconstructed values of samples in the causal template
RANGE	range of prediction error representation
RegularModeProcessing	a label in the C programming language
RESET	threshold value at which <i>A</i> , <i>B</i> , and <i>N</i> are halved
<i>Rltype</i>	index for run interruption coding
<i>rk</i>	<i>rg</i> is the <i>rk</i> -th power of 2
<i>rg</i>	Golomb code order for run mode, a power of 2
RST_m	restart marker number <i>m</i>
RunModeProcessing	a label in the C programming language
<i>RUNcnt</i>	repetitive sample count for run mode
<i>RUNindex</i>	index for run mode order
<i>RUNval</i>	repetitive reconstructed sample value in a run
<i>Rx</i>	reconstructed value of the current sample
SIGN	auxiliary variable used to hold the sign of a context
SOF₅₅	JPEG-LS frame marker
SOI	start-of-image marker
SOS	start-of-scan marker
TABLE [0.. MAXTAB]	mapping table
T1 , T2 , T3	thresholds for local gradients
Td_iTa_i	replaced field in scan header
TEMP	auxiliary variable used in the calculation of the Golomb variable in run interruption coding

TID	mapping table identification number
Tm_i	mapping table selector for the <i>i</i> th component
Tq_i	field in frame header (not used in the procedures of this Recommendation International Standard).
VERS	SPIFF parameter
V_i	vertical sampling factor for the <i>i</i> th component
V_{max}	largest vertical sampling factor
Wt	width of table entries, in bytes
W_{xy}	number of bytes used to represent Y and X
x	current sample position
x_i	number of samples per line in the <i>i</i> th component
X	number of samples per line in the component with largest horizontal dimension
Xe	number of samples per line in the component with largest horizontal dimension, specified in an LSE marker segment
X'values'	values within the quotes are hexadecimal
Y	number of lines in the component with the largest vertical dimension
Ye	number of lines in the component with the largest vertical dimension, specified in an LSE marker segment
y_i	number of lines in the <i>i</i> th component

4 General description

4.1 Purpose

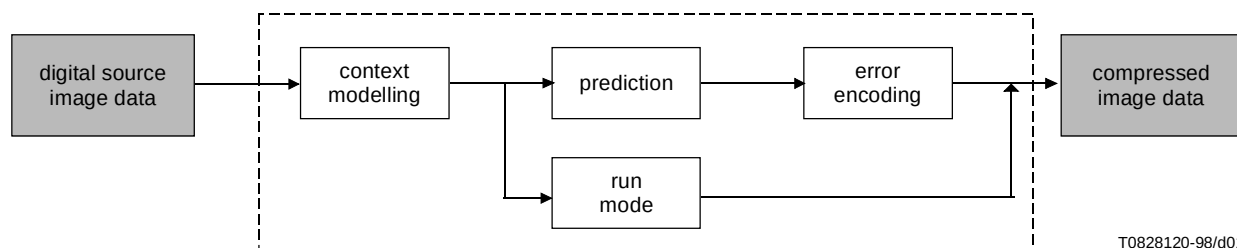
There are three main elements specified in this Recommendation | International Standard:

- Encoder:** An embodiment of an *encoding process*. An encoder takes as input *source image data* and *parameter specifications* and, by means of a specified set of *procedures*, generates as output *compressed image data*. Encoding procedures are specified in Annexes A, B, and D.
- Decoder:** An embodiment of a *decoding process* and a *sample transformation process*. A decoder takes as input compressed image data and parameter specifications and, by means of a specified set of procedures, generates as output *reconstructed image data*. Decoding procedures are described in Annex F.
- Interchange format:** A compressed image data representation which includes all parameter specifications used in the encoding process. The interchange format is for exchange between application environments. The interchange format is specified in Annex C.

4.2 Coding principles

The main procedures for the lossless (and near-lossless) encoding process specified in this Recommendation | International Standard are shown in Figure 1. Procedures in Figure 1 are presented in this clause in the order the encoding process is carried out (see Annex A).

In this Recommendation | International Standard, a source image is input to the encoder sample after sample in a pre-defined scan pattern, and lossless image compression is formulated as an inductive inference problem as follows. When coding the current sample, after having scanned past data, inferences can be made on the value of this sample by assigning a conditional probability $\wp$ for the value of the current image sample, conditioned on previously received samples. This inference method is called modelling. The minimum average code length contribution of the current sample is $-\log_2(\wp)$. For near-lossless image compression this principle is modified to use reconstructed values of the preceding samples (instead of the original values) as conditioning data. During the encoding process, shorter codes are assigned to more probable events. The decoder can reconstruct the conditional probability used to encode the current samples, since it depends only on already decoded data.



T0828120-98/d01

Figure 1 – Simplified encoder diagram

4.3 Source image

Source images to which the encoding process specified in this Recommendation | International Standard can be applied are defined in this clause.

4.3.1 Dimensions and sampling factors

As shown in Figure 2, a source image is defined to consist of N_f components. Each component, with unique identifier C_i , is defined to consist of a rectangular array of samples of x_i columns by y_i lines. The component dimensions are derived from two parameters, X and Y , where X is the maximum of the x_i values and Y is the maximum of the y_i values for all components in the frame. For each component, sampling factors H_i and V_i are defined relating component dimensions x_i and y_i to maximum dimensions X and Y , according to the following expressions:

$$x_i = \left\lceil X \times \frac{H_i}{H_{\max}} \right\rceil \quad \text{and} \quad y_i = \left\lceil Y \times \frac{V_i}{V_{\max}} \right\rceil$$

where $H_{\max}$ and $V_{\max}$ are the maximum sampling factors for all components in the frame.

As an example, consider an image having 3 components with maximum dimensions of 512 lines and 512 columns, and with the following sampling factors:

- Component 0 $H_0 = 4, V_0 = 1$
- Component 1 $H_1 = 2, V_1 = 2$
- Component 2 $H_2 = 1, V_2 = 1$

Then $X = 512$, $Y = 512$, $H_{\max} = 4$, $V_{\max} = 2$, and x_i and y_i for each component are:

- Component 0 $x_0 = 512, y_0 = 256$
- Component 1 $x_1 = 256, y_1 = 512$
- Component 2 $x_2 = 128, y_2 = 256$

NOTE – The X , Y , H_i , and V_i parameters are contained in the frame header of the compressed image data, whereas the individual component dimensions x_i and y_i are derived by the decoder. Source images with dimensions which do not satisfy the expressions above for x_i and y_i cannot be properly reconstructed.

4.3.2 Sample precision and point transform

A sample is an integer with precision P bits, with any value in the range 0 through $2^P - 1$. All samples of all components within an image shall have the same precision P . P is restricted to the range 2-16 bits.

Samples may optionally be divided by a power of 2 by a point transform applied prior to encoding. The point transform is an integer divide by 2^{Pt} , where Pt is the value of the point transform parameter. The output of the decoding process is re-scaled by multiplying by 2^{Pt} .

4.3.3 Orientation

Figure 2 indicates the orientation of an image component by the terms top, bottom, left, and right. The order by which the samples of an image component are input to the encoding procedures is defined to be left-to-right and top-to-bottom within the component. File formats (such as SPIFF, see 4.8.1) determine which edges of a source image are defined as top, bottom, left, and right.

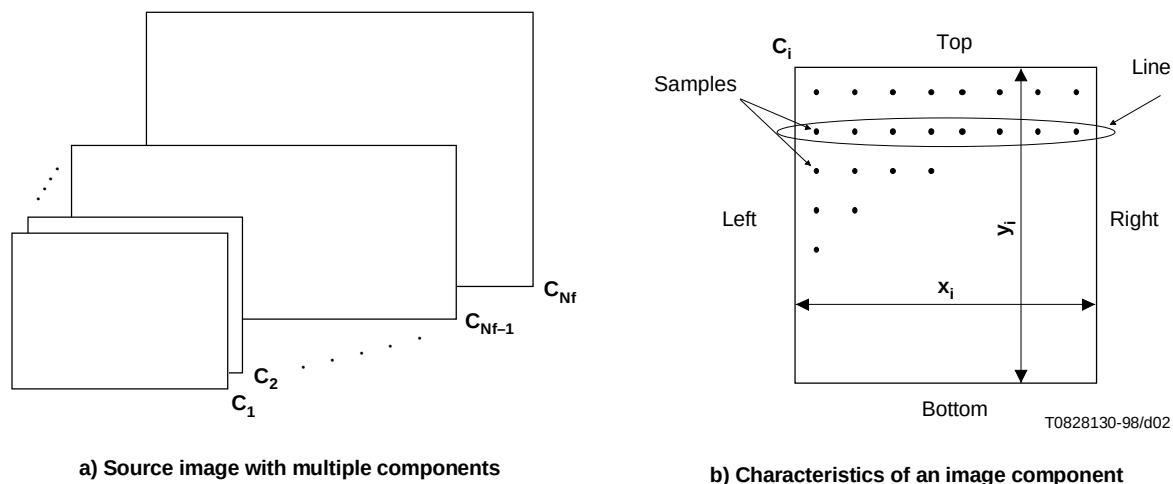


Figure 2 – Source image characteristics

4.4 Encoding process

4.4.1 Context modelling – Basics

The encoding process is described in outline below. The encoding process is specified in Annexes A, B, and D. An informative Annex, G, is included for additional explanation.

In this Recommendation | International Standard, the modelling approach used is based on the notion of "context". In context modelling, the encoding of each sample value is performed by conditioning on a small number of neighbouring samples. The context modelling procedure determines a probability distribution used to encode the current sample, whose position, x , is shown in Figure 3. The context is determined from four neighbourhood reconstructed samples at positions a , b , c , and d of the same component, as shown in Figure 3. From the values of the reconstructed samples at a , b , c , and d , the context first determines if the information in the sample x should be encoded in the regular or run mode:

- the run mode is selected when it is estimated from the context that successive samples are very likely to be nearly identical within the tolerances required for near-lossless coding (identical, for lossless coding);
- the regular mode is selected when it is estimated from the context that samples are not very likely to be nearly identical within the tolerances required for near-lossless coding (identical, for lossless coding).

4.4.2 Regular mode: Prediction and error encoding

In the regular mode, the context determination procedure is followed by a prediction procedure. The predictor combines the reconstructed values of the three neighbourhood samples at positions a , b , and c to form a predicted sample value at position x as shown in Figure 3. The prediction error is computed as the difference between the actual sample value at position x and its predicted value. This prediction error is then corrected by a context-dependent term to compensate for systematic biases in prediction. In the case of near-lossless coding, the prediction error is then quantized.

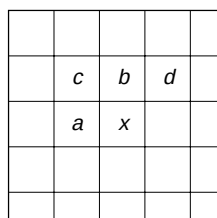


Figure 3 – Causal template used for context modelling and prediction

The corrected prediction error (further quantized for near-lossless coding) is then encoded using a procedure derived from Golomb coding (specified in Annex A and further described in Annex G).

NOTE – Golomb coding corresponds to Huffman coding for a geometric distribution.

The Golomb coding procedures specified in this Recommendation | International Standard depend on the context as well as prediction errors previously encoded for the same context.

4.4.3 Run mode

If the reconstructed values of the samples at *a*, *b*, *c*, and *d* are identical for lossless coding, or the differences between them (the local gradients, specified in Annex A) are within the bounds set for near-lossless coding, the context modelling procedure selects the run mode and the encoding process skips the prediction and error encoding procedures. In run mode, the encoder looks, starting at *x*, for a sequence of consecutive samples with values identical (or within the bound specified for near-lossless coding) to the reconstructed value of the sample at *a*. A run is ended by a sample of a different value (or one which exceeds the bound specified for near-lossless coding), or by the end of the current line, whichever comes first. The length information, which also specifies one of the above two run-ending alternatives, is encoded using a procedure specified in Annex A, which is extended from Golomb coding but has improved performance and adaptability.

4.5 Decoding process

The encoding and decoding processes are approximately symmetrical. Annex A specifies the encoding process, and Annex F describes the decoding process. The decoding process is followed by a sample mapping procedure which uses the value of each decoded sample as an index to a look-up table, provided in the compressed image data. The corresponding table entry might be of a different precision to that of the encoded sample.

NOTE – The sample mapping procedure is aimed at facilitating the use of this Recommendation | International Standard for the encoding of palletised and symbolic images. In the case of palletised images, the encoder would encode only one component, and the difference in precision would permit, for example, the display of single-component images as pseudo-colour ones. In the case of symbolic images, each decoded sample is a representation of the real image value, which is in the corresponding entry in the look-up table.

If no table is provided for a specific component, the output of the sample mapping procedure is identical to the input. The use of sample mapping is specified in Annex C.

4.6 Coding of multiple component images

The coding processes described in this Recommendation | International Standard can be applied to multiple components of an image, as well as to a single component image. Annex B describes how the coding processes shall be applied to images containing multiple components.

4.7 Compressed image data

The compressed image data output by the encoding process consists of marker segments and coded image data segments. The marker segments contain information required by the decoding process, including the image dimensions. The marker syntax is specified in Annex C, while the procedures for encoding the image data are specified in Annexes A, B, and D.

4.8 Interchange format

The interchange format is specified in Annex C based on Annex B of CCITT Rec. T.81 | ISO/IEC 10918-1. The interchange format allows a decoder to decode the compressed image data, regardless of specific application environments. Applications requiring further image information, for example for identifying an image to higher level applications, are recommended to use the SPIFF image file format, specified in Annex F of ITU-T Rec. T.84 | ISO/IEC 10918-3.

The following extension has been made to the SPIFF file format to allow its use with image data coded according to this Recommendation | International Standard:

4.8.1 Addition to SPIFF file header

In the file header described in Annex F.2.1 of ITU-T Rec. T.84 | ISO/IEC 10918-3, an additional value is allocated to parameter **C** describing the compression type. For data streams coded in accordance with this Recommendation | International Standard, **C** has the value X'06'. This has been standardised in ITU-T Rec. T.84 | ISO/IEC 10918-3 Amd.1, which also changes the version number **VERS** field in the SPIFF header to X'0200'.

5 Interchange format requirements

The interchange format is the coded representation of compressed image data for exchange between application environments.

The interchange format requirements are that any compressed image data represented in interchange format shall comply with the syntax and code assignments appropriate for the coding processes defined in this Recommendation | International Standard, as specified in Annex C.

6 Encoder requirements

An encoding process converts source image data (as defined in 4.2) to compressed image data (as defined in Annex C). Annexes A, B and D specify the encoding process.

An encoder is an embodiment of the encoding process. In order to conform with this Recommendation | International Standard, an encoder shall satisfy at least one of the following two requirements.

An encoder shall:

- a) convert source image data to compressed image data which conform to the interchange format syntax specified in Annex C;
- b) convert source image data to compressed image data which comply with the abbreviated format for compressed image data syntax specified in Annex C.

Conformance tests for the above requirements are specified in clause 8 of this Recommendation | International Standard.

NOTE – There is no requirement in this Recommendation | International Standard that any encoder which embodies the encoding process specified in Annexes A, B and D shall be able to operate for all ranges of the parameters which are allowed. An encoder is only required to meet the applicable conformance tests specified in clause 8, and to generate the compressed image data format according to Annex C for those parameter values which it does use.

7 Decoder requirements

A decoding process converts compressed image data to reconstructed image data. Since the decoding process is uniquely defined by the encoding process, there is no separate normative definition of the decoding process.

NOTE – The decoding process is not specified for non-compliant compressed image data.

A subsequent sample mapping procedure uses the value of each sample output by the decoding process as an index to map each sample value to an output sample value using the mapping tables specified for that sample component in Annex C. If no table is specified for that sample component, then the output sample value is identical to the sample value output by the decoding process. In this case, an inverse point transform may also be applied (see 4.3.2), thus completing a sample transformation process.

A decoder is an embodiment of the decoding process implicitly specified by the encoding process as specified in Annexes A, B and D, followed by the embodiment of the sample transformation process defined above. In order to conform to this Recommendation | International Standard, a decoder shall satisfy all three of the following requirements.

A decoder shall:

- a) convert to reconstructed image data any compressed image data with parameters within the range supported by the application, and which comply with the interchange format syntax specified in Annex C. In the reconstructed image data output by the embodiment of the decoding process (before sample transformation), the value of each sample shall be identical to the reconstructed value defined in the encoding process specified in Annex A;
- b) accept and properly store any table-specification data which conform to the abbreviated format for table-specification data syntax specified in Annex C;
- c) convert to reconstructed image data any compressed image data with parameters within the range supported by the application, and which conforms to the abbreviated format for compressed image data syntax specified in Annex C, provided that the table specification data required for sample mapping has previously been installed in the decoder.

For the decoding process implicitly specified by the encoding process as specified in Annexes A, B and D, the conformance tests for the above requirements are specified in clause 8 of this Recommendation | International Standard.

NOTE – There is no requirement in this Recommendation | International Standard that any decoder which embodies the decoding process implicitly specified in Annexes A, B and D and the sample transformation process shall be able to operate for all ranges of the parameters which are allowed. A decoder is only required to meet the applicable conformance tests specified in clause 8, and to decode the compressed image data format specified in Annex C for those parameter values which it does use.

8 Conformance testing

8.1 Purpose

The conformance tests specified in this Recommendation | International Standard are intended to increase the likelihood of compressed image data interchange by specifying a range of tests for both encoders and decoders. The tests are not exhaustive tests of the respective functionality, and hence do not guarantee complete interoperability between independently implemented encoders and decoders. The main purpose of these conformance tests is to verify the validity of encoding and decoding process implementations, and the corresponding compressed image data. It is not an objective of these tests to carry out extensive verification of the interchange format or marker segment syntax. The marker segment syntax follows closely the interchange formats specified in Annex B of CCITT Rec. T.81 | ISO/IEC 10918-1, and testing procedures similar to those specified in ITU-T Rec. T.83 | ISO/IEC 10918-2 can be used for the purpose of verifying interchange format and marker segment syntax.

The tests are based on a set of test images which are incorporated into this specification in digital form. In addition, a reference encoder and decoder supporting the conformance tests specified in this Recommendation | International Standard (for the IBM PC) are incorporated into this specification in digital form.

8.2 Encoder conformance tests

Encoders are tested by encoding a source test image (see Annex E) using the encoder under test, and then decoding the compressed image data thus produced using a reference decoder. The image reconstructed by the reference decoder shall match the source image exactly in the case of lossless coding (parameter **NEAR** = 0, see Annex A). In the case of near-lossless coding (parameter **NEAR** > 0), the image reconstructed by the reference decoder shall match the image reconstructed by the reference decoder when fed with the compressed test image data.

The encode/decode cycle shall be carried out for each of the tests listed in Table E.2 using the test images listed in the "Source Image" column, and using the parameters specified in the "**NEAR**", "**ILV**", "Sub-sampling", and "Other JPEG-LS parameters" columns of the table. Restart markers shall not be inserted. The encoder testing procedure is illustrated in Figure 4.

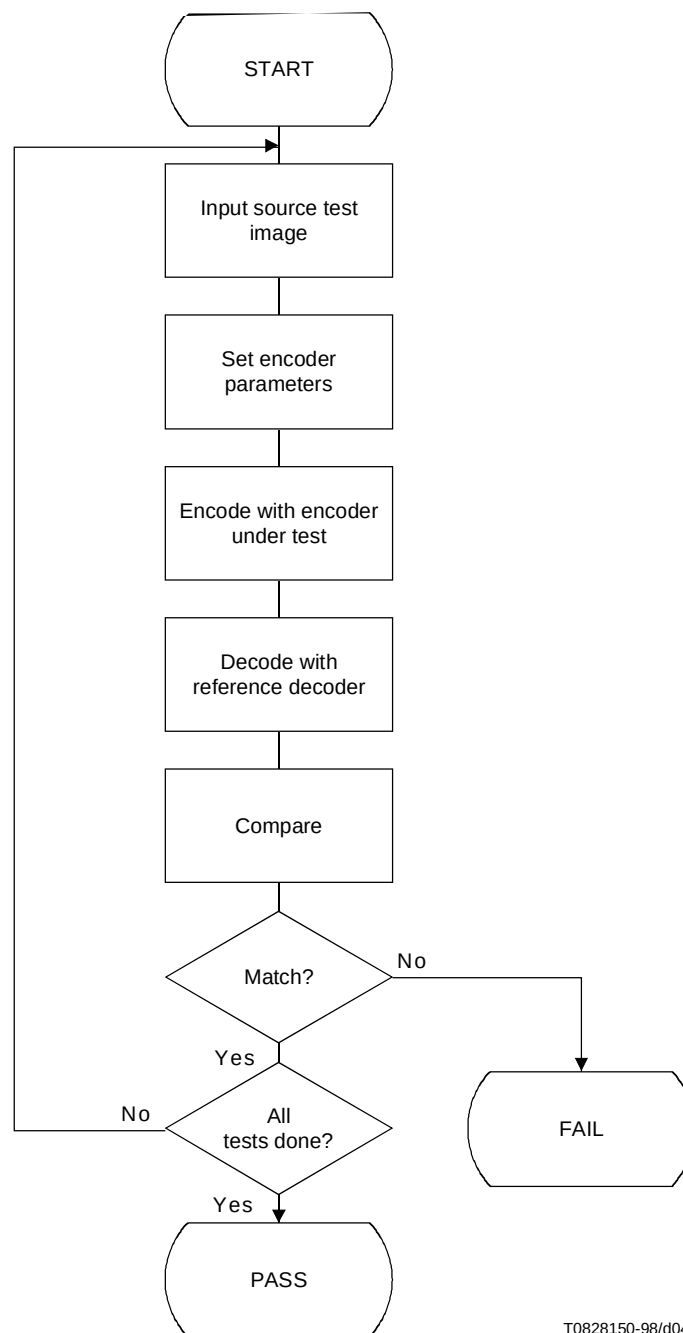
NOTE 1 – An encoder can additionally be tested without a reference decoder by comparing the compressed image data produced by the encoder, for each of the tests in Table E.2, to the compressed test image data listed in Table E.2. This comparison shall be restricted to the coded data segments only, excluding marker segments (as different marker segments may represent the same coding parameters). The coded data segments produced by the encoder shall match those contained in the compressed test image data exactly. This procedure is illustrated in Figure 5.

NOTE 2 – The tests described in this clause represent minimal encoder conformance verification. More extensive encoder testing can be achieved by encoding arbitrary source images with the encoder under test, decoding the compressed image thus produced using a reference decoder, and comparing the image data reconstructed by the reference decoder with the original source image data. The image reconstructed by the reference decoder should match the source test image exactly in the case of lossless coding (parameter **NEAR** = 0, see Annex A). In the case of near-lossless coding (parameter **NEAR** > 0), the image reconstructed by the reference decoder should match the output image data generated by the application of an encoding / decoding cycle to the same source image.

The above conformance tests shall be performed without sample mapping and with **Pt** = 0.

8.3 Decoder conformance tests

Decoders are tested by decoding compressed test image data (see Annex E) using the decoder under test and comparing the reconstructed image to the corresponding source test image. The image reconstructed by the decoder under test shall exactly match the source test image in the case of lossless coding (**NEAR** = 0). In the case of near-lossless coding (**NEAR** > 0), the image reconstructed by the decoder under test shall match the image reconstructed by the reference decoder when fed with the compressed test image data.



T0828150-98/d04

Figure 4 – Encoder testing with reference decoder

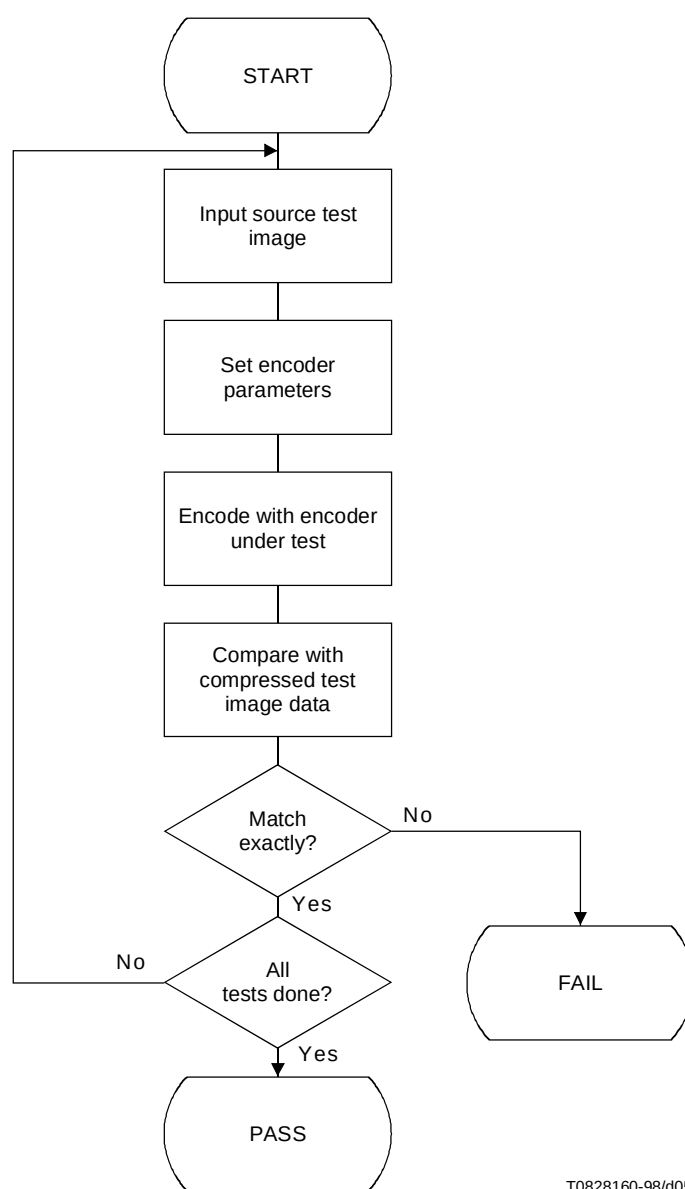
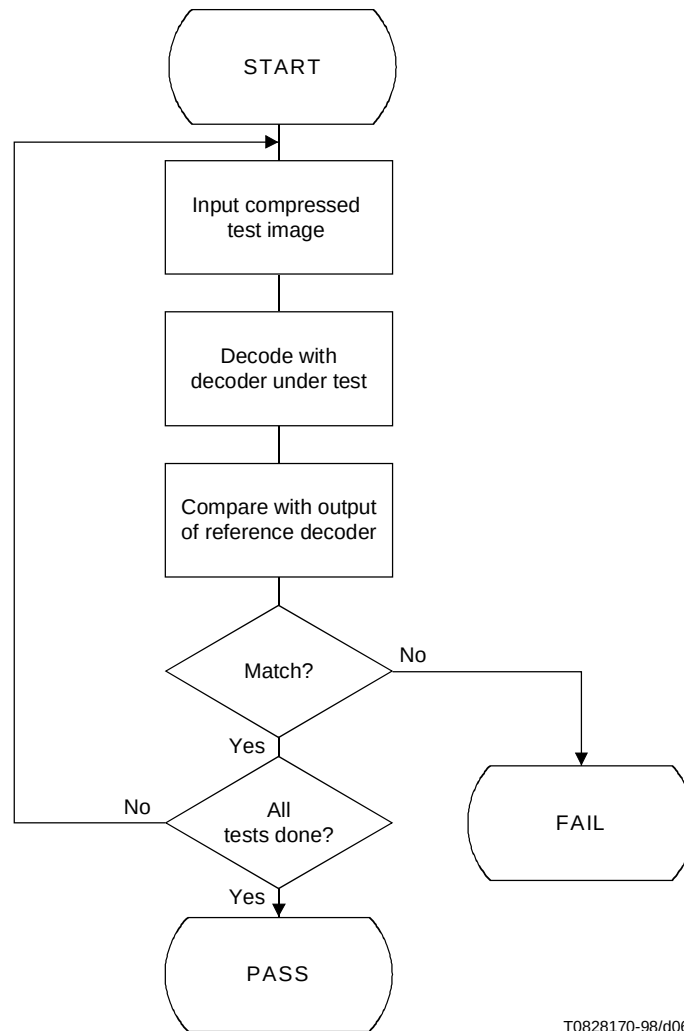


Figure 5 – Encoder testing without reference decoder

The decoding conformance tests shall be carried out for each of the tests listed in Table E.2, using as an input the compressed test image data listed in the "Compressed file name" column, with the parameters specified in the "**NEAR**", "**ILV**", "Sub-sampling", and "Other JPEG-LS parameters" columns of the table. The source test images used for the comparison are listed in the "Source image" column of Table E.2. In the case of lossless coding, no reference encoder or decoder is necessary for this decoder conformance test. The decoder testing procedure is illustrated in Figure 6.

NOTE – The tests specified represent minimal decoder conformance verification. More extensive decoder testing may be achieved by encoding arbitrary source image data with a reference encoder, decoding the compressed image data thus produced using the decoder under test and a reference decoder, and comparing the image data output by both decoders. The image data reconstructed by the decoder under test should match that reconstructed by the reference decoder for all samples.

**Figure 6 – Decoder testing procedure**

Annex A

Encoding procedures for a single component

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

This annex specifies the encoding procedures defined by this Recommendation | International Standard. Clauses A.1 to A.7 define the encoding process. Clause A.8 summarises the provisions of this annex. The encoding procedures in this annex correspond to scans of a single component. The necessary modifications for dealing with multiple-component scans are specified in Annex B. Annex G (informative) includes a general description of the encoding process.

NOTE – There is **no requirement** in this Recommendation | International Standard that any encoder or decoder shall implement the procedures in precisely the manner specified in this annex. It is necessary only that an encoder or decoder implement the **function** specified in this annex. The sole criterion for an encoder or a decoder to be considered in conformance with this Recommendation | International Standard is that it satisfy the requirements determined by the conformance tests given in clause 8.

A.1 Coding parameters and compressed image data

A number of parameters are necessary to specify the coding process in this Recommendation | International Standard. The coding of these parameters in the compressed image data, and a normative set of default values for some of these parameters are specified in Annex C. This Recommendation | International Standard does not specify how these parameters are set in the encoding process by any application using it, if non-default values are used.

The bits generated by the encoding process forming the compressed image data shall be packed into 8-bit bytes. These bits shall fill bytes in decreasing order of significance. As an example, when outputting a binary code $a_n, a_{n-1}, a_{n-2}, \dots, a_0$, where a_n is the first output bit, and a_0 is the last output bit, a_n will fill the most significant available bit position in the currently incomplete output byte, followed by a_{n-1}, a_{n-2} , and so on. When an output byte is completed, it is placed as the next byte of the encoded bit stream, and a new byte is started. An incomplete byte, just before a marker, is padded with zero-valued bits before the insertion of any marker.

NOTE 1 – This padding differs from the padding method specified in CCITT Rec. T.81 | ISO/IEC 10918-1.

Marker segments are inserted in the data stream as specified in Annex D. In order to provide for easy detection of marker segments, a single byte with the value X'FF' in a coded image data segment shall be followed with the insertion of a single bit '0'. This inserted bit shall occupy the most significant bit of the next byte. If the X'FF' byte is followed by a single bit '1', then the decoder shall treat the byte which follows as the second byte of a marker, and process it in accordance with Annex C. If a '0' bit was inserted by the encoder, the decoder shall discard the inserted bit, which does not form part of the data stream to be decoded.

NOTE 2 – This marker segment detection procedure differs from the one specified in CCITT Rec. T.81 | ISO/IEC 10918-1.

A.2 Initialisations and conventions

A.2.1 Initialisations

The context modelling procedure specified in this annex uses the causal template a, b, c and d depicted in Figure 3. When encoding the first line of a source image component, the samples at positions b, c , and d are not present, and their reconstructed values are defined to be zero. If the sample at position x is at the start or end of a line so that either a and c , or d is not present, the reconstructed value for a sample in position a or d is defined to be equal to Rb , the reconstructed value of the sample at position b , or zero for the first line in the component. The reconstructed value at a sample in position c , in turn, is copied (for lines other than the first line) from the value that was assigned to Ra when encoding the first sample in the previous line.

The following initialisations shall be performed at the start of the encoding process of a scan, as well as in other situations specified in Annex D. All variables are defined to be integers with sufficient precision to allow the execution of the required arithmetic operations without overflow or underflow, given the bounds on the parameters indicated in Annex C:

- 1) Compute the parameter **RANGE**: For lossless coding (**NEAR** = 0), **RANGE** = **MAXVAL** + 1. For near-lossless coding (**NEAR** > 0):

$$\mathbf{RANGE} = \left\lfloor \frac{\mathbf{MAXVAL} + 2 * \mathbf{NEAR}}{2 * \mathbf{NEAR} + 1} \right\rfloor + 1$$

NOTE – **MAXVAL** and **NEAR** are coding parameters whose values are either default or set by the application (see Annex C).

Compute the parameters **qbpp** = $\lceil \log \mathbf{RANGE} \rceil$, **bpp** = $\max(2, \lceil \log(\mathbf{MAXVAL}+1) \rceil)$, and **LIMIT** = $2 * (\mathbf{bpp} + \max(8, \mathbf{bpp}))$.

- 2) Initialise the variables *N*[0..366], *A*[0..366], *B*[0..364] and *C*[0..364], where the nomenclature [0..*i*] indicates that there are *i*+1 instances of the variable. The instances are indexed by [*Q*], where *Q* is an integer between 0 and *i*. The indexes [0..364] correspond to the regular mode contexts (total of 365, see A.3), whilst the indexes [365] and [366] correspond to run mode interruption contexts. For example, *C*[5] corresponds to the variable *C* in the regular mode context indexed by 5. Each one of the entries of *A* is initialised with the value

$$\max \left(2, \left\lfloor \frac{\mathbf{RANGE} + 2^5}{2^6} \right\rfloor \right)$$

those of *N* are initialised with the value 1, and those of *B* and *C* with the value 0.

- 3) Initialise the variables for the run mode: *RUNindex*=0 and *J*[0..31] = {0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 9, 10, 11, 12, 13, 14, 15}.
- 4) Initialise the two run interruption variables *Nn*[365] and *Nn*[366] to 0.

A.2.2 Conventions for code segments

In the remaining clauses of this annex, various procedures of the encoding process are specified in software code segments, written in the C programming language, as specified in ISO/IEC 9899. The syntax and semantics of C shall be assumed in all code segments contained in this annex.

All variables used in the code segments are assumed to be integer, and to have sufficient precision to allow the execution of the required arithmetic operations without overflow or underflow, given the bounds on the parameters indicated in Annex C. When division and right shift operations are indicated, all variables used are non-negative integers so that the exact computation of quotients, remainders and shifted quantities is unambiguously specified. The code segments are used to specify parts of the encoding process, and do not constitute, by themselves or in any aggregation, a full implementation of the process.

In addition to the variables and parameters specified in 3.1 for the encoding and decoding processes, the following auxiliary labels, global variables, and functions are used in the software code segments:

abs(<i>i</i>)	Function: returns the absolute value of <i>i</i> in accordance with the definition in 3.1.3
RunModeProcessing	Label: indicates the beginning of the run mode process as defined in A.7
RegularModeProcessing	Label: indicates the gradient quantization step as defined in A.3.3
max(<i>i</i>, <i>j</i>)	Function: returns the maximum of <i>i</i> and <i>j</i> in accordance with the definition in 3.1.47
min(<i>i</i>, <i>j</i>)	Function: returns the minimum of <i>i</i> and <i>j</i> in accordance with the definition in 3.1.48