INTERNATIONAL STANDARD

ISO/IEC 14495-2

First edition
2002-03-15

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

Part 2:
Extensions

Technologies de l'information — Compression sans perte et quasi sans perte d'images fixes à modelé continu —

Partie 2: Extensions

Reference number
ISO/IEC 14495-2:2002(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 copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Contents

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.

International Standard ISO/IEC 14495-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*, in collaboration with ITU-T, but is not published as common text at this time.

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 I form a normative part of this part of ISO/IEC 14495. Annexes J and K are for information only.

INFORMATION TECHNOLOGY — LOSSLESS AND NEAR-LOSSLESS COMPRESSION OF CONTINUOUS-TONE STILL IMAGES — PART 2: EXTENSIONS

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 (including bi-level), gray-scale, or colour digital still images.

This Recommendation | International Standard

– specifies extensions (including arithmetic coding, extension of near lossless coding, extension of prediction and extension of Golomb coding) to processes for converting source image data to compressed image data
– specifies extensions to processes for converting compressed image data to reconstructed image data including an extension for sample tranformation for inverse colour transforms
– 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 International 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 editions of the Recommendations and International Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The ITU-T Telecommunication Standardization Bureau (TSB) 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.87 (2000) | ISO/IEC 14495-1:2000, *Information technology – Lossless and near-lossless compression of continuous-tone still images: Baseline.*

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 9899:1999, *Programming languages - C.*

3. Definitions, abbreviations, symbols and conventions

3.1 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply in addition to the definitions used in ITU-T Rec. T.87 | ISO/IEC 14495-1:2000.

3.1.1 arithmetic encoder: An embodiment of an arithmetic encoding procedure.

3.1.2 arithmetic encoding: A procedure which encodes a sample as a binary representation of the sequence of previously encoded samples by means of a recursive subdivision of a unit interval.

3.1.3 arithmetic decoder: An embodiment of an arithmetic decoding procedure.

3.1.4 arithmetic decoding: A procedure which recovers source data from an encoded bit stream produced by an arithmetic encoder.

3.1.5 binary context: Context used to determine the binary arithmetic coding of the present binary decision

3.1.6 binary decision: Choice between two alternatives.

3.1.7 colour transform: A procedure for sample transformation for inverse colour transform.

3.1.8 sign flipping: The procedure which reverses the sign of a prediction error according to accumulated prediction errors.

3.1.9 symbol packing: A procedure which may be applied to source images in which sample values are sparsely distributed.

3.1.10 visual quantization: An extended function of near-lossless coding which enables to change the difference bound according to the context.

3.2 Abbreviations

In additions to the abbreviations used in ITU-T Rec. T.87 | ISO/IEC 14495-1:2000, the abbreviations used in this Recommendation | International Standard are listed below.

FLC: Fixed length code.

LPS: Less probable symbol

MPS: More probable symbol

3.3 Symbols

In addition to the symbols used in ITU-T Rec. T.87 | ISO/IEC 14495-1:2000, the 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.

Areg current numerical-line interval being renormalized

ArithmeticEncode() a function in the C programming language

Av[0..30] 31 constants corresponding to LPS probability estimate

Avd auxiliary variable storing modified **Av**

BASIC_T1, BASIC_T2, BASIC_T3, BASIC_T4 basic default threshold values

Bin binary decision

Buf[0..1] bytes stored to avoid carry-over propagation to the encoded bit stream

Creg value of code register storing the trailing bits of the encoded bit stream

ENT indication of the coding process used for the scan

Flag[0..**MAXVAL**] **MAXVAL**+1 flags which indicate if corresponding sample values already occurred

GetBinaryContext() a function in the C programming language

GetByte() a function in the C programming language

GetGolombk() a function in the C programming language

Hd auxiliary variable storing an integer value corresponding to a half of the full range but shifted according to the size of the current interval

LPScnt[0..**MAXS**] accumulated occurrence count of the LPS (less probable symbol) at each binary context

MAXcnt threshold value at which *MLcnt* and *LPScnt* are halved

MAXS maximum index of binary contexts

MLcnt[0..**MAXS**] accumulated occurrence count of each binary context.

MPSvalue[0..**MAXS**] sense of the MPS (more probable symbol) at each binary context

nearq context-dependent difference bound for near-lossless coding using visual quantization.

NEARRUN difference bound for near-lossless coding in run mode.

NMCU number of MCU's

Prob LPS probability estimated from *MLcnt* and *LPScnt*

Qx the (quantized) value of a sample to be encoded with fixed length code

S *index for binary contexts*

$\mathbf{SOF}_{57}$ JPEG-LS frame marker for this extension

SPf[0..**RANGE**] **RANGE**+1 flags indicating if corresponding mapped error values already occurred

SPm[0..**RANGE**] mapping table of *MErrval* or *EMErrval* for symbol packing

SPt the smallest positive integer greater than all mapped error values that occurred in the scan up to this point.

SPx number of the different mapped error values that already occurred.

T1, T2, T3 thresholds for local gradients

T4 threshold for an additional local gradient

TEMErrval auxiliary variable storing *EMErrval.*

Th[0..29] threshold to determine suitable value of **Av**

TMErrval auxiliary variable storing *MErrval*

TQ visual quantization threshold

wct number of bits by which *Areg* is shifted

Zerograd flag indicating local gradients are all zero.

4. General

The purpose of this clause is to give an overview of this Recommendation | International Standard.

This Recommendation | International Standard defines extensions to the elements specified in ITU-T Rec. T.87 | ISO/IEC 14495-1:2000. Extensions which pertain to encoding or decoding are defined as procedures which may be used in combination with the encoding and decoding processes of ITU-T Rec. T.87 | ISO/IEC 14495-1:2000. This Recommendation | International Standard also defines extensions to the compressed data formats, i.e., interchange. Each encoding or decoding extension shall only be used in combination with particular coding processes and only in accordance with the requirements set forth herein. These extensions are backward compatible in the sense that decoders which implement these extensions will also support configuration subsets that are currently defined by ITU-T Rec. T.87 | ISO/IEC 14495-1:2000.

4.1 Extensions specified by this Recommendation | International Standard

The following extensions are specified:

— An extension which provides for arithmetic coding. This extension will provide higher compression ratio, especially with high-skew images.

— An extension which provides for variable near-lossless coding. This extension will provide a wider variety of possible nearly lossless reconstructions of a source image than ITU-T Rec. T.87 | ISO/IEC 14495-1:2000. There are two types of variable near-lossless coding, depending on whether the difference bound is changed:

  a) according to its context; or

  b) in vertical direction.

— An extension which provides for a modified prediction procedure in source images in which sample values are sparsely distributed.

— An extension which provides for a modified Golomb coding procedure. This modification avoids possible expansion of compressed image data, and improves coding efficiency by eliminating code words that are not used.

— An extension which provides for fixed length coding.

— An extension which provides for a modified sample transformation process. This extension can be used to define inverse colour transforms in order to achieve greater efficiency by compressing a source image in a different colour representation.

The following sections describe these extensions in greater detail.

4.1.1 Encoding with arithmetic coding

In JPEG-LS baseline, specified in ITU-T Rec. T.87 | ISO/IEC 14495-1:2000, simple but efficient coding is achieved by the combination of Golomb coding (regular mode) and the run mode. However, for some very high-skewed image data such as computer generated images, the compression efficiency is affected by the use of symbol-by-symbol coding in contexts that present highly skewed distributions. Therefore, coding procedures based on arithmetic coding are specified in this Recommendation | International Standard as an extended function, which enables alphabet extension for every context (rather than only in run mode) and provides higher compression performance with a moderate increase of the coder complexity.

The arithmetic coder adopted in this Recommendation | International Standard is characterized by its fast multiplication-free arithmetic operation and radix-255 representation. The details are described in Annex **A** and **B**.

4.1.2 Extension of Near-lossless coding

The extension of the near-lossless coding capabilities of ITU-T Rec. T.87 | ISO/IEC 14495-1:2000 is to allow the **NEAR** parameter to vary during the process of encoding a source image. There are two types of variable near-lossless coding, serving two different purposes.

4.1.2.1 Visual quantization

Visual quantization takes into account the human visual system by primarily performing the quantization in high activity regions of the image where the activity may mask for the quantization noise. Therefore, by extending the near-lossless coding capabilities of ITU-T Rec. T.87 | ISO/IEC 14495-1:2000 so that the **NEAR** parameter can change according to its context, it becomes possible to provide reconstructed images whose distance from the source image is between those obtained with compression schemes using **NEAR**=n and **NEAR**=n+1, where n denotes a non-negative integer.

4.1.2.2 Re-specification of NEAR value

This type of variable near-lossless coding can vary the **NEAR** parameter according to the vertical direction. The main purpose of this extension of the near-lossless coding capabilities of ITU-T Rec. T.87 | ISO/IEC 14495-1:2000 is to provide a mechanism by which an encoder can change the value of **NEAR** according to the observed compressibility of the source image, which is useful to control the total length of compressed image data within some specified amount. By this extension, an encoder can compress a source image to less than a pre-specified size with a single sequential pass over the image. The capability is valuable to applications which utilize a fixed-size compressed image memory.

4.1.3 Extension of prediction

The prediction and error coding procedure specified in ITU-T Rec. T.87 | ISO/IEC 14495-1:2000 is not suitable for images with sparse histograms, such as limited color images or fewer-bit images expressed by byte form. These images contain only a subset of the possible sample values in each component, and the edge-detecting predictor specified in Figure **A.5** of ITU-T Rec. T.87 | ISO/IEC 14495-1:2000 would tend to concentrate the value of the prediction errors into a reduced set. However, the prediction correction procedure specified in Figure **A.6** of ITU-T Rec. T.87 | ISO/IEC 14495-1:2000 tends to spread these values over the entire range. In addition, the Golomb coding procedure would assign short code words to small prediction errors, even those that do not occur in the image component.

The goal of this extension is to modify the prediction procedure in order to alleviate this unwanted effect by checking the occurrence of the corrected predicted value *Px* in the past. This extension also provides a modified coding procedure to improve the coding efficiency for these images.

4.1.4 Extension of Golomb coding

In addition to the specification of Golomb coding defined in ITU-T Rec. T.87 | ISO/IEC 14495-1:2000, two modifications are incorporated in this Recommendation | International Standard as follows:

4.1.4.1 Golomb code completion

More effective usage of Golomb code words is specified in this Recommendation | International Standard, in which the final bit "1" in the longest possible unary representation used in a Golomb code, which is redundant, shall be omitted. This procedure improves the coding efficiency especially in cases in which this Recommendation | International Standard is also applied to bi-level images.

4.1.4.2 Omission of run interruption sample coding

In cases in which this Recommendation | International Standard is also applied to bi-level images and the mode is not sample interleaved, the encoding of the run interruption sample is superfluous and shall be omitted.

4.1.5 Fixed length coding

In this Recommendation | International Standard, a procedure to avoid situations in which the compressed image data is larger than the source image data is incorporated. The values of image samples, or the quantized values in near-lossless coding, are encoded with a fixed length code. The region in a scan to be encoded with a fixed length code is specified by appending a marker indicating the beginning of the fixed length coding and the end of the fixed length coding after an appropriate MCU.

4.1.6 Sample transformation for inverse colour transforms

In this Recommendation | International Standard, a procedure for sample transformation is provided, in addition to the ones defined in ITU-T Rec. T.87 | ISO/IEC 14495-1:2000. This procedure uses corresponding values of decoded samples in the various components, to reconstruct the source image data, which is of the same precision as the encoded data. The goal of this extension is to facilitate the use of this Recommendation | International Standard in conjunction with colour transforms to improve coding efficiency.

4.2 Descriptions of extended functions

The coding procedure specified in Annex **A** of ITU-T Rec. T.87 | ISO/IEC 14495-1:2000 is referred to as baseline coding process. The newly introduced coding procedure, modified from the baseline coding process, is referred to as arithmetic-based process. The context modelling for the arithmetic-based coding process is described in Annex **A**, and the arithmetic coding procedure of a binary symbol for the given context is described in Annex **B**. The functions outlined in **4.1.2**, **4.1.3**, and **4.1.6**, can be used on either arithmetic-based coding process or baseline coding process. The functions outlined in **4.1.4** and **4.1.5.** can be used only in baseline coding process. All the extended functions are optional and the combinations of the extended functions are arbitrary under this general rule.

The use of the extended functions outlined in **4.1.2** and **4.1.3** in conjunction with the arithmetic-based coding process is also described in Annex **A**. The use of the extended functions outlined in **4.1.2** and **4.1.3** in conjunction with the baseline coding process is described in Annex **D** by referring to the differences with respect to the coding process of the non-extended functions described in ITU-T Rec. T.87 | ISO/IEC 14495-1:2000.The functions outlined in **4.1.4** are also described in Annex **D**.

The extended functions outlined in **4.1.5** are described in Annex **E**. The extended functions outlined in **4.1.6** on both arithmetic-based and baseline coding are described in Annex **F**.

The contents of the Annexes for extended functions are summarized in Table **1**.

Table 1 – Combination of extended functions and corresponding Annex

	Extension of near-lossless coding	Extension of prediction	Arithmetic coding procedure	Extension of Golomb coding	Fixed length coding	Colour transform
Baseline coding process	Annex **D**	Annex **D**	---	Annex **D**	Annex **E**	Annex **F**
Arithmetic-based coding process	Annex **A**	Annex **A**	Annex **B**	---	---	

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 process and extensions selected, as defined in Annex **C** of ITU-T T. Rec. 87 | ISO/IEC 14495-1:2000 and Annex **G** of this Recommendation | International Standard.

6. Encoder requirements

An encoding process converts source image data to compressed image data. ITU-T Rec. T.87 | ISO/IEC 14495-1:2000 specifies baseline coding processes. This Recommendation | International Standard defines arithmetic-based coding process and encoding extensions which may be used in combination with baseline coding process or arithmetic-based coding process.

An extended encoder is an embodiment of one (or more) of the encoding processes specified in this Recommendation | International Standard or ITU-T Rec. T.87 | ISO/IEC 14495-1:2000 used in combination with one or more of the encoding extensions specified herein. In order to comply with this Recommendation | International Standard, an extended encoder shall satisfy at least one of the following two requirements.

An extended encoder shall

a) convert source image data to compressed image data which conform to the interchange format syntax specified in Annex **G**;

b) convert source image data to compressed image data which comply with the abbreviated format for compressed image data syntax specified in Annex **G**.

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 one of the encoding processes and extensions 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, and to generate the compressed data format according to Annex **G** 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 processes. The values of samples output by the decoding process are used as vector components in an inverse colour transform. The inverse colour transform is specified in Annex **F**. If no transform is specified for a sample component, then the colour-transformed 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** of ITU-T Rec. T.87 | ISO/IEC 14495-1:2000). A subsequent sample mapping procedure uses the value of each sample output by the inverse colour transform procedure to map each sample value to sample-mapped value using the mapping tables specified for that sample component in Annex **C** of ITU-T Rec. T.87 | ISO/IEC 14495-1:2000. Again, if no table is specified for that sample component, then the sample-mapped value is identical to the colour-transformed sample value (after possible inverse point transform).

A decoder is an embodiment of the decoding process implicitly specified by the encoding process as specified in ITU-T Rec. T.87 | ISO/IEC 14495-1:2000 and this Recommendation | International Standard, followed by the embodiment of the sample transformation process defined above. In order to comply with this Recommendation | International Standard, an extended decoder shall satisfy all three of the following requirements.

An extended 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 **G**. In the reconstructed image data output by the embodiment of the decoding process (before sample transform), the values of each sample shall be identical to the reconstructed values defined in the encoding process;

b) Accept and properly store any table-specification data which conform to the abbreviation format for table-specification data syntax specified in Annex **C** of ITU-T Rec. T.87 | ISO/IEC 14495-1:2000;

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 **G**, provided that the table specification data required for sample mapping has previously been installed in the decoder.

NOTE – There is no requirement in this Recommendation | International Standard that any decoder which embodies one of the decoding processes and extensions 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, and to decode the compressed image data format specified in Annex **G** for those parameter values which it does use.

8. Conformance testing for extensions

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 ITU-T T.81 | ISO/IEC 10918-1, and testing procedures similar to those specified in ITU-T 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.

8.2 Encoder conformance tests

Encoders are tested by encoding a source test image (see Annex **I**) using the encoder under test, and then comparing the compressed image data produced by the encoder to the compressed image data listed in Table **I.2**. The coded data segments of the compressed image data shall match those of the compressed image data in Table **I.2**.

The encoding shall be carried out for each of the tests listed in Table **I.2** using the test images listed in the “Source Image” column, and using the parameters specified in the table. Restart markers shall not be inserted. The encoder testing procedure is illustrated in Figure **1**.

NOTE – This testing is restricted to conformance of the coded data segments only, excluding marker segments (as different marker segments may represent the same coding parameters).

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

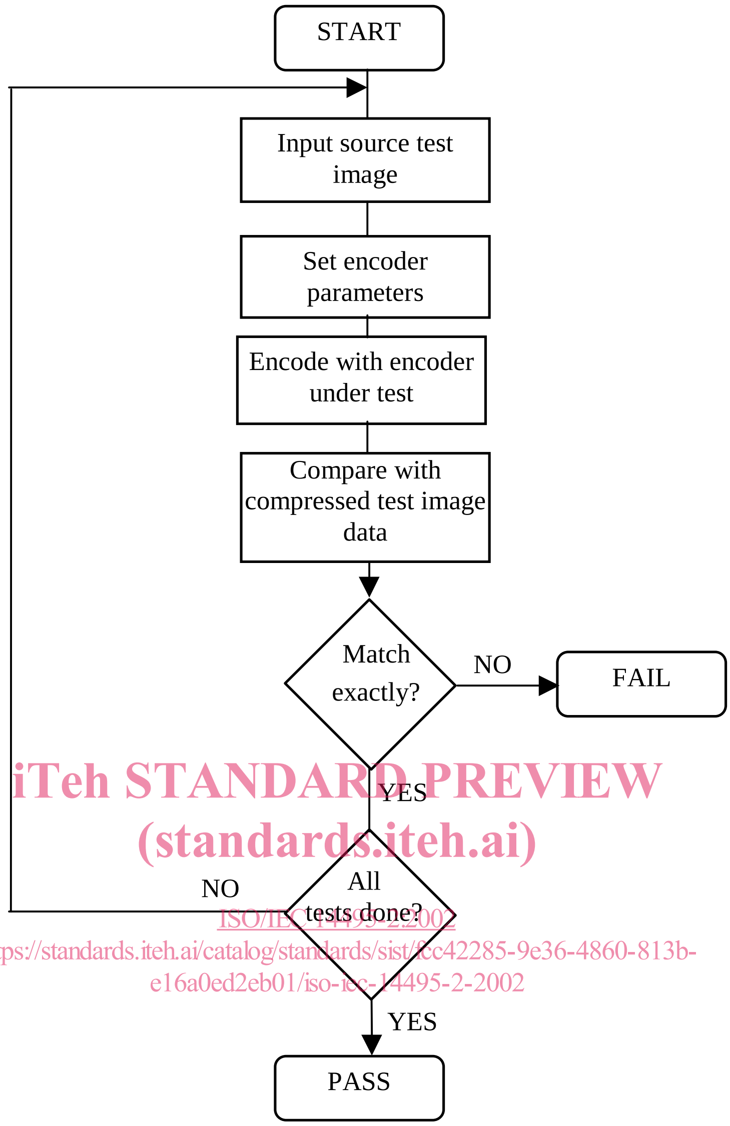

Figure 1 - Encoder testing procedure

8.3 Decoder conformance tests

Decoders are tested by decoding compressed test image data (see Annex **I**) 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 the source image data with **NEAR** for all samples.

The decoding conformance tests shall be carried out for each of the tests listed in Table **I.2**, using as an input the compressed test image data listed in the “Compressed file name” column, with the parameters specified in the table. The source test images used for the comparison are listed in the “Source image” column of Table **I.2**. The decoder testing procedure is illustrated in Figure **2**.

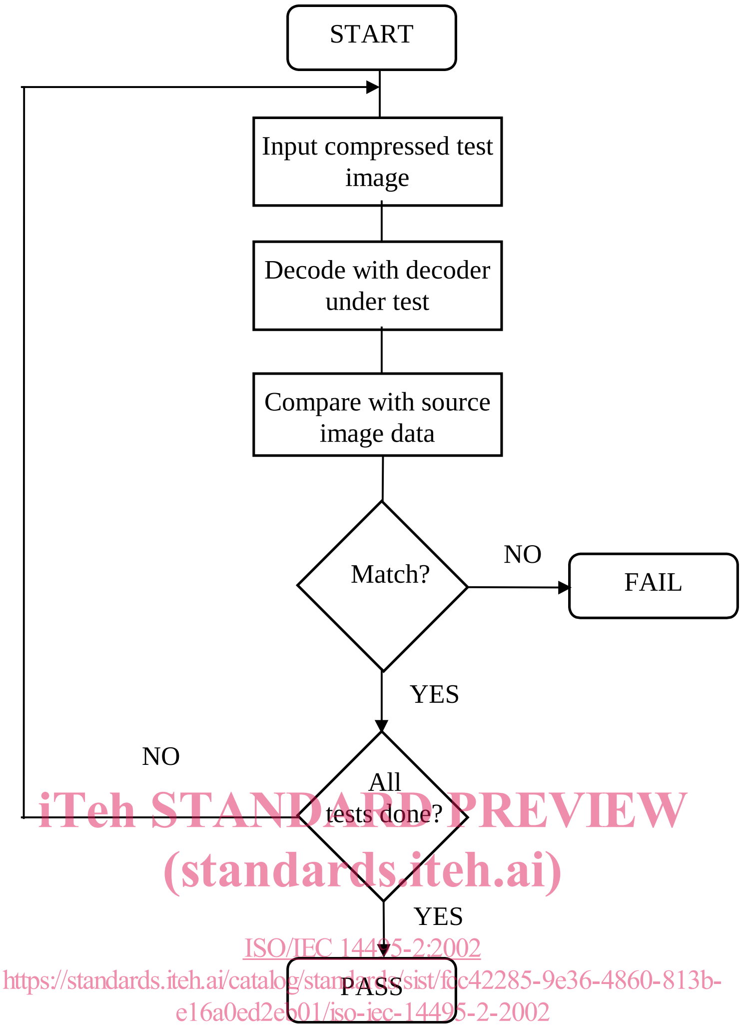

Figure 2 - Decoder testing procedure

Annex A

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

Encoding procedures with arithmetic coding for a single component

This Annex specifies the encoding procedures using arithmetic coding. The encoding procedures using arithmetic coding (arithmetic-based coding process) is similar to those using Golomb coding (baseline coding process), which is specified in Annex **A** of ITU-T Rec. T.87 | ISO/IEC 14495-1:2000. However, since there are many differences in detail, this independent Annex is provided for its description. The main differences are as follows:

- Binary arithmetic coding is used in stead of Golomb coding.
- No classification for regular mode and run mode.
- Binarization of mapped-error and context modeling for binary arithmetic coding are performed.
- The number of samples used in context modeling is increased from four to five.

This Annex presumes a single component. The necessary modifications for dealing with multiple-component scans are specified in Annex **C**.

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 arithmetic-based 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 **G**. 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},....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.

Marker segments are inserted in the data stream as specified in Annex **G**. In the coded data segment, a bit '0' is inserted after the X'FF' byte like the procedures specified in ITU-T Rec. T.87 | ISO/IEC 14495-1:2000 in the baseline coding process, but in the arithmetic-based coding process, which is specified in Annex **B**, no such treatment is necessary because every X'FF' byte, if happened, shall be always followed by X'00' byte in the data stream and will not be mistaken as marker code.

A.2 Initialisations and conventions

A.2.1 Initialisations

The differences from the initialisations specified in **A.2.1** of ITU-T Rec. T.87 | ISO/IEC 14495-1:2000 are as follows :

- Initialisation of reconstruction values outside the border of an image
- Initialisation of additional counters of *A*, *B*, *C* and *N*, which are caused by the increase of samples used for context modeling
- Initialisation of variables for arithmetic coding.
- Elimination of initialisation of variables for the run mode.

The context modeling procedure specified in this Annex uses the causal template *a*, *b*, *c*, *d* and *e* depicted in Figure **A.1**. 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*, *c* and *e*, or *d* is not present, the reconstructed value for a sample (samples) in position *a* and *e*, 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. If the sample at position *x* is at the second column of a line so that *e* is not present, the reconstructed value for a sample in position *e* is copied from the value that was assigned to *Ra* when encoding the previous sample.

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 **H**. 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 **G**:

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** of of ITU-T Rec. T.87 | ISO/IEC 14495-1:2000).

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

2. Initialise the variables *N*[0..1091], *A*[0..1091], *B*[0..1091] and *C*[0..1091], 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. For example, *C*[5] corresponds to the variable C 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 arithmetic coder, *LPScnt*[0..**MAXS**], *MLcnt*[0..**MAXS**] and *MPSvalue*[0..**MAXS**], where **MAXS** is the maximum index of binary contexts. At the nodes for the unary representation of the Golomb code tree described in **A.5.2**, counters are initialised as *LPScnt*[*S*]=2 and *MLcnt*[*S*]=4, where *S* is an index of the binary context. At the nodes in sub-trees, counters are initialised as *LPScnt*[*S*]=4 and *MLcnt*[*S*]=8. The sense of MPS *MPSvalue*[*S*] is initialised as *MPSvalue*[*S*]=0 for all the binary contexts.

4. Initialise the error tolerance for near-lossless coding as *nearq*=**NEAR** (in lossless, *nearq*=0). If an extended function of near-lossless coding is indicated, initialise the visual quantization threshold **TQ** as is specified in the LSE marker segment..

Note - In the extended near-lossless coding specified in this Recommendation | International Standard, the error tolerance *nearq* is variable, although in coding and near-lossless coding without the extension and lossless coding, it is fixed to **NEAR** during the encoding of a scan.

5. If the extension of prediction is indicated, initialise the variable *Flag*[1..**MAXVAL**]. *Flag*[0] is initialised with the value 1, and *Flag*[1..**MAXVAL**] with the value 0.

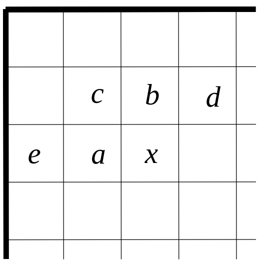

Figure A.1 - Causal template used for encoding with arithmetic coding.

A.2.2 Conventions for figures

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 9899:1990. 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** of ITU-T Rec. T.87 | ISO/IEC 14495-1:2000 and Annex **G** of this Recommendation | International Standard. When division and right shift operations are indicated, all variables used are non-negative integers so that the exact computation of quotients,