INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

T.84

(07/96)

TERMINALS FOR TELEMATIC SERVICES

INFORMATION TECHNOLOGY – DIGITAL COMPRESSION AND CODING OF CONTINUOUS-TONE STILL IMAGES: EXTENSIONS

ITU-T Recommendation T.84

(Previously "CCITT Recommendation")

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. Some 179 member countries, 84 telecom operating entities, 145 scientific and industrial organizations and 38 international organizations participate in ITU-T which is the body which sets world telecommunications standards (Recommendations).

The approval of Recommendations by the Members of ITU-T is covered by the procedure laid down in WTSC Resolution No. 1 (Helsinki, 1993). In addition, the World Telecommunication Standardization Conference (WTSC), which meets every four years, approves Recommendations submitted to it and establishes the study programme for the following period.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC. The text of ITU-T Recommendation T.84 was approved on 3rd of July 1996. The identical text is also published as ISO/IEC International Standard 10918-3.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

CONTENTS

Page

Summary

This Recommendation | International Standard, Digital compression and coding of continuous-tone still images, is published as three parts:

- CCITT Rec. T.81 | ISO/IEC 10918-1: Requirements and guidelines;
- ITU-T Rec. T.83 | ISO/IEC 10918-2: Compliance testing;
- ITU-T Rec. T.84 | ISO/IEC 10918-3: Extensions.

This Recommendation | International Standard sets out requirements and guidelines for encoding and decoding extensions to the processes defined by CCITT Rec. T.81 | ISO/IEC 10918-1, and for the coded representation of compressed image data of these extensions. This Recommendation | International Standard also defines tests for determining whether implementations comply with the requirements for the various encoding and decoding extensions.

This Recommendation | International Standard:

- defines extensions [including variable quantization, selective refinement, composite tiling, and a Still Picture Interchange File Format (SPIFF)] to processes for converting source image data to compressed image data;
- defines extensions to processes for converting compressed image data to reconstructed image data;
- defines coded representations for compressed image data;
- gives guidance and examples on how to implement these extensions in practice;
- describes compliance tests for these extensions.

Patents

ISO/IEC JTC1/SC29/WG1 and ITU-T TSB are not aware of any patent restrictions applying to the extensions in this Recommendation | International Standard.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – DIGITAL COMPRESSION AND CODING OF CONTINUOUS-TONE STILL IMAGES: EXTENSIONS

1 Scope

This Recommendation | International Standard is applicable to continuous-tone – grayscale or colour – digital still image data. It is applicable to a wide range of applications which require use of compressed images.

This Recommendation | International Standard:

- defines extensions [including variable quantization, selective refinement, tiling, and a Still Picture Interchange File Format (SPIFF)] to processes for converting source image data to compressed image data;
- defines extensions to processes for converting compressed image data to reconstructed image data;
- defines coded representations for compressed image data;
- gives guidance and examples on how to implement these extensions in practice;
- describes compliance tests for these extensions.

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 (TSB) of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation H.262 (1995) | ISO/IEC 13818-2:1995, *Information technology – Generic coding of moving pictures and associated audio information: Video*.
- 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.82 (1993) | ISO/IEC 11544:1993, *Information technology – Coded representation of picture and audio information – Progressive bi-level image compression*.
- ITU-T Recommendation T.83 (1994) | ISO/IEC 10918-2:1995, *Information technology – Digital compression and coding of continuous-tone still images: Compliance testing*.

2.2 Additional references

- ISO 3166:1993[1], *Codes for the representation of names of countries*.
- ISO 5807:1985, *Information processing – Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts*.
- ISO 8601:1988, *Data elements and interchange formats – Information interchange – Representation of dates and times*.

[1] Currently under revision.

- ISO 8859-1:1987, *Information processing – 8-bit single byte coded graphic character sets – Part 1: Latin alphabet No. 1*.
- ISO 8859-2:1987, *Information processing – 8-bit single-byte coded graphic character sets – Part 2: Latin alphabet No. 2*.
- ISO 8859-3:1988, *Information processing – 8-bit single-byte coded graphic character sets – Part 3: Latin alphabet No. 3*.
- ISO 8859-4:1988, *Information processing – 8-bit single-byte coded graphic character sets – Part 4: Latin alphabet No. 4*.
- ISO/IEC 8859-5:1988, *Information processing – 8-bit single-byte coded graphic character sets – Part 5: Latin/Cyrillic alphabet*.
- ISO 8859-6:1987, *Information processing – 8-bit single-byte coded graphic character sets – Part 6: Latin/Arabic alphabet*.
- ISO 8859-7:1987, *Information processing – 8-bit single-byte coded graphic character sets – Part 7: Latin/Greek alphabet*.
- ISO 8859-8:1988, *Information processing – 8-bit single-byte coded graphic character sets – Part 8: Latin/Hebrew alphabet*.
- ISO/IEC 8859-9:1989, *Information processing – 8-bit single-byte coded graphic character sets – Part 9: Latin alphabet No. 5*.
- ISO/IEC 8859-10:1992, *Information technology – 8-bit single-byte coded graphic character sets – Part 10: Latin alphabet No. 6*.
- ISO/IEC 10646-1:1993, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*.
- ISO/IEC 11172-2:1993, *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 2: Video*.
- ITU-T Recommendation T.4 (1993), *Standardization of group 3 facsimile apparatus for document transmission*.
- CCITT Recommendation T.6 (1988), *Facsimile coding schemes and coding control functions for Group 4 facsimile apparatus*.
- ITU-T Recommendation T.30 (1993), *Procedures for document facsimile transmission in the general switched telephone network*.
- ITU-T Recommendation T.42 (1994), *Continuous-tone colour representation method for facsimile*.
- CCITT Recommendation T.51 (1992), *Latin based coded character sets for telematic services*.
- ITU-T Recommendation T.85 (1995), *Application profile for Recommendation T.82 – Progressive bi-level image compression (JBIG coding scheme) for facsimile apparatus*.
- CCITT Recommendation T.503 (1991), *A document application profile for the interchange of Group 4 facsimile documents*.
- CIE 1976 (L* a* b*) space, *CIE Publication No. 15.2, Colorimetry, 2nd. Ed. (1986)*.
- Recommendation ITU-R BT.470-3 (1995), *Television systems*.
- Recommendation ITU-R BT.601-4 (1992), *Encoding parameters of digital television for studios*.
- Recommendation ITU-R BT.709-1, *Basic parameter values for the HDTV standards for the studio and for international programme exchange*.
- SMPTE 170M – 1994, *For television – Composite analog video signal – NTSC for studio applications*.
- SMPTE 240M – 1994, *For television – Signal parameters – 1125 line high definition production systems*.

3 Definitions, abbreviations, symbols and conventions

3.1 Definitions

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

3.1.1 capability indicator: A parameter of the version marker segment that indicates which of the processes of CCITT Rec. T.81 | ISO/IEC 10918-1 or the extensions of ITU-T Rec. T.84 | ISO/IEC 10918-3 are required to decode the bitstream which follows. Each bit of this parameter signals the presence of a particular capability.

3.1.2 component registration: Specification of the spatial positioning of samples within components relative to the samples of other components.

3.1.3 component selective refinement: A type of selective refinement which allows for a region of an image to contain fewer than the total number of colour components defined in the frame header.

3.1.4 composite tiling: A type of tiling which allows for overlapping tiles without restrictions on tile size and other compression parameters.

3.1.5 grid: A mathematical abstraction consisting of a two-dimensional array of elements used for composite tiling.

3.1.6 hierarchical selective refinement: A type of selective refinement which allows for a region of an image to be further refined by the next differential frame of a hierarchical sequence.

3.1.7 hierarchical sequence: A hierarchical progression of frames starting with a DHP marker segment.

3.1.8 image grid: For composite tiled images, a grid associated with the down-sampled collaged image.

3.1.9 parameters: Fixed length unsigned integers 4, 8, 16, or 32 bits in length, or variable length strings terminated by a single byte with all bits set to 0, used in the compressed data formats and the Still Picture Interchange File Format.

3.1.10 progressive selective refinement: A type of selective refinement which allows for more non-zero DCT coefficients, more bits to the DCT coefficients, or both to be added to those previously coded in a region of an image.

3.1.11 pyramidal tiling: A type of tiling which allows for multiple resolution levels. Tiles within each resolution level adhere to simple tiling definitions.

3.1.12 reference grid: For composite tiled images, a grid associated with the up-sampled collaged image.

3.1.13 resolution: Samples per unit of linear dimension.

3.1.14 selective refinement: An extension which provides for further processing of a region of an image, typically for the purpose of image quality enhancement. There are three types of selective refinement: hierarchical selective refinement, progressive selective refinement, and component selective refinement.

3.1.15 simple tiling: The definition of an image by an association of contiguous non-overlapping sub-images, each with identical parameters.

3.1.16 tile: A sub-image of any size and resolution which is part of a larger image.

3.1.17 tiled image: An image composed of rectangular sub-images. These sub-images are called tiles.

3.1.18 tile grid: For composite tiled images, a grid associated with a tile.

3.1.19 variable quantization: The process of rescaling quantization values used to quantize the AC DCT coefficients at any 8 by 8 block of an image, as described in Annex C.

3.2 Abbreviations

The abbreviations used in this Recommendation | International Standard are listed below.

3.2.1 JBIG: Joint Bi-level Image experts Group – The joint ISO/ITU committee responsible for developing standards for bi-level image coding. It also refers to the standard produced by this committee: ITU-T Rec. T.82 | ISO/IEC 11544.

3.2.2 JFIF: JPEG File Interchange Format – A basic file format specification for the delivery and interchange of JPEG compressed images.

3.2.3 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.

3.2.4 JTIP: JPEG Tiled Image Pyramid – A file organization where large images are tiled, reduced, and packed into a multi-resolution pyramidal JPEG image.

3.2.5 MPEG: Moving Pictures Expert Group – ISO-IEC committee responsible for developing standards for coding for video and audio. It also refers to the relevant standards produced by the group: MPEG1-ISO/IEC 11172 and MPEG2-ISO/IEC 13818.

3.2.6 SPIFF: Still Picture Interchange File Format – A file format defined by ITU-T Rec. T.84 | ISO/IEC 10918-3 intended for use by a wide variety of applications to exchange still pictures.

3.3 Symbols

In addition to the symbols used in CCITT Rec. T.81 | ISO/IEC 10918-1, the following symbols used in this Recommendation | International Standard are listed below:

BPS Bits per sample (SPIFF file header parameter)

C Compression type (SPIFF file header parameter)

CAP_i Capability indicator – V + 1 bytes

CHARSET Character set to be used to interpret the character data in SPIFF files

C_i Component identifier

CONTLOC File offset of contact information or zero (SPIFF directory entry parameter)

COPYRLOC File offset of copyright information or zero (SPIFF directory entry parameter)

COPYRID Copyright identifier (SPIFF directory entry parameter)

CREATLOC File offset of creator identification or zero (SPIFF directory entry parameter)

CROFFSETi Component registration vertical and horizontal offsets (SPIFF directory entry parameter)

CRvo Component registration vertical offset

CRho Component registration horizontal offset

DATE Modification date (SPIFF directory entry parameter)

DCR Define component registration marker

DESCLOC File offset of the image description or zero (SPIFF directory entry parameter)

DQS Define quantizer scale selection marker

DTI Define tiled image marker

DTT Define tile marker

DTTINDX File offset of the offset list of DTT markers (SPIFF directory entry parameter)

EDATA SPIFF directory entry data – The parameter field of an ETAG, which has a format specific to the corresponding ETAG value

ELEN SPIFF directory entry length – The length of the directory entry in bytes, minus 2

EMN SPIFF directory entry magic number – A two-byte code signalling the start of a directory entry

EOD SPIFF end of directory

EODLEN SPIFF EOD entry length – The exact length of the EOD entry in bytes

EODTAG SPIFF end of directory tag – Identifies the EOD entry

ETAG SPIFF directory entry tag

HEIGHT Image height (SPIFF file header parameter)

HLEN Header length (SPIFF file header parameter)

HRES Horizontal resolution (SPIFF file header parameter)

IDENT Additional identifier (SPIFF file header parameter)

IMGFLIP Image flip (SPIFF directory entry parameter)

IMGOR Image orientation (SPIFF directory entry parameter)

Lcr Length of parameters in DCR segment

LEVAUT Level of authenticity (SPIFF directory entry parameter)

Lqs Length of parameters in DQS segment

Lrf Length of parameters in SRF segment

Lrs	Length of parameters in SRS segment
Ltf	Length of parameters in DTT segment
Lti	Length of parameters in DTI segment
Lv	Length of parameters in VER segment
MN	Magic number – The first four bytes of a SPIFF file (SPIFF file header parameter)
NC	Number of components (SPIFF file header parameter)
NUMDTT	Number of DTT markers (SPIFF directory entry parameters)
NUMRST	Number of restart markers in a scan or zero (SPIFF directory entry parameter)
NUMSCAN	Number of scans (SPIFF directory entry parameter)
$\mathbf{O_{hf}}$	Selectively refined frame horizontal offset
$\mathbf{O_{hs}}$	Selectively refined scan horizontal offset
$\mathbf{O_{vf}}$	Selectively refined frame vertical offset
$\mathbf{O_{vs}}$	Selectively refined scan vertical offset
P	Profile identification (SPIFF file header parameter)
Q_SCALE	quantization scale
QS_CHANGE	Quantization scale change – The DC magnitude category that signals a new value of SCALE_CODE
R	Resolution units (SPIFF file header parameter)
REFNO1	Reference number one (SPIFF directory entry parameter)
REFNO2	Reference number two (SPIFF directory entry parameter)
REFNO3	Reference number three (SPIFF directory entry parameter)
REGAUT	Contact registration authority (SPIFF directory entry parameter)
REGCON	Registration country (SPIFF directory entry parameter)
REGID	Registration identifier (SPIFF directory entry parameter)
Rev	Revision number
RGhs	Reference grid width
RGvs	Reference grid height
RSTLIST	Restart list – The file offset to the start of the restart marker list for the scan or zero (SPIFF directory entry parameter)
S	Colour space in which the sample values define coordinates (SPIFF file header parameter)
SCALE_CODE	A 5-bit value which specifies the scale factor Q_SCALE
SCANEND	File offset of the first marker after the scan's compressed data (SPIFF directory entry parameter)
SCANLIST	Scan list containing one 4-word entry per scan (SPIFF directory entry parameter)
SCANSTRT	File offset to the X'FF' byte of the SOS (SPIFF directory entry parameter)
$\mathbf{S_{hs}}$	Selectively refined scan horizontal size
SRF	Selectively refined frame marker
SRS	Selectively refined scan marker
STRLOC	SPIFF file offset of a character string or zero
$\mathbf{S_{vs}}$	Selectively refined scan vertical size
Tc	Quantizer scale selector
TFho	Tile horizontal offset
TFhs	Tile horizontal scale
TFvo	Tile vertical offset
TFvs	Tile vertical scale
TIhs	Tiled image horizontal scale

TIME	Modification time (SPIFF directory entry parameter)
TITLELOC	The file offset of the image title or zero (SPIFF directory entry parameter)
TIvs	Tiled image vertical scale
TNBPS	Thumbnail bits per sample (SPIFF directory entry parameter)
TNC	Thumbnail compression type (SPIFF directory entry parameter)
TNDATA	Thumbnail data offset (SPIFF directory entry parameter)
TNHEIGHT	Thumbnail vertical size (SPIFF directory entry parameter)
TNS	Thumbnail colour space (SPIFF directory entry parameter)
TNWIDTH	Thumbnail horizontal size (SPIFF directory entry parameter)
TRANCHAR	Transfer characteristics (SPIFF directory entry parameter)
TT	Tiling type
V	Version number
VER	Version marker
VERS	SPIFF version (SPIFF file header parameter)
VERSNLOC	File offset of the image version or zero (SPIFF directory entry parameter)
VRES	Vertical resolution (SPIFF file header parameter)
WIDTH	Image width (SPIFF file header parameter)

3.4 Conventions

The flowcharts use the conventions given in ISO 5807. One of the conventions is that arrows are not needed when the flow is from left-to-right and from top-to-bottom. Arrows are sometimes used in such cases to increase clarity.

4 General

The purpose of this clause is to give an overview of this Recommendation | International Standard. Another purpose is to introduce some of the terms which are defined in clause 3. (Terms defined in clause 3 of CCITT Rec. T.81 | ISO/IEC 10918-1 and clause 3 of ITU-T Rec. T.83 | ISO/IEC 10918-2 continue to apply in this Recommendation | International Standard.)

This Recommendation | International Standard defines extensions to the elements specified in CCITT Rec. T.81 | ISO/IEC 10918-1. This Recommendation | International Standard also describes compliance testing for embodiments of these extensions. Extensions which pertain to encoding or decoding are defined as procedures which may be used in combination with the encoding and decoding processes of CCITT Rec. T.81 | ISO/IEC 10918-1; no new processes are defined. This Recommendation | International Standard also defines extensions to the compressed data formats, i.e. interchange format and the abbreviated formats. 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 CCITT Rec. T.81 | ISO/IEC 10918-1.

4.1 Extensions specified by this Recommendation | International Standard

The following extensions are specified:

- An extension which provides for variable quantization within a scan. The variable quantization extension may be used in conjunction with any of the DCT-based processes with the exception of the baseline process. The variable quantization extension provides for scaling of all quantization tables at the 8×8 block level;
- An extension which provides for selective refinement. The selective refinement extension refers to selecting rectangular regions of an image components for further refinement. There are several types of selective refinement:
  a) The first type of selective refinement, referred to as hierarchical selective refinement, allows only a region of one or more components to be further refined by the next differential frame of a hierarchical sequence.

b) The second type of selective refinement, referred to as progressive selective refinement, applies to the DCT-based progressive mode of operation. This type of selective refinement allows more non-zero DCT coefficients, more bits to the DCT coefficients, or both, to be added to a region of one or more component.

c) The third type of selective refinement, referred to as component selective refinement, is used to specify a region of an image, which contains colour components which do not exist in other regions of the image.

– An extension which provides for tiling. The tiling extension is used to associate a number of sub-images, also called tiles, in order to form a single tiled image. The three types of tiling are summarized below:

a) For simple tiling, all tiles except possibly those on the right and bottom border have the same maximum dimensions, number of components, component IDs and scaling factors. Tiles (i.e. their components' arrays) are non-overlapping and contiguous. The tiles are coded sequentially from left-to-right and top-to-bottom.

b) For pyramidal tiling, multiple resolution versions of the same image (i.e. each version has different maximum dimensions) may be stored together in the same data stream. Each version of the image (also called a "resolution level") is stored as a tiled image with simple tiling.

c) For composite tiling, there are no restrictions except that all tiles shall have the same component identifiers.

– An extension which provides for the interchange of compressed image files between application environments. This extension is referred to as the Still Picture Interchange File Format (SPIFF) extension.

– Other extensions include addition of a version number to the compressed data format and increasing the limit on the number of data units in a minimum coded unit to 20.

The following subclauses describe these extensions in greater detail.

4.1.1 Variable quantization extension

The variable quantization extension is an enhancement to the quantization procedure of DCT-based processes which provides for changes to the quantization table values within a scan at the 8×8 block level. This extension may be used in conjunction with any of the DCT-based processes with the exception of the Baseline Process.

The quantization procedure as defined by CCITT Rec. T.81 | ISO/IEC 10918-1 is the step in the encoding process where each of the 64 DCT coefficients are quantized using one of 64 corresponding values from a quantization table. CCITT Rec. T.81 | ISO/IEC 10918-1 permits quantization tables to be redefined prior to the start of a scan but does not allow quantization table values to be changed within a scan. The variable quantization extension defined by this Recommendation | International Standard provides for scaling of quantization values at the 8×8 block level.

The variable quantization extension introduces a quantizer scale factor which may be coded in the compressed data stream at the start of any 8×8 block. The quantizer scale factor is used to scale the quantization table values which correspond to the AC coefficients in the quantization procedure. All defined quantization tables are scaled by the same quantizer scale factor.

This extension provides the following capabilities:

– The ability to compress an image to less than a bounded size with a single sequential pass over the image. The capability is valuable to applications which utilize a fixed-size compressed picture memory.

– The ability to use the masking properties of the human visual system more effectively, and thereby achieve greater compression rates for the same subjective quality.

– The ability to transcode, i.e. entropy decoding followed by entropy encoding, between some coded data representations defined by this Recommendation | International Standard and those defined by some other standards, including ISO/IEC 11172-2 and ITU-T Rec. H.262 | ISO/IEC 13818-2 (MPEG).

4.1.2 Selective refinement extension

The selective refinement extension is used to select a rectangular region of one or more components of an image for further refinement. The different types of selective refinement are described below.

4.1.2.1 Hierarchical selective refinement

Hierarchical selective refinement is used in the hierarchical mode of operation for refining a rectangular region of one or more components of an image. The location of the region of each component of the image to be selectively refined is specified immediately prior to a differential frame within an hierarchical sequence. The size of the region of each component is specified in the differential frame header. The difference image data reconstructed from the differential frame is then added only to the specified region of each component. One of the primary uses for this type of selective refinement is for coding a particular region of interest with greater detail than the remainder of the image.

4.1.2.2 Progressive selective refinement

The second type of selective refinement, referred to as progressive selective refinement, is used in the DCT-based progressive mode of operation. This type of selective refinement is used for similar reasons as hierarchical selective refinement. Progressive selective refinement may be applied to the DCT-based processes which use spectral selection, successive approximation, or both procedures in combination.

When progressive selective refinement is applied to a scan which uses the spectral selection procedure, more non-zero DCT coefficients are added to a region of each component. When it is applied to a scan which uses the successive approximation procedure, more bits are added to the DCT coefficients in a region of each component. Progressive selective refinement may also be applied to scans which use both procedures. In all cases, the location and size of the region of each component to be selectively refined is specified immediately prior to the scan which is used for selective refinement.

4.1.2.3 Component selective refinement

The third type of selective refinement, referred to as component selective refinement, may be used in all modes of operation for specifying a region of an image which contains colour components that do not exist in other regions of the image. The most common use for this type of selective refinement is for representing images which are mixed grayscale and colour.

4.1.3 Tiling extension

The tiling extension is used to associate a number of sub-images, also called tiles, in order to form a single tiled image. The tiling extension is also used to represent images which have maximum dimensions larger than 65535 samples on a side. The three types of tiling are described separately below.

4.1.3.1 Simple tiling

Simple tiling can be conceptualized as simply breaking up a larger image into smaller rectangular “tiles”. Simple tiling is useful for dividing a large image into easier to manage pieces and to allow random access to part of the compressed image.

For simple tiling, the tiles have the same maximum dimensions, with the exception of the tiles on the right and bottom sides of the image when either tiled image maximum dimension is not an integer multiple of the corresponding tile maximum dimension. Tiles are contiguous, non-overlapping and are coded sequentially from left-to-right and top-to-bottom. They all must have the same component identifiers, sampling factors, and be encoded with the same coding process. All tiles, with the possible exception of the tiles on the right and bottom sides of the image, must have an integral number of MCU rows and MCU columns.

4.1.3.2 Pyramidal tiling

Pyramidal tiling offers a method of storing multiple resolution versions of an image within the same compressed data stream. Pyramidal tiling is useful for providing direct access to lower resolution versions of a larger higher resolution image (i.e. versions with smaller maximum dimensions obtained by down-sampling of the image). For example, pyramidal tiling provides the capability to view large images on a screen, as required for browsing through “thumbnail” versions of images in a database or for side-by-side image comparison.

Pyramidal tiling allows for tiles of one resolution level to overlap those of other resolution levels. The lowest resolution level must be positioned first in the data stream and be followed by levels of increasing resolution. Tiles within a single resolution layer must conform to the rules for simple tiling.

4.1.3.3 Composite tiling

For composite tiling, there are no restrictions except that all tiles shall have the same component identifiers. Composite tiling is useful for relating diverse sub-images into an image collage, i.e. a single composite image.

In order to relate different images to each other as tiles of a composite image, it is necessary to define a geometry for each tile. The tile geometry is defined with respect to the tile grid which has width and height equal to the maximum horizontal and vertical dimensions of the image. The dimensions of each of the tile's components are computed from the width and height of the tile grid and the component sampling factors.

The tile geometry also requires the registration for all components to be defined. Component registration is simply the geometric relationship of the samples within a component to the tile grid. The DCR marker segment is used to define component registration by specifying the spatial position of the sample in the upper left corner of each component with respect to the tile grid.

The composite image uses a reference grid to specify the spatial relationship of the individual tiles. The tile grid of each tile is scaled and placed at a specified location within the reference grid. A third grid, the image grid, relates the resolution of the reference grid to the resolution of the intended output device. The use of a higher resolution reference grid can reduce border effects for multiple resolution images.

4.1.4 Still Picture Interchange File Format (SPIFF) extension

The Still Picture Interchange File Format (SPIFF) extension provides for the interchange of image files between application environments. SPIFF is a generic file format intended for interchange only and does not include many of the features found in application-specific formats.

SPIFF is based on the interchange format for compressed image data specified in CCITT Rec. T.81 | ISO/IEC 10918-1 and this Recommendation | International Standard. The interchange format omits certain parameters such as aspect ratio and colour space designation because they are not strictly required for decoding the image component values. SPIFF is a complete coded image representation, i.e. it includes all parameters necessary to reconstruct and present the decoded picture accurately on an output device.

4.1.5 Other extensions

A mandatory version number marker segment has been specified as part of the compressed data formats which utilize the extensions defined by this Recommendation | International Standard and for future extensions. The first version number marker segment has a version specifier which identifies the minimum level of functionality required to decode the entire compressed data stream (for images without composite tiling), or the first tile (for images with composite tiling). Additional version number marker segments may be inserted to identify portions of the stream which require lower levels of functionality (for images without composite tiling), or identify tiles which require other levels of functionality (for images with composite tiling). The version number marker segment also contains a capability indicator field which identifies the coding processes and extensions used in the compressed data stream.

Another extension defined by this Recommendation | International Standard increases the limit on the number of data units per Minimum Coded Unit (MCU) during interleaved encoding to 20. A data unit is a sample for lossless processes and an 8×8 block of samples for DCT-based processes.

4.2 Compliance tests for the extensions

The purpose of compliance tests is to provide designers, manufacturers, or users of a product with a set of procedures for determining whether the product meets a specified set of requirements with some confidence. In addition, the compliance tests specified herein are intended to achieve the following specific goals:

- increase the likelihood of compressed data interchange;
- decrease the likelihood that DCT-based encoders or decoders will yield reduced image quality as a result of computing the DCT or quantization procedures with insufficient accuracy;
- help implementors to meet this Recommendation | International Standard requirements for encoders and decoders as fully as possible.

For each of the processes defined by CCITT Rec. T.81 | ISO/IEC 10918-1, the compliance tests for the above requirements are specified in ITU-T Rec. T.83 | ISO/IEC 10918-2. For each of the extensions defined by this Recommendation | International Standard, the compliance tests are specified in Annex G.

4.2.1 Availability of compliance test data

Standardized compliance test data is used to perform the encoder and decoder compliance tests. There are two types of compliance test data which are used by the encoder compliance tests: source image test data and encoder reference test data. Similarly, there are two types of compliance test data which are used by the decoder compliance tests: compressed test data and decoder reference test data.

Information about compliance test data for the encoder compliance tests and the generic decoder compliance tests is available from ISO and ITU to parties who wish to determine compliance of an encoder or decoder. Information about compliance test data for application-specific compliance tests should be obtained from the standards body which maintains standards for the particular application area.

Contact points:

- International Organization for Standardization
  1, rue de Varembé
  CH-1211 Genève 20, Switzerland

- International Telecommunication Union
  Place des Nations
  CH-1211 Genève 20, Switzerland

5 Compressed data format requirements

The compressed data formats specified by CCITT Rec. T.81 ISO/IEC 10918-1 are:

a) the interchange format;

b) the abbreviated format for compressed image data;

c) the abbreviated format for table-specification data.

This Recommendation | International Standard extends the specification of these compressed data formats to support the use of the extensions in combination with the processes defined by CCITT Rec. T.81 | ISO/IEC 10918-1.

The compressed data format requirements are that any compressed image data represented in interchange format, abbreviated format for compressed image data, or abbreviated format for table-specification data shall comply with the syntax and code assignments appropriate for the decoding process and extensions selected, as specified in Annex B of CCITT Rec. T.81 | ISO/IEC 10918-1 and Annex B of this Recommendation | International Standard.

6 Encoder requirements

An encoding process converts source image data to compressed image data. CCITT Rec. T.81 | ISO/IEC 10918-1 specifies a number of distinct encoding processes. This Recommendation | International Standard defines encoding extensions which may be used in combination with the encoding processes defined by CCITT Rec. T.81 | ISO/IEC 10918-1.

An extended encoder is an embodiment of one (or more) of the encoding processes specified in CCITT Rec. T.81 | ISO/IEC 10918-1 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 the requirements stated in clause 6 of CCITT Rec. T.81 | ISO/IEC 10918-1 and satisfy at least one of the following two additional requirements.

An extended encoder shall:

a) with appropriate accuracy, convert source image data to compressed image data which comply with the interchange format syntax specified in Annex B for the encoding process(es) and extension(s) embodied by the encoder;

b) with appropriate accuracy, convert source image data to compressed image data which comply with the abbreviated format for compressed image data syntax specified in Annex B for the encoding process(es) and extension(s) embodied by the encoder.

For each of the processes defined by CCITT Rec. T.81 | ISO/IEC 10918-1, the compliance tests for the above requirements are specified in ITU-T Rec. T.83 | ISO/IEC 10918-2. For each of the extensions defined by this Recommendation | International Standard, the compliance tests for the above requirements are specified in Annex G.

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 for that process and extension. An encoder is only required to meet the compliance tests, and to generate the compressed data format according to Annex B for those parameter values which it does use.

7 Decoder requirements

A decoding process converts compressed image data to reconstructed image data. CCITT Rec. T.81 | ISO/IEC 10918-1 specifies a number of distinct decoding processes. This Recommendation | International Standard defines decoding extensions which may be used in combination with the decoding processes defined by CCITT Rec. T.81 | ISO/IEC 10918-1.

An extended decoder is an embodiment of one (or more) of the decoding processes specified in CCITT Rec. T.81 | ISO/IEC 10918-1 used in combination with one or more of the decoding extensions specified herein. In order to comply with this Recommendation | International Standard, an extended decoder shall satisfy the requirements stated in clause 7 of CCITT Rec. T.81 | ISO/IEC 10918-1 and satisfy all three of the following additional requirements.

An extended decoder shall:

a) with appropriate accuracy, 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 B for the decoding process(es) and extension(s) embodied by the decoder;

b) accept and store any table-specification data which comply with the abbreviated format for table-specification data syntax specified in Annex B for the decoding process(es) and extension(s) embodied by the decoder;

c) with appropriate accuracy, convert to reconstructed image data any compressed image data which comply with the abbreviated format for compressed image data syntax specified in Annex B for the decoding process(es) and extension(s) embodied by the decoder, provided that the table-specification data required for decoding the compressed image data has previously been installed into the decoder.

Annex A

Mathematical definitions

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

This annex specifies the changes and additions to Annex A of CCITT Rec. T.81 | ISO/IEC 10918-1.

A.1 Geometries of image data

The following extensions are in addition to the specifications of CCITT Rec. T.81 | ISO/IEC 10918-1.

A.1.1 Simple tiling

In simple tiling, the tiles with the same maximum dimensions are referred to as 'nominal'. All tiles on the right border, apart from the bottom tile, have the same maximum vertical dimension as the nominal tiles and their maximum horizontal dimension is less than or equal to the maximum horizontal dimension of the nominal tiles. All tiles on the bottom border apart from the rightmost tile have the same maximum horizontal dimension as the nominal tiles and their maximum vertical dimension is equal to or less than the maximum vertical dimension of the nominal tiles. The maximum horizontal dimension of the bottom rightmost tile is the same as the maximum horizontal dimension of the right border tiles and its maximum vertical dimension is the same as the maximum vertical dimension of the bottom border tiles. The maximum horizontal dimension and maximum vertical dimension of the nominal tiles shall be multiples of $8*H_{max}$ and $8*V_{max}$ respectively so that no padding is required, except possibly in the case of the border tiles.

A.1.2 Tile geometry for composite tiling

For composite tiling, the concept of tile geometry is needed. Note that this concept is not included in CCITT Rec. T.81 | ISO/IEC 10918-1. The tile geometry is defined by additional meaning associated with the X, Y and sampling factors parameters.

The X and Y parameters of the frame header or DHP marker segment shall also define the width and height of the tile. The tile is associated with a rectangular grid made of X*Y square elements. The dimension of each element is one tile grid unit.

For each component the subsampling ratios shall be derived from the H_i and V_i sampling factors as follows:

– The horizontal subsampling ratio (SFH_i) is given by H_{max}/H_i, where H_{max} is the largest H_i. The vertical subsampling ratio (SFV_i) is given by V_{max}/V_i, where V_{max} is the largest V_i.

The registration of each component relative to the tile grid (and to each other component) is defined to give "even coverage" of the tile grid by the component samples. The vertical registration distance and horizontal registration distance are specified in units of half a tile grid unit.

The case of $H_{max} = 4$ (or $V_{max} = 4$) and where at least one of the other H_i (or V_i) has a value of 3, is not covered properly by the above. The composing of tiles in such a case is not specified and is left to the application.

The number of possible values for the horizontal registration distance of a component is given by $2*H_{max}/H_i$. The number of possible values for the vertical registration distance of a component is given by $2*V_{max}/V_i$.

Individual tiles used in composite tiling require a component registration selection (DCR) marker segment to specify the spatial positioning of the components.

A.1.3 Reference and image grids

Two additional grids are defined for composite tiling: a reference grid and an image grid. The dimension of each "grid element" of the reference grid is one reference grid unit. The dimension of each "grid element" of the image grid is one image grid unit.

The tile can be scaled up with an integer scaling factor in each dimension and put in the reference grid at the specified offset. The resulting tiled image described on the reference grid can then be scaled down with an integer scaling factor in each dimension, onto a third grid called the image grid. The up-scaling and down-scaling algorithms are not specified.

A.2 DCT coefficient quantization (informative) and dequantization (normative)

The following are extensions to the quantization and dequantization specified in CCITT Rec. T.81 | ISO/IEC 10918-1.

After the FDCT is computed for a block, each of the resulting 63 AC coefficients S_{vu} is quantized by the following equation, where '/' indicates integer division with truncation towards zero:

$$Sq_{vu} = round\{(S_{vu} * 16)/(Q_{vu} * Q\_SCALE)\}$$

The quantizer scaling does not affect the quantization of the DC coefficient:

$$Sq_{00} = round(S_{00} / Q_{00})$$

The quantizer step size is the element Q_{vu} from the quantization table specified by the frame parameter T_{qi}. The scale factor Q_SCALE is the entry in Table C.1 whose index is the current value of SCALE_CODE (see C.2).

At the decoder, the dequantization is specified by the following equations:

$$R_{vu} = Sq_{vu} * Q_{vu} * Q\_SCALE / 16$$

and,

$$R_{00} = Sq_{00} * Q_{00}$$

Annex B

Compressed data formats

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

This annex specifies the changes and additions to Annex B of CCITT Rec. T.81 | ISO/IEC 10918-1.

B.1 General aspects of the compressed data format specifications

B.1.1 Marker assignments

All markers shall be assigned two-byte codes: a X'FF' byte followed by a second byte which is not equal to X'00' or X'FF'. The second byte is specified in Table B.1 for each defined marker. An asterisk (*) indicates a marker which is significant in its own right and is not the start of a marker segment.

B.1.2 Syntax

In B.2 to B.7 the interchange format syntax is specified. For the purpose of this Recommendation | International Standard, the syntax consists of:

- the required ordering of markers, parameters, and entropy-coded segments;
- identification of optional or conditional constituent parts;
- the name, symbol, and definition of each marker and parameter;
- the allowed values of each parameter;
- any restrictions on the above which are specific to the various coding processes or extensions.

The ordering of constituent parts and the identification of which are optional or conditional is specified by the syntax figures in B.2 to B.7. Names, symbols, definitions, allowed values, conditions, and restrictions are specified immediately below each syntax figure.

B.1.3 Conventions for syntax figures

The syntax figures in B.2 to B.7 are a part of the interchange format specification. The following conventions, illustrated in Figure B.1, apply to these figures:

- **Segment indicator:** A thick-lined box encloses either a marker segment, an entropy-coded data segment, or combinations of these.
- **Optional/conditional indicator:** Square brackets indicate that a marker or marker segment is only optionally or conditionally present in the compressed image data.
- **Parameter/marker indicator:** A thin-lined box encloses either a marker or a single parameter.
- **Parameter length indicator:** The width of a thin-lined box is proportional to the parameter length (4, 8, 16, or 32 bits, shown as E, B, D, and G respectively in Figure B.1) of the marker or parameter it encloses; the width of thick-lined boxes is not meaningful.
- **Ordering:** In the interchange format, a parameter or marker shown in a figure precedes all of those shown to its right, and follows all of those shown to its left;
- **Entropy-coded data indicator:** Angled brackets indicate that the entity enclosed has been entropy encoded.

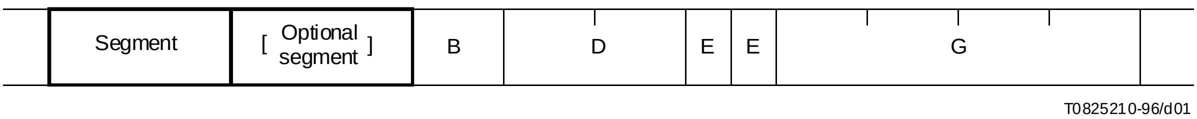

Figure B.1 – Syntax notation conventions

Table B.1 – Marker code assignments

Code assignment	Symbol	Description
Start of Frame markers, non-differential, Huffman coding		
X'FFC0' X'FFC1' X'FFC2' X'FFC3'	SOF_0 SOF_1 SOF_2 SOF_3	Baseline DCT Extended sequential DCT Progressive DCT Lossless (sequential)
Start of Frame markers, differential, Huffman coding		
X'FFC5' X'FFC6' X'FFC7'	SOF_5 SOF_6 SOF_7	Differential sequential DCT Differential progressive DCT Differential lossless (sequential)
Start of Frame markers, non-differential, arithmetic coding		
X'FFC8' X'FFC9' X'FFCA' X'FFCB'	JPG SOF_9 SOF_{10} SOF_{11}	Reserved for additional JPEG extensions Extended sequential DCT Progressive DCT Lossless (sequential)
Start of Frame markers, differential, arithmetic coding		
X'FFCD' X'FFCE' X'FFCF'	SOF_{13} SOF_{14} SOF_{15}	Differential sequential DCT Differential progressive DCT Differential lossless (sequential)
Huffman table specification		
X'FFC4'	DHT	Define Huffman table(s)
Arithmetic coding conditioning specification		
X'FFCC'	DAC	Define arithmetic coding conditioning(s)
Restart interval termination		
X'FFD0' through X'FFD7'	RST_m*	Restart with modulo 8 count "m"
Other markers		
X'FFD8' X'FFD9' X'FFDA' X'FFDB' X'FFDC' X'FFDD' X'FFDE' X'FFDF' X'FFE0' through X'FFEF' X'FFF7' through X'FFFD' X'FFFE'	SOI* EOI* SOS DQT DNL DRI DHP EXP APP_n JPG_n COM	Start of image End of image Start of scan Define quantization table(s) Define number of lines Define restart interval Define hierarchical progression Expand reference component(s) Reserved for application segments Reserved for additional JPEG extensions Comment
Reserved markers		
X'FF01' X'FF02' through X'FFBF'	TEM* RES	For temporary private use in arithmetic coding Reserved
Version 1 extensions		
X'FFF0' X'FFF1' X'FFF2' X'FFF3' X'FFF4' X'FFF5' X'FFF6'	VER DTI DTT SRF SRS DCR DQS	Version Define tiled image Define tile Selectively refined frame Selectively refined scan Define component registration Define quantizer scale selection

B.1.4 Conventions for symbols, code lengths, and values

Following the figure for each syntax defined in B.2 to B.7, the symbol, name, and definition for each marker and parameter shown in the figures, are specified. For each parameter, the length and allowed values are also specified in tabular form.

The following conventions apply to symbols for markers and parameters:

– all marker symbols have three upper-case letters, and some also have a subscript. Examples: SOI, SOF_n;

– all parameter symbols have one or two upper-case letters; some also have one or two lower-case letters and some have subscripts. Examples include: TFvs, Y, Nf, H_i, Tq_i.

B.2 General sequential and progressive syntax

This subclause specifies the interchange format syntax which applies to all coding processes for sequential DCT-based, progressive DCT-based, and lossless modes of operation.

B.2.1 Scan header syntax

Figure B.2 specifies the scan header which shall be present at the start of a scan. This header specifies which component(s) are contained in the scan, specifies the destinations from which the entropy tables to be used with each component are retrieved, and (for the progressive DCT) specifies which part of the DCT quantized coefficient data is contained in the scan. For lossless processes the scan parameters specify the predictor and the point transform.

NOTE – If there is only one image component present in a scan, that component is, by definition, non-interleaved. If there is more than one image component present in a scan, the components present are, by definition, interleaved.

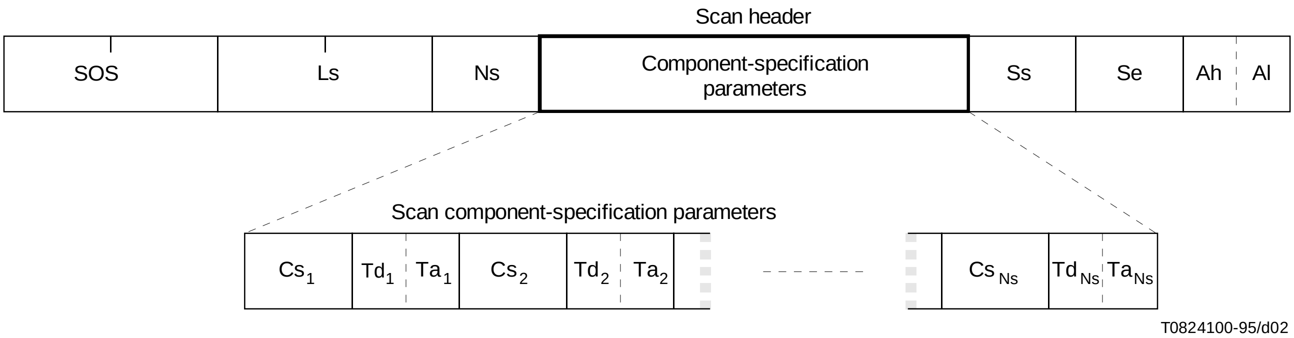

Figure B.2 – Scan header syntax

The marker and parameters shown in Figure B.2 are defined below. The size and allowed values of each parameter are given in Table B.2.

– **SOS:** Start of scan marker – Marks the beginning of the scan parameters.

– **Ls:** Scan header length – Specifies the length in bytes of the scan header shown in Figure B.2. This length parameter encodes the number of bytes in the marker segment, including the length parameter but excluding the two byte marker.

– **Ns:** Number of image components in scan – Specifies the number of source image components in the scan. The value of Ns shall be equal to the number of sets of scan component parameters (Cs_j, Td_j, and Ta_j) present in the scan header.

– **Cs_j:** Scan component selector – Selects which of the Nf image components specified in the frame parameters shall be the *j*th component in the scan. Each Cs_j shall match one of the C_i values specified in the frame header, and the ordering in the scan header shall follow the ordering in the frame header. If Ns > 1, the order of interleaved components in the MCU is Cs_1 first, Cs_2 second, etc. If Ns > 1, the following restriction shall be placed on the image components contained in the scan:

- For compressed data streams which do not use the 20 blocks per MCU extension defined in this Recommendation | International Standard:

$$\sum_{j=1}^{N_s} H_j \times V_j \leq 10$$

- For compressed data streams which use the 20 blocks per MCU extension defined in this Recommendation | International Standard:

$$\sum_{j=1}^{N_s} H_j \times V_j \leq 20$$

where H_j and V_j are the horizontal and vertical sampling factors for scan component j. These sampling factors are specified in the frame header for component i, where i is the frame component index for which frame component identifier C_i matches scan component selector Cs_j.

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

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

Then the summation of $H_j \times V_j$ is $(4 \times 1) + (1 \times 2) + (2 \times 2) = 10$.

The value of Cs_j shall be different from the values of Cs_1 to Cs_{j-1}.

– **Td_j:** DC entropy coding table destination selector – Specifies one of four possible DC entropy coding table destinations from which the entropy table needed for decoding of the DC coefficients of component Cs_j is retrieved. The DC entropy table shall have been installed in this destination (see B.2.4.2 and B.2.4.3 in CCITT Rec. T.81 | ISO/IEC 10918-1) by the time the decoder is ready to decode the current scan. This parameter specifies the entropy coding table destination for the lossless processes.

– **Ta_j:** AC entropy coding table destination selector – Specifies one of four possible AC entropy coding table destinations from which the entropy table needed for decoding of the AC coefficients of component Cs_j is retrieved. The AC entropy table selected shall have been installed in this destination (see B.2.4.2 and B.2.4.3 CCITT Rec. T.81 | ISO/IEC 10918-1) by the time the decoder is ready to decode the current scan. This parameter is zero for the lossless processes.

– **Ss:** Start of spectral or predictor selection – In the DCT modes of operation, this parameter specifies the first DCT coefficient in each block in zigzag order which shall be coded in the scan. This parameter shall be set to zero for the sequential DCT processes. In the lossless mode of operations this parameter is used to select the predictor.

– **Se:** End of spectral selection – Specifies the last DCT coefficient in each block in zigzag order which shall be coded in the scan. This parameter shall be set to 63 for the sequential DCT processes. In the lossless mode of operations this parameter has no meaning. It shall be set to zero.

– **Ah:** Successive approximation bit position high – This parameter specifies the point transform used in the preceding scan (i.e. successive approximation bit position low in the preceding scan) for the band of coefficients specified by Ss and Se. This parameter shall be set to zero for the first scan of each band of coefficients. In the lossless mode of operations this parameter has no meaning. It shall be set to zero.

– **Al:** Successive approximation bit position low or point transform – In the DCT modes of operation this parameter specifies the point transform, i.e. bit position low, used before coding the band of coefficients specified by Ss and Se. This parameter shall be set to zero for the sequential DCT processes. In the lossless mode of operations, this parameter specifies the point transform, Pt.

The entropy coding table destination selectors, Td_j and Ta_j, specify either Huffman tables (in frames using Huffman coding) or arithmetic coding tables (in frames using arithmetic coding). In the latter case, the entropy coding table destination selector specifies both an arithmetic coding conditioning table destination and an associated statistics area.

Table B.2 – Scan header parameter size and values

Parameter	Size (bits)	Values			
		Sequential DCT		Progressive DCT	Lossless
		Baseline	Extended		
Ls	16	$6 + 2 \times Ns$			
Ns	8	1-4			
Cs_j	8	0-255[a)]			
Td_j	4	0-1	0-3	0-3	0-3
Ta_j	4	0-1	0-3	0-3	0
Ss	8	0	0	0-63	1-7[b)]
Se	8	63	63	Ss-63[c)]	0
Ah	4	0	0	0-13	0
Al	4	0	0	0-13	0-15

a) Csj Shall be a member of the set of C_i specified in the frame header.

b) 0 For lossless differential frames in the hierarchical mode.

c) 0 If Ss equals zero.

NOTE – The contents of this table are identical to Table B.3 in CCITT Rec. T.81 | ISO/IEC 10918-1

B.3 Version marker segment syntax

Inclusion of the version marker segment is mandatory for a data stream which uses any of the capabilities listed in Table B.5. The first version marker segment shall appear after SOI and before any other marker apart from APPn or COM marker segments. Except when using composite tiling, its parameters shall match the capabilities needed to decode the complete compressed image data stream. For composite tiling its parameters shall match the capabilities needed to decode the first tile (see Annex E).

A version marker segment with 0.0 version specifier shall not appear as the first (or only) version marker segment. This will provide protection for version 0.0 decoders which are not required to interpret (or jump over) version marker segments.

Other version marker segments may appear in any place of the bitstream where tables or miscellaneous marker segments are legal. These additional marker segments (if they appear) shall describe the capability needed to decode the subsequent part of the bitstream until the EOI marker or the next version marker segment. For images without composite tiling, these additional marker segments are optional. For composite tiling, if the level of functionality specified in the last encountered version marker segment which appears before a tile (and not within a tile) is not sufficient to decode the next tile, an additional marker segment shall appear before that tile that will specify the required functionality.

Figure B.3 specifies the version marker segment.

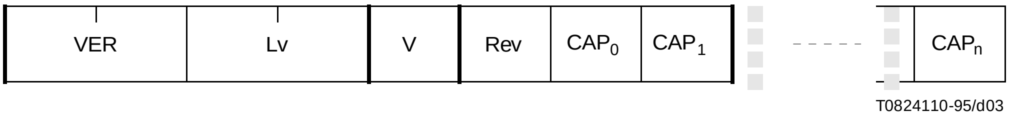

Figure B.3 – Version marker segment syntax

The marker and parameters shown in Figure B.3 are defined below. The size and allowed values of each parameter are given in Table B.3:

– **VER:** Version marker – Marks the beginning of the version marker parameters.

– **Lv:** Version marker segment length – Specifies the sum total length of all version marker segment parameters shown in Figure B.3

– **V:** Version number – Specifies the major version.

– **Rev:** Revision number – Specifies the minor revision of the version.

The two bytes composite parameter <version number>.<revision number> (**V.Rev**) is referred to as the Version specifier.

The version specifier for CCITT Rec. T.81 | ISO/IEC 10918-1 is 0.0. The first revision with the extensions described in ITU-T Rec. T.84 | ISO/IEC 10918-3 shall have the version specifier 1.0.

Encoders shall indicate the smallest version specifier having sufficient capabilities to decode the bitstream, except that a version marker segment with 0.0 version specifier shall not appear as the first (or only) version marker segment.

Decoders shall not attempt to decode the subsequent part of the bitstream if the major version number is greater than that implemented by the decoder. In this event, decoders shall parse but not decode the subsequent compressed data until the next version marker segment or the end of the compressed data stream is reached. If the major version number code is less than or equal to that implemented by the decoder, the decoder shall attempt to decode the subsequent part of the bitstream regardless of the value of the minor revision number.

– **CAP_i:** Capability indicator – Added capability indicator for version number *i*. The number of capability indicator bytes shall be equal to **V** + 1.

Table B.3 – Version marker segment parameter sizes and values

<table>
<tr><th rowspan="3">Parameter</th><th rowspan="3">Size (bits)</th><th colspan="4">Values</th></tr>
<tr><th colspan="2">Sequential DCT</th><th rowspan="2">Progressive DCT</th><th rowspan="2">Lossless</th></tr>
<tr><th>Baseline</th><th>Extended</th></tr>
<tr><td>Lv</td><td>16</td><td colspan="4">5, V = 0
6, V = 1</td></tr>
<tr><td>V</td><td>8</td><td colspan="4">0, 1</td></tr>
<tr><td>Rev</td><td>8</td><td colspan="4">0</td></tr>
<tr><td>CAP_i</td><td>8
8</td><td colspan="4">CAP_0 use Table B.4, version = 0
CAP_1 use Table B.5, version = 1
CAP_i (i > 1) reserved for future versions</td></tr>
</table>

The bits of the capability indicator byte for Version 0 have the following meaning (from most significant to least significant): Hierarchical, lossless, progressive, full progression, 12-bits, arithmetic, and extended. Only the 29 combinations defined in Table B.4 are legal for $V = 0$. Note that for the hierarchical processes, the capability indicator coded is the largest of the binary values which represent the capabilities for each stage of the hierarchical progression. Also, if the first frame of a hierarchical progression uses a DCT-based process, the capability indicator shall indicate a DCT-based process whether or not there is a final lossless stage.

Table B.4 – Capability indicator byte for Version 0

Number	Coding process	CAP_0 value
1	Baseline sequential	0000 0000
2	Extended sequential, Huffman, 8-bits	0000 0001
3	Extended sequential, arithmetic, 8-bits	0000 0011
4	Extended sequential, Huffman, 12-bits	0000 0101
5	Extended sequential, arithmetic, 12-bits	0000 0111
6	Spectral selection, Huffman, 8-bits	0001 0001
7	Spectral selection, arithmetic, 8-bits	0001 0011
8	Full progression, Huffman, 8-bits	0001 1001
9	Full progression, arithmetic, 8-bits	0001 1011
10	Spectral selection, Huffman, 12-bits	0001 0101
11	Spectral selection, arithmetic, 12-bits	0001 0111
12	Full progression, Huffman, 12-bits	0001 1101
13	Full progression, arithmetic, 12-bits	0001 1111
14	Lossless, Huffman	0010 0001
15	Lossless, arithmetic	0010 0011
16	Hierarchical, sequential Huffman, 8-bits	0100 0001
17	Hierarchical, sequential arithmetic, 8-bits	0100 0011
18	Hierarchical, sequential, Huffman, 12-bits	0100 0101
19	Hierarchical, sequential, arithmetic, 12-bits	0100 0111
20	Hierarchical, Spectral selection, Huffman, 8-bits	0101 0001
21	Hierarchical, Spectral selection, arithmetic, 8-bits	0101 0011
22	Hierarchical, Full progression, Huffman, 8-bits	0101 1001
23	Hierarchical, Full progression, arithmetic, 8-bits	0101 1011
24	Hierarchical, Spectral selection, Huffman, 12-bits	0101 0101
25	Hierarchical, Spectral selection, arithmetic, 12-bits	0101 0111
26	Hierarchical, Full progression, Huffman, 12-bits	0101 1101
27	Hierarchical, Full progression, arithmetic, 12-bits	0101 1111
28	Hierarchical, Lossless, Huffman	0110 0001
29	Hierarchical, Lossless, arithmetic	0110 0011

The bits of the additional capability indicator byte for Version 1 shall have the following meaning (from least significant to most significant): up to 20 blocks per MCU, variable quantization, selective refinement (3 bits), tiling type (2-bits), and 1 reserved bit. The reserved bit, bit 7, shall be set to zero (see Table B.5).

Table B.5 – Capability indicator byte for Version 1

Capability	Bit positions
10 < blocks per MCU <= 20	0xxx xxx1
Variable quantization	0xxx xx1x
Hierarchical selective refinement	0xxx x1xx
Progressive selective refinement	0xxx 1xxx
Component selective refinement	0xx1 xxxx
Tiling No tiling Simple tiling Pyramidal tiling Composite tiling	 000x xxxx 001x xxxx 010x xxxx 011x xxxx
NOTE – "x" indicates "don't care".	

B.4 Selective refinement syntax

B.4.1 General

Selective refinement refers to selecting a (rectangular) region of one or more of the image components for further refinement.

There are several types of selective refinement. One is when only a region of one or more components is further refined by the next differential frame of a hierarchical sequence. This type of refinement is called hierarchical selective refinement. The syntax for hierarchical selective refinement is specified in B.4.2.

A second type of selective refinement provides for the next scan to add more coefficients, more bits to the coefficients, or both, to a selected region of one or more components. This type includes spectral selection progressive selective refinement, successive approximation progressive selective refinement, and progressive selective refinement combining both spectral selection and successive approximation. The syntax for progressive selective refinement is specified in B.4.3.

The third type of selective refinement allows for some colour components to be used in selected regions of the image only. This type is called component selective refinement. The syntax for component selective refinement is the same as for progressive selective refinement specified in B.4.3.

B.4.2 Hierarchical selective refinement syntax

Hierarchical selective refinement of a differential frame is signalled by the inclusion of a Selectively Refined Frame (SRF) marker segment as one of the table-specification segments or miscellaneous marker segments preceding the differential frame. A differential frame which completely refines the previous layer (as in CCITT Rec. T.81 | ISO/IEC 10918-1) shall not have a preceding SRF marker segment.

Figure B.4 shows the Selectively Refined Frame (SRF) marker segment syntax.

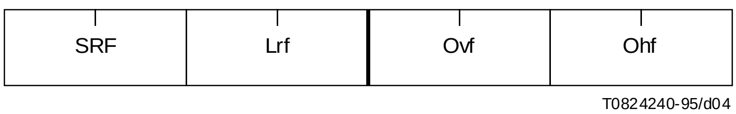

Figure B.4 – Selectively refined frame marker segment syntax

The marker and parameters shown in Figure B.4 are defined below. The size and allowed values of each parameter are given in Table B.6.

- **SRF:** Selectively refined frame marker – Marks the beginning of the selectively refined frame parameters.
- **Lrf:** Selectively refined frame marker segment length – Specifies the sum total length of all selectively refined frame parameters shown in Figure B.4.
- **Ovf:** Selectively refined frame vertical offset – Specifies the maximum distance, in lines, between the top border of the refined region of each component and the top border of the component array (possibly upsampled from previous stages).
- **Ohf:** Selectively refined frame horizontal offset – Specifies the maximum distance, in samples, between the left border of the refined region of each component and the left border of the component (possibly upsampled from previous stages).

The maximum horizontal and vertical dimensions specified in the SOF marker segment of the differential frame which is preceded by a SRF marker segment shall specify the reduced maximum dimensions of the component regions being refined. The region of each component being refined shall fall entirely within the dimensions of the component array that was upsampled from the previous stages.

Table B.6 – Selectively refined frame parameter sizes and values

Parameter	Size (bits)	Values
Lrf	16	6
Ovf	16	$0 \le Ovf \le 2^{16} - 1$
Ohf	16	$0 \le Ohf \le 2^{16} - 1$

B.4.3 Progressive and component selective refinement syntax

Progressive and component selective refinement of a scan is signalled by the inclusion of a Selectively Refined Scan (SRS) marker segment as one of the table-specification segments or miscellaneous marker segments preceding that scan. A scan which completely “covers” the whole image (as in CCITT Rec. T.81 | ISO/IEC 10918-1) shall not have a preceding SRS marker segment. Also, component regions defined by a selectively refined scan shall not partially overlap the component regions defined by a preceding scan containing the same components.

Figure B.5 shows the selectively refined scan (SRS) marker segment syntax.

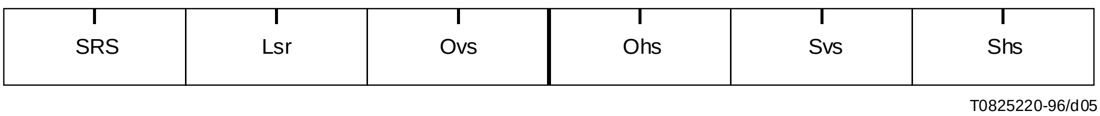

Figure B.5 – Selectively refined scan marker segment syntax

The marker and parameters shown in Figure B.5 are defined below. The size and allowed values of each parameter are given in Table B.7.

- **SRS:** Selectively refined scan marker – Marks the beginning of the selectively refined scan parameters.
- **Lrs:** Selectively refined scan marker segment length – Specifies the length of all selectively refined scan parameters shown in Table B.7.
- **Ovs:** Selectively refined scan vertical offset – Specifies the maximum distance, in lines, between the top border of the refined region of each component and the top border of the component array.

– **Ohs:** Selectively refined scan horizontal offset – Specifies the maximum distance, in samples, between the left border of the refined region of each component and the left border of the component array.

– **Svs:** Selectively refined scan vertical size – Specifies the maximum vertical dimension of the refined components regions, in lines.

– **Shs:** Selectively refined scan horizontal size – Specifies the maximum horizontal dimension of the refined components regions, in samples.

For progressive selective refinement, Ovs and Svs must be specified as multiples of $8*V_{max}$. Similarly, Ohs and Shs must be specified as multiples of $8*H_{max}$.

Table B.7 – Selectively refined scan parameter sizes and values

Parameter	Size (bits)	Values
Lrs	16	10
Ovs	16	$0 \leq \text{Ovs} \leq 2^{16} - 1$
Ohs	16	$0 \leq \text{Ohs} \leq 2^{16} - 1$
Svs	16	$1 \leq \text{Svs} \leq 2^{16} - 1$
Shs	16	$1 \leq \text{Shs} \leq 2^{16} - 1$

B.5 Tiling syntax

This subclause specifies the syntax which applies to all tiled images.

B.5.1 High-level syntax

Figure B.6 specifies the order of the high level constituent parts for all tiled images.

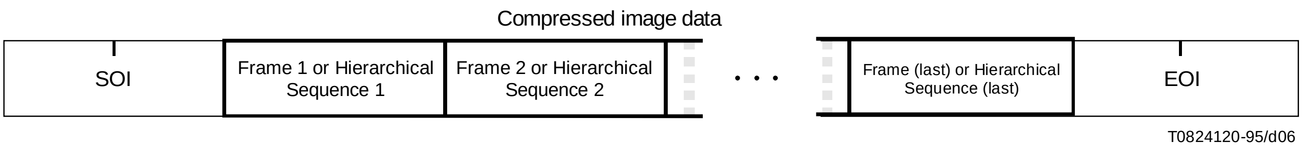

Figure B.6 – Image tiling high level syntax

– **SOI:** Start of image marker – Marks the start of a compressed image.

– **EOI:** End of image marker – Marks the end of a compressed image.

Figure B.6 specifies that a tiled image shall begin with an SOI marker, shall contain at least two frames (where each frame is as specified in Figure B.2 of CCITT Rec. T.81 | ISO/IEC 10918-1) or at least two hierarchical sequences (where each sequence is defined as all the bitstream between the SOI and the EOI in Figure B.13 of CCITT Rec. T.81 | ISO/IEC 10918-1), or at least one frame and one hierarchical sequence, and end with an EOI marker.

The DTI marker segment shall appear once in the bitstream after SOI and before any of the DTT marker segments.

A DTT marker segment shall appear in each frame of the bitstream before the SOF marker of the frame, and in each hierarchical sequence of the bitstream before the DHP marker segment of the hierarchical sequence.

B.5.2 Tiled image syntax

A tiled image is signalled by inclusion of a Define Tiled Image (DTI) marker segment as one of the table-specification segments or miscellaneous marker segments preceding any of the SOF or DHP marker segments. An image which includes only one tile or one hierarchical sequence, and hence is not tiled, shall not include a DTI marker segment.

Figure B.7 shows the Define Tiled Image (DTI) marker segment syntax.

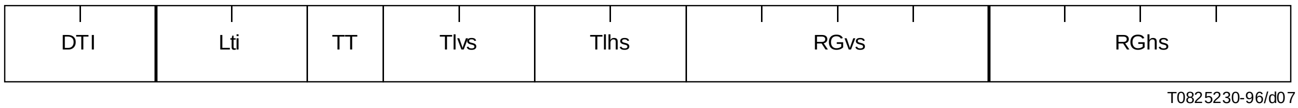

Figure B.7 – Define tiled image marker segment syntax

The marker and parameters shown in Figure B.7 are defined below. The size and allowed values of each parameter are given in Table B.8.

- **DTI:** Define tiled image marker – Marks the beginning of the define tiled image parameters.
- **Lti:** Define tiled image marker segment length – Specifies the sum total length of all define tiled image parameters shown in Figure B.7.
- **TT:** Tiling type – Specifies whether simple, pyramidal, or composite tiling is to be used.
- **TIvs:** Tiled image vertical scale – For simple and pyramidal tiling, not used (set to 1); for composite tiling, the integer ratio of the reference grid height to the image grid height.
- **TIhs:** Tiled image horizontal scale – For simple and pyramidal tiling, not used (set to 1); for composite tiling, the integer ratio of the reference grid width to the image grid width.
- **RGvs:** Reference grid height – For simple and pyramidal tiling, the maximum vertical dimension of the original image; for composite tiling, the reference grid height.
- **RGhs:** Reference grid width – For simple and pyramidal tiling, the maximum horizontal dimension of the original image; for composite tiling, the reference grid width.

Table B.8 – Define tiled image parameter sizes and values

Parameter	Size (bits)	Values
Lti	16	15
TT	8	0 = simple, 1 = pyramidal, 2 = composite
TIvs	16	1 for simple and pyramidal tiling $1 \le \mathrm{TIvs} \le 2^{16} - 1$ for composite tiling
TIhs	16	1 for simple and pyramidal tiling $1 \le \mathrm{TIhs} \le 2^{16} - 1$ for composite tiling
RGvs	32	$1 \le \mathrm{RGvs} \le 2^{32} - 1$
RGhs	32	$1 \le \mathrm{RGhs} \le 2^{32} - 1$

B.5.3 Tile syntax

A tile is signalled by inclusion of a Define Tile (DTT) marker segment as the first of the table specification segments or miscellaneous marker segments included in each frame or hierarchical sequence and preceding the SOF or DHP marker segments of that frame or hierarchical sequence respectively. Occurrence of the DTT marker segment shall reset the restart interval, arithmetic coding conditioning, and quantizer scale selection to their default values. These values may be redefined by a subsequent DRI, DAC, or DQS marker segment respectively.

All table specification and miscellaneous marker segments required to decode the frame or hierarchical sequence shall appear either once near the start of the compressed image data stream or immediately after each DTT marker segment. If these table and miscellaneous marker segment appear only once, they shall precede the DTI marker segment of the tiled image.

Figure B.8 shows the Define Tile (DTT) marker segment syntax.

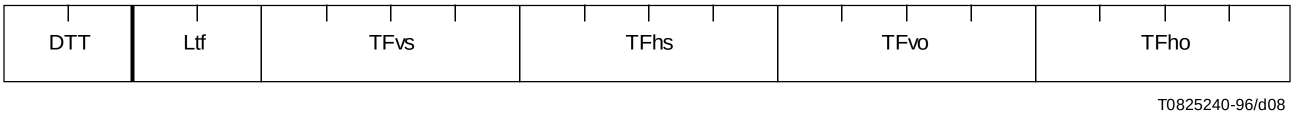

Figure B.8 – Define tile marker segment syntax

The marker and parameters shown in Figure B.8 are defined below. The size and allowed values of each parameter are given in Table B.9.

- **DTT:** Define tile marker – Marks the beginning of the define tile parameters.
- **Ltf:** Define tile marker segment length – Specifies the sum total length of all define tile parameters shown in Figure B.8.
- **TFvs:** Tile vertical scale – For simple tiling, not used (set to 1); for pyramidal tiling, the integer ratio between the maximum vertical dimension of the original image, and the maximum vertical dimension of the tiled image to which this tile belongs; for composite tiling, the integer ratio of the reference grid height to the tile grid height.
- **TFhs:** Tile horizontal scale – For simple tiling, not used (set to 1); for pyramidal tiling, the integer ratio between the maximum horizontal dimension of the original image, and the maximum horizontal dimension of the tiled image to which this tile belongs; for composite tiling, the integer ratio of the reference grid width to the tile grid width.
- **TFvo:** Tile vertical offset – For simple tiling, the maximum number, in lines, between the top border of the tile component array and the top border of the same component array of the tiled image; for pyramidal tiling, the maximum distance, in lines, between the top border of the tile component array and the top border of the same component array of the tiled image of the same resolution level; for composite tiling, the distance, in reference grid units, between the top border of the reference grid and the top border of the up-scaled tile grid (see Figure E.6).
- **TFho:** Tile horizontal offset – For simple tiling, the maximum number, in samples, between the left border of the tile components array and the left border of the same component array of the tiled image; for pyramidal tiling; the maximum distance, in samples, between the left border of the tile component array and the left border of the same component array of the tiled image of the same resolution level; for composite tiling, the distance, in reference grid units, between the left border of the reference grid and the left border of the up-scaled tile grid (see Figure E.6).

Table B.9 – Define tile parameter sizes and values

Parameter	Size (bits)	Values
Ltf	16	18
TFvs	32	$1 \le \text{TFvs} \le 2^{32} - 1$
TFhs	32	$1 \le \text{TFhs} \le 2^{32} - 1$
TFvo	32	$0 \le \text{TFvo} \le 2^{32} - 1$
TFho	32	$0 \le \text{TFho} \le 2^{32} - 1$

B.6 Component registration syntax

This marker segment specifies component registration for images that use composite tiling. Component registration specifies the spatial positioning of samples within components relative to the samples of other components. Component registration is signalled by inclusion of a Define Component Registration (DCR) marker segment as one of the table-specification segments or miscellaneous marker segments included in each frame or hierarchical sequence, and preceding the SOF or DHP marker segments of that frame or hierarchical sequence respectively. One DCR marker segment should be present for each component specified in the frame header or DHP marker segment. If no DCR marker segment is present for a particular component, the default values for the offsets shall be assumed to be zero.

Figure B.9 shows the Define Component Registration (DCR) marker segment syntax.

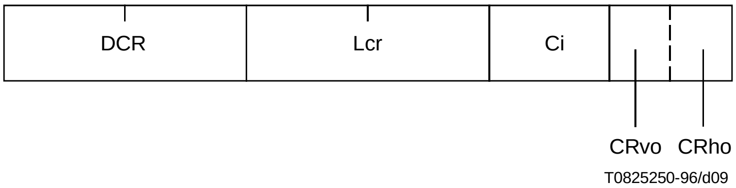

Figure B.9 – Define component registration marker segment syntax

The marker and parameters shown in Figure B.9 are defined below. The size and allowed values of each parameter are given in Table B.10.

– **DCR:** Define component registration marker – Marks the beginning of the define component registration parameters.

– **Lcr:** Define component registration marker segment length – Specifies the sum total length of all define component registration parameters shown in Figure B.9.

– **Ci:** Component identifier – Indicates which component the registration applies to.

– **CRvo:** Component registration vertical offset – Specifies the vertical distance in one-half tile grid units between the top border of the tile grid and the top border of the component array.

– **CRho:** Component registration horizontal offset – Specifies the horizontal distance in one-half tile grid units between the left border of the tile grid and the left border of the component array.

TABLE B.10 – Define component registration parameter sizes and values

Parameter	Size (bits)	Values
Lcr	16	4
Ci	8	$0 \le Ci \le 255$
CRvo	4	$0 \le CRvo \le 8$
CRho	4	$0 \le CRho \le 8$

B.7 Quantizer scale selection syntax

Figure B.10 specifies the marker segment that defines the table for selecting the quantizer scale factors. The Define Quantizer Scale Selection (DQS) marker segment may appear as one of the table-specification or miscellaneous marker segments preceding a scan or a frame, but may not appear between the scans of a frame.

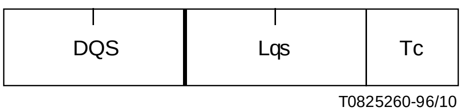

Figure B.10 – Define quantizer scale selection marker segment syntax

The marker and parameters shown in Figure B.10 are defined below. The size and allowed values of each parameter are given in Table B.11.

– **DQS:** Define quantizer scale selection marker – Marks the beginning of the define quantizer scale selection parameters.

– **Lqs:** Define quantizer scale selection marker segment length – Specifies the sum total length of all define quantizer scale selection parameters shown in Figure B.10.

– **Tc:** Quantizer scale selector – Tc = 0 indicates the linear table specified in Table C.1; Tc = 1 indicates the non-linear table specified in Table C.1.

Table B.11 – Define quantizer scale selection parameter sizes and values

Parameter	Size (bits)	Values
Lqs	16	3
Tc	8	Tc = 0, 1

Annex C

Variable quantization

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

This annex specifies the method for supporting variable quantization within a picture by extending the syntax for DCT based processes specified in CCITT Rec. T.81 | ISO/IEC 10918-1.

C.1 Introduction

In general, it is desirable to change the quantization matrix values within an image component in order to provide rate control or to adjust to the changing characteristics of an image (e.g. edge, detail) for improved coding efficiency. The procedure described here provides a means for changing the quantization matrix values by allowing for the rescaling of such values on a 8×8 block basis. The technical details of this procedure are explained below.

C.2 Description and parameter definition

The table of symbols used for the coding and decoding of the DC coefficient differences are extended by one symbol referred to as QS_CHANGE. As a result, the tables used by 8-bit processes are extended to 13 symbols and the tables used by the 12-bit processes are extended to 16 symbols. The QS_CHANGE symbol is used to signal that the next five bits should be decoded to specify a new quantizer scale factor. These five bits define a parameter called SCALE_CODE that is used as an address into a look-up table. The entries of the look-up table specify the value of Q_SCALE, the parameter used to scale all the AC values of the quantization matrix.

Two different look-up tables are allowed: a linear table, and a logarithmic table. The entries of the two tables are shown in Table C.1. A marker segment called DQS (Define Quantizer Select) indicates which of the two tables should be used. This marker segment can be located at the place indicated in Figure B.2 of CCITT Rec. T.81 | ISO/IEC 10918-1 where the table-specification segments or miscellaneous marker segments may be present. Once the value of Q_SCALE has been determined, it is used to quantize all the succeeding 8×8 blocks regardless of their component specification until the next occurrence of the QS_CHANGE symbol. The QS_CHANGE symbol only affects components in the current scan.

C.3 QS_CHANGE signalling

The following symbol is added to the DC table to specify a quantizer scale change

QS_CHANGE = X'15'

The five bits following specify the value of the SCALE_CODE parameter.

C.4 DCT coefficient quantization and dequantization

The value of Q_SCALE is set to 16 at the start of each scan and restart interval until it is modified by a subsequent occurrence of QS_CHANGE. Q_SCALE is derived from the SCALE_CODE parameter using Table C.1 and is then used to quantize the AC coefficients according to the process defined in A.2 The linear look-up table of Table C.1 is selected by previously coding a DQS marker segment with parameter Tc = 0. The non-linear look-up table of Table C.1 is selected by Tc = 1. A DQS marker segment shall appear before the SCALE_CODE parameter is coded in the data stream, and shall not appear between the scans of a frame.

For progressive encoding processes, the changes in Q_SCALE are only signalled in the first scan of each component. These changes shall apply to all subsequent scans on the same block boundaries as signalled in the first scan.

C.5 Huffman decoding of SCALE_CODE or quantizer scale change

The procedure outlined in F.2.2.1 of CCITT Rec. T.81 | ISO/IEC 10918-1 for the Huffman decoding of DC coefficients is modified to allow for the signalling of the quantizer scale change. First, the DECODE procedure outlined in Figure F.16 of CCITT Rec. T.81 | ISO/IEC 10918-1 is invoked to decode the difference magnitude category. If the decoded value is X'15', then the presence of the QS_CHANGE is detected. Consequently, the five bits following the decoded symbol are used to specify the value of SCALE_CODE. Finally, the value of SCALE_CODE is used as an index into Table C.1 to specify the value of Q_SCALE.

Table C.1 – Relationship between SCALE_CODE values and Q_SCALE
(taken from Table 7-6 of ITU-T Rec. H.262 | ISO/IEC 13818-2 (MPEG2))

SCALE_CODE	Q_SCALE	
	Linear	Non-linear
0	(Forbidden)	
1	2	1
2	4	2
3	6	3
4	8	4
5	10	5
6	12	6
7	14	7
8	16	8
9	18	10
10	20	12
11	22	14
12	24	16
13	26	18
14	28	20
15	30	22
16	32	24
17	34	28
18	36	32
19	38	36
20	40	40
21	42	44
22	44	48
23	46	52
24	48	56
25	50	64
26	52	72
27	54	80
28	56	88
29	58	96
30	60	104
31	62	112

C.6 Arithmetic decoding of SCALE_CODE or quantizer scale change

For arithmetic coding, the decoding procedures described in F.2.4.1 of CCITT Rec. T.81 | ISO/IEC 10918-1 are modified to allow for the signalling of the quantizer scale change. A coding decision shall be introduced at the beginning of coding each block. The decision shall have its own context per component. A binary decision “1” indicates a scale change and a binary decision “0” indicates no scale change. The decision is initialized to an LPS probability of 0.5 (Qe = X’5A1D’ and MPS = ‘0’) (see Figure C.1).

The context for the scale change decision, SQ, shall be entry 49, and the context for the scale code, SC, shall be entry 50 in Table F.4 of CCITT Rec. T.81 | ISO/IEC 10918-1. When the decision is X’01’ for the scale change decision, five bits shall be decoded consequently which specify the value of SCALE_CODE. Each bit of SCALE_CODE shall be decoded with fixed probability of 0.5 (Qe = X’5A1D’ and MPS = 0). When the decision is negative, no bits shall be decoded. The outcome of this decision shall in no way affect the statistics of subsequent decisions that are a part of the coding procedures as outlined in Figures F.4 and F.6 – F.9 of CCITT Rec. T.81 | ISO/IEC 10918-1. Finally, the value of SCALE_CODE is used as an index into Table C.1 to specify the value of Q_SCALE.

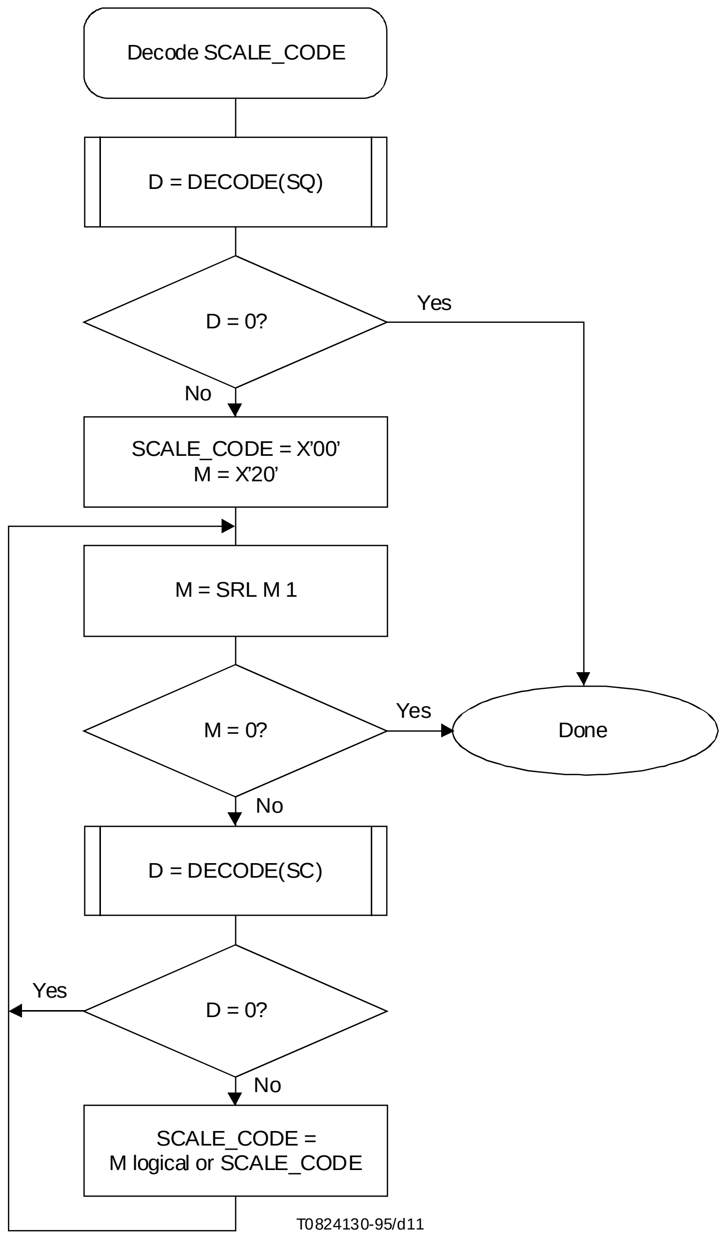

Figure C.1 – Decode SCALE_CODE procedure for arithmetic decoding processes

Annex D

Selective refinement

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

This annex provides a functional specification of the selective refinement extension. The selective refinement extension is an operation for further refining a (rectangular) region of one or more components of an image. The different types of selective refinement are specified below.

D.1 Hierarchical selective refinement

Hierarchical selective refinement is used in the hierarchical mode of operation for refining a region of each component of an image. The location of the region of each component to be selectively refined is specified immediately prior to a differential frame within an image. The location is defined in terms of offsets into the component with the largest size in the respective dimension. The size of the region is specified in the differential frame header. The difference image data reconstructed from the differential frame is then added only to the specified region of each component. Multiple selective refinement regions should not partially overlap and, for each hierarchical layer in turn, should be placed into the compressed image data stream in raster order based on the offset parameters.

Hierarchical selective refinement is signalled by an SRF marker segment as one of the tables/miscellaneous marker segments preceding a differential frame in an hierarchical sequence. Differential frames having the same maximum dimensions as the (possibly upsampled) preceding layer shall not be preceded by an SRF marker segment. Non-differential frames shall never be preceded by an SRF marker segment.

D.1.1 Modifications to the control procedure for decoding an image

The control procedure specified in Annex J of CCITT Rec. T.81 | ISO/IEC 10918-1 for decoding an image is modified during hierarchical selective refinement. The differential components are added, modulo 2^{16}, to the (possibly upsampled) reference components starting at the horizontal and vertical offsets specified in the preceding SRF marker segment. The horizontal and vertical offsets are specified with respect to the largest component of the frame in each dimension. The X and Y parameters along with the sampling factors of the differential frame following the SRF marker segment, give the number of lines and the number of samples per line of each component to be added to the upsampled reference components.

For a component with identifier C_i and sampling factors H_i and V_i , the distance, in lines, between the top border of the refined region of the component and the top border of the component array that was (possibly) upsampled from previous stages is given by:

$$\left\lceil O_{vf} * \frac{V_{max}}{V_i} \right\rceil$$

where $\lceil \; \rceil$ is the ceiling function. O_{vf} is the value of the parameter in the preceding SRF marker segment and V_{max} is the maximum of all vertical sampling factors in the DHP marker segment.

The distance, in samples, between the left border of the refined region of the component and the left border of the component array that was (possibly upsampled) from previous stages is given by:

$$\left\lceil O_{hf} * \frac{H_{max}}{H_i} \right\rceil$$

where O_{hf} is the value of the parameter in the preceding SRF marker segment and H_{max} is the maximum of all horizontal sampling factors in the DHP marker segment.

D.2 Progressive selective refinement

The second type of selective refinement, referred to as progressive selective refinement, is used in the DCT-based progressive mode of operation. Progressive selective refinement may be applied to the DCT-based processes which use spectral selection, successive approximation, or both procedures in combination.

When progressive selective refinement is applied to a scan which uses the spectral selection procedure, more non-zero DCT coefficients are added to a (rectangular) region of one or more components of an image. When it is applied to a scan which uses the successive approximation procedure, more bits are added to the DCT coefficients in a region of one or more components. Progressive selective refinement may also be applied to scans which use both procedures. In all cases, the location and size of the region of one or more components to be selectively refined is specified immediately prior to the scan which is used for selective refinement and shall be limited to MCU boundaries. The location is defined in terms of offsets into the component with the largest size in the respective dimension.

Progressive selective refinement is signalled by a Selectively Refined Scan (SRS) marker segment as one of the tables/miscellaneous marker segments preceding a scan in a progressive sequence. Scans having components of the same size as that signalled in the frame header shall not be preceded by an SRS marker segment.

D.2.1 Modifications to the control procedure for decoding an image

The control procedure specified in Annexes G and J of CCITT Rec. T.81 | ISO/IEC 10918-1 for decoding an image is modified during progressive selective refinement. The decoded DCT coefficients or, for successive approximation, the next bit of the decoded DCT coefficients are stored starting at the horizontal and vertical offsets specified in the preceding SRS marker segment. The horizontal and vertical offsets are specified with respect to the component with the largest size in the respective dimension. The maximum number of lines and maximum number of samples per line are given in the preceding SRS marker segment.

For a component with identifier C_i and sampling factors H_i and V_i, the distance, in lines, between the top border of the refined region of the component and the top border of the component array is given by:

$$\left\lceil O_{vs} * \frac{V_{max}}{V_i} \right\rceil$$

where $\lceil\ \rceil$ is the ceiling function. O_{vs} is the value of the parameter in the preceding SRS marker segment and V_{max} is the maximum of all vertical sampling factors in the frame header.

The distance, in samples, between the left border of the refined region of the component and the left border of the component array is given by:

$$\left\lceil O_{hs} * \frac{H_{max}}{H_i} \right\rceil$$

where O_{hs} is the value of the parameter in the preceding SRS marker segment and H_{max} is the maximum of all horizontal sampling factors in the frame header.

The vertical size, in lines, of the refined region of the component is given by:

$$\left\lceil S_{vs} * \frac{V_{max}}{V_i} \right\rceil$$

where S_{vs} is the value of the parameter in the preceding SRS marker segment. The horizontal size, in samples, of the refined region of the component is given by:

$$\left\lceil S_{hs} * \frac{H_{max}}{H_i} \right\rceil$$

where S_{hs} is the value of the parameter in the preceding SRS marker segment.

D.3 Component selective refinement

The third type of selective refinement, referred to as component selective refinement, may be used in all modes of operation for specifying a (rectangular) region of an image which contains colour components which do not exist in other regions of the image.

Component selective refinement is signalled by a Selectively Refined Scan (SRS) marker segment as one of the tables/miscellaneous marker segments preceding a scan. Scans having components of the same size as that signalled in the frame header shall not be preceded by an SRS marker segment. The location is defined in terms of offsets into the component with the largest size in the respective dimension.

D.3.1 Modifications to the control procedure for decoding an image

The control procedures specified in CCITT Rec. T.81 | ISO/IEC 10918-1 for decoding an image are modified during component selective refinement. The decoded samples of the components coded in the scan are stored starting at the horizontal and vertical offsets specified in the preceding SRS marker segment. The horizontal and vertical offsets are specified with respect to the component with the largest size in the respective dimension. The maximum number of lines and maximum number of samples per line are given in the preceding SRS marker segment.

The location and size of the refined region is determined in the same manner as in D.2.1.

Annex E

Tiling

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

This annex defines the technical features and uses of the tiling extension.

E.1 Introduction

Tiling is used when an image is too large to be easily processed by either the compressor or the decompressor. Tiling divides a large image into a set of small sub-images in order to:

- display a single tile on a given size screen;
- provide random access to image regions of interest;
- allow application specific methods to provide access control to parts or resolution levels of an image.

Tiling is completed before compression and the tile reconstruction of the image is completed after decompression. This technique allows for the reconstruction of the image with no border effects between tiles.

Techniques that are described in this annex, in order of increasing complexity, include simple tiling, pyramidal tiling and composite tiling.

E.2 Simple tiling

Simple tiling is used only to divide a large image into multiple sub-image tiles (see Figure E.1). These non-overlapping, contiguous sub-images have the same maximum dimensions, with the possible exception of tiles that fall on the bottom and right of the source image. All tiles are restricted to having the same component IDs, the same sampling factors, and must be encoded with the same coding process. Other parameters (i.e. quantization table and Huffman tables) may be changed for each tile. The tiles are placed into the image compression data stream in raster order. As an example, the tiles shown in Figure E.1 are placed into the image compression data stream as shown in Figure E.2. Tables E.1 and E.2 show the values of variable parameters for simple tiling, in the DTI and DTT marker segments respectively.

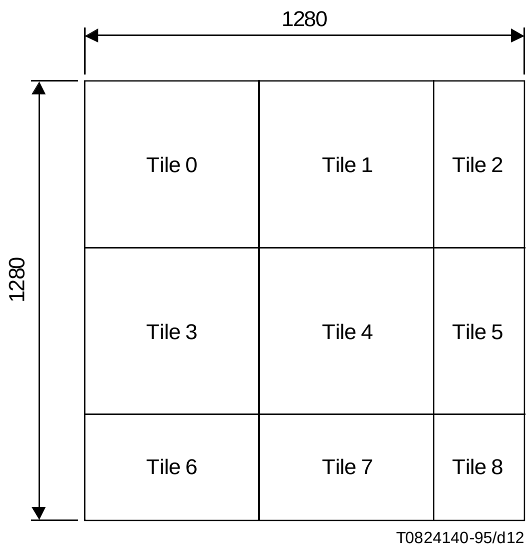

Figure E.1 – Simple tiling example of a 1280 × 1280 image transformed into 9 tiles (mostly 512 × 512)

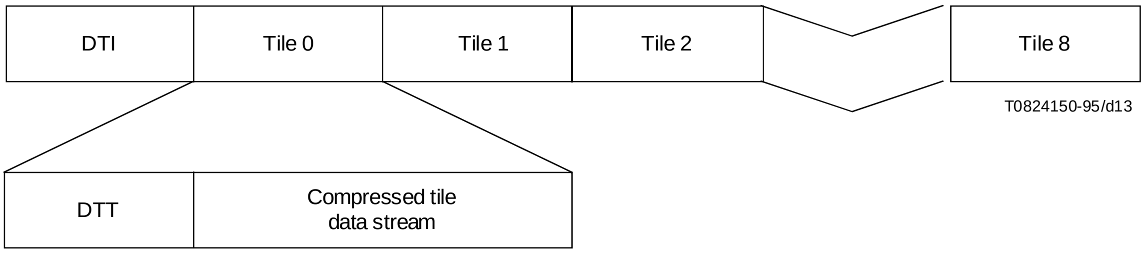

Figure E.2 – The ordering of the tiles in the compressed image data file for the example in Figure E.1

Table E.1 – Values of the variable parameters in the DTI marker segment for simple tiling

Define tiled image	
TT	0 (Simple tiling)
TIvs	1
TIhs	1
RGvs	The maximum vertical dimension of the original image
RGhs	The maximum horizontal dimension of the original image

Table E.2 – Values of the variable parameters in the DTT marker segment for simple tiling

Define tile	
TFvs	1
TFhs	1
TFvo	The maximum distance, in lines, between the top border of the tile component array and the top border of the same component array of the tiled image (see Figure E.3)
TFho	The maximum distance, in samples, between the left border of the tile component array and the left border of the same component array of the tiled image (see Figure E.3)

E.3 Pyramidal tiling

Pyramidal tiling is used to store multiple resolutions of an image as well as to divide a given resolution level into multiple sub-image tiles (see Figure E.4). Tiles in one resolution level may overlap tiles in other resolution levels but do not overlap tiles in the same resolution level. Within a given resolution level all tiles adhere to the simple tiling specifications (see E.2). All tiles of an image version with smaller maximum dimensions (lower resolution level) are placed into the compressed image data stream before the tiles of an image version with larger maximum dimensions. Within a given resolution level the tiles are placed in raster order, left-to-right and top-to-bottom (in numerical order for the example in Figure E.4). Tables E.3 and E.4 show the values of variable parameters for pyramidal tiling, in the DTI and DTT marker segments respectively.

The ratio between the maximum dimensions of the different image versions and the maximum dimensions of the original image must be integers. It must also be consistent within a given resolution level. The maximum dimensions of the tiled image at the highest resolution level must be equal to the maximum dimensions of the original image.

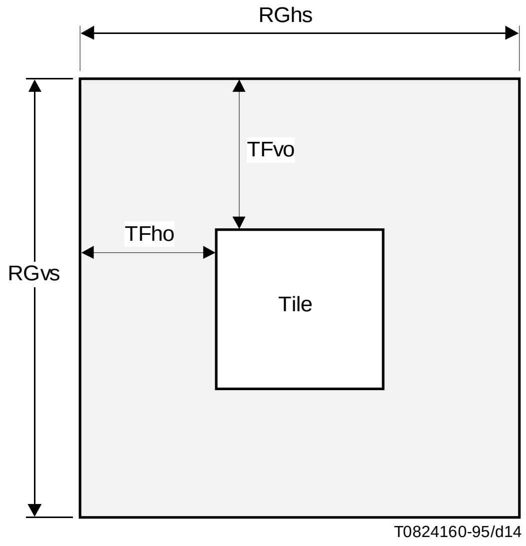

Figure E.3 – A geometric representation of the horizontal and vertical offsets for a given tile

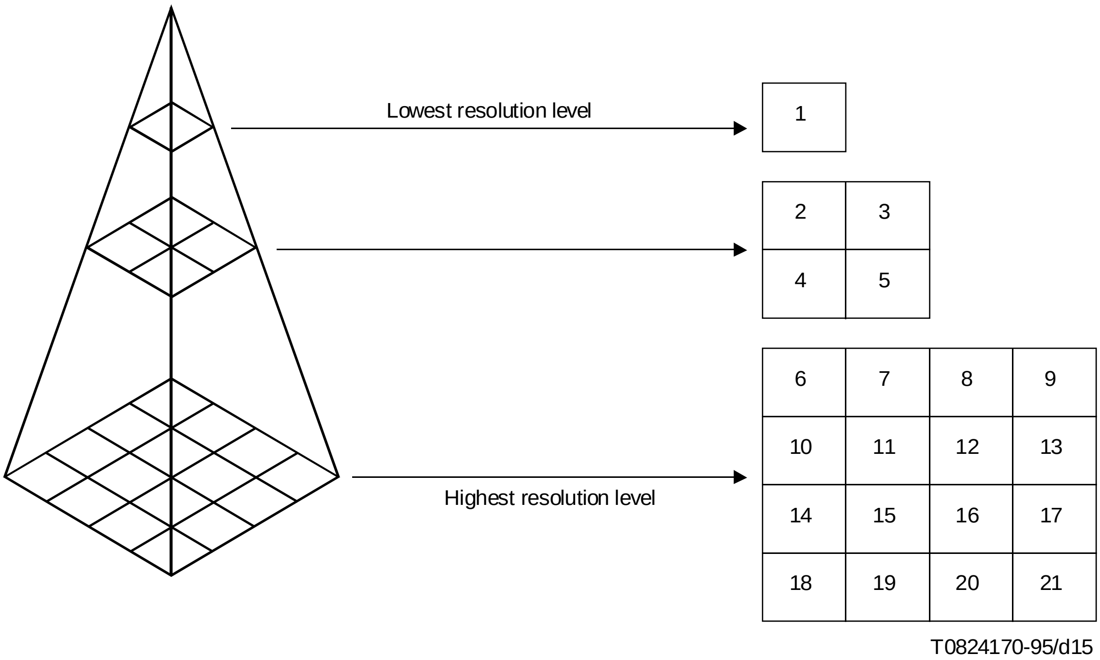

Figure E.4 – An example of multiple resolution, overlapping tiles used in pyramidal tiling

Table E.3 – Values of the variable parameters in the DTI marker segment for pyramidal tiling

Define tiled image	
TT	1 (pyramidal tiling)
TIvs	1
TIhs	1
RGvs	The maximum vertical dimension of the original image
RGhs	The maximum horizontal dimension of the original image

Table E.4 – Values of the variable parameters in the DTT marker segment for pyramidal tiling

Define tile	
TFvs	The integer ratio between the maximum vertical dimension of the original image, and the maximum vertical dimension of the tiled image to which this tile belongs
TFhs	The integer ratio between the maximum horizontal dimension of the original image, and the maximum horizontal dimension of the tiled image to which this tile belongs
TFvo	The maximum distance, in lines, between the top border of the tile component array and the top border of the same component array of the original image
TFho	The maximum distance, in samples, between the left border of the tile component array and the left border of the same component array of the original image

E.4 Composite tiling

Composite tiling allows multiple resolutions on a single display image plane. A higher resolution reference grid is used so that multiple resolution image tiles can be combined without resampling of a tile. The only restriction in tile grid width, height, ratios and offsets is that the up-sampled tile grid will fall completely within the reference grid. Tables E.5 and E.6 show the values of variable parameters for composite tiling, in the DTI and DTT marker segments respectively.

Table E.5 – Values of the variable parameters in the DTI marker segment for composite tiling

Define tiled image	
TT	2 (composite tiling)
TIvs	The integer ratio of the reference grid height to the image grid height
TIhs	The integer ratio of the reference grid width to the image grid width
RGvs	Reference grid height
RGhs	Reference grid width

Table E.6 – Values of the variable parameters in the DTT marker segment for composite tiling

Define tile	
TFvs	The integer ratio of the reference grid height to the tile grid height
TFhs	The integer ratio of the reference grid width to the tile grid width
TFvo	The distance, in reference grid units, between the top border of the reference grid and the top border of the up-scaled tile grid (Figure E.6)
TFho	The distance, in reference grid units, between the left border of the reference grid and the left border of the up-scaled tile grid (Figure E.6)

The first version number marker segment in the compressed data stream shall describe the capability needed to decode the first tile. If succeeding tiles require different capabilities, then additional version number marker segments shall be encoded immediately following the DTT marker segments of those tiles requiring different capabilities.

After decoding, each tile component is placed on the tile grid using the parameters of the preceding DCR marker segment. The tile grid is then up-sampled and placed over the reference grid according to the DTT marker segment parameters, replacing, in each reference grid element, any previously decoded data.

For the example in Figure E.5, multiple images with different resolution and size are compressed and stored as tiles, i.e. one 1024×1024 image at 200 pels/25.4 mm (abbreviated to PPI), one 1024×1024 image at 600 PPI and one 512×500 image at 100 PPI. Composite tiling would allow for the displaying of all the tiles on one display plane, without resampling. The TFvs and TFhs would be 3 for tile 1, 6 for tile 2 and 1 for tile 3. Figure E.5 shows a possible collage of these different resolution images, with TIvs and TIhs with the value 1. In this example tile 3 would be placed last in the compressed data stream.

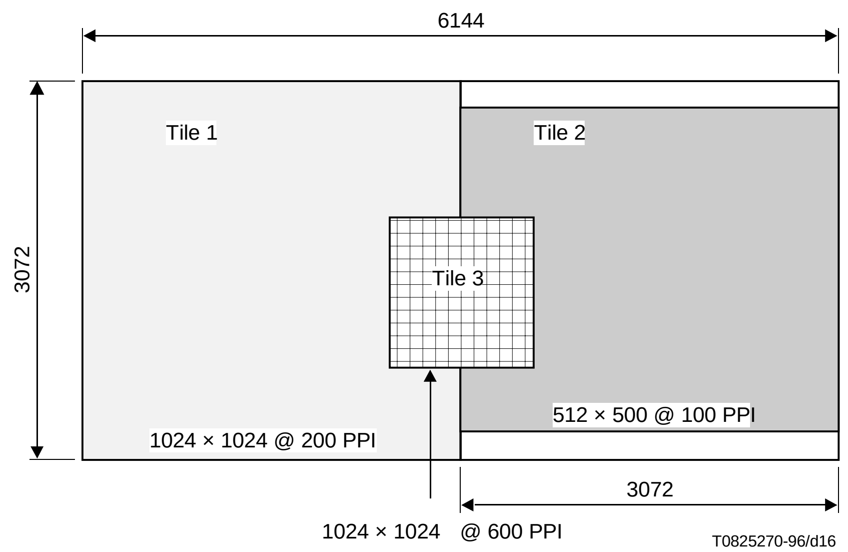

Figure E.5 – An example of multiple resolution tiles collaged into a single display image by the use of composite tiling

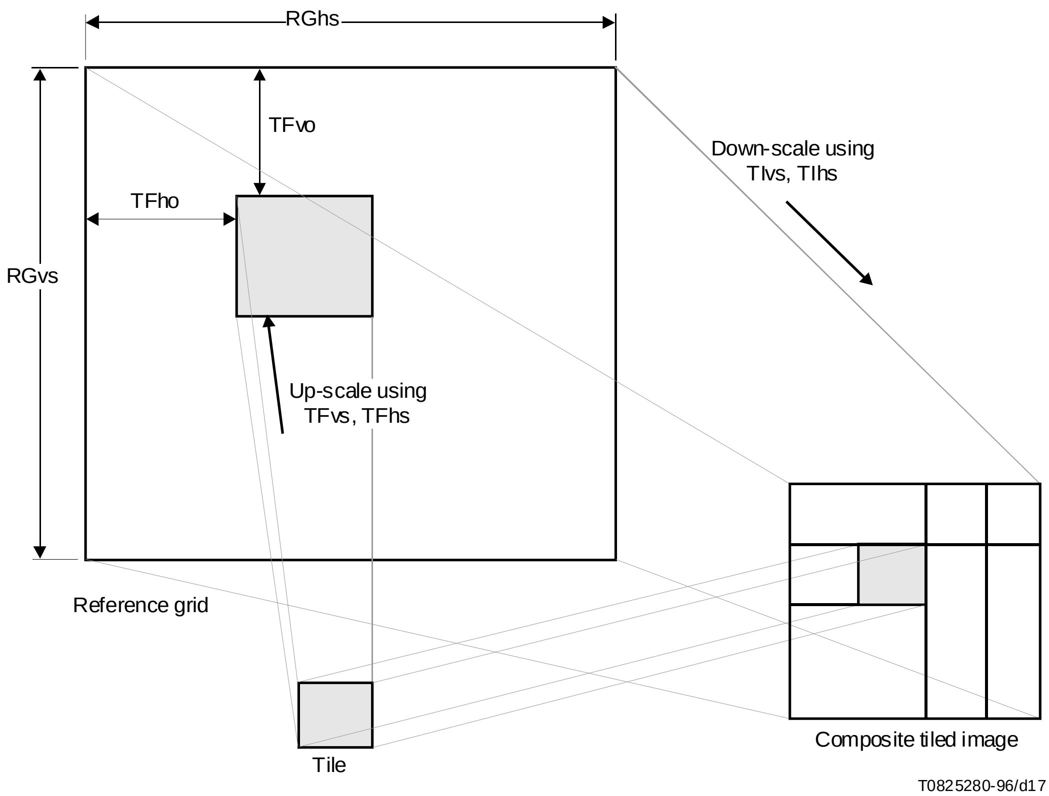

Figure E.4 – An example of the geometric relationship between the tile, the reference grid and the composite tiled image

Annex F

Still Picture Interchange File Format (SPIFF)[2)]

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

This annex specifies a file format that can be used for the interchange of image files, containing compressed image data, between application environments. This Still Picture Interchange File Format (SPIFF) is intended to be a generic format that is simple in nature and does not include many of the features found in application specific file formats.

SPIFF files may contain image data for bi-level or continuous-tone (grayscale or colour) images. Several different standard compression algorithms are supported: MH, MR, MMR, JBIG, and JPEG (See F.1.2.3). In addition to the image data, SPIFF includes information necessary to render it on common output devices, within the constraints imposed by that device.

NOTE – The Still Picture Interchange File Format is designed to incorporate functionality found in some (ad hoc) image file formats (such as JFIF) that encapsulate compressed image data streams. It is intended that transcoding between any of these file formats and SPIFF should be simple.

F.1 General aspects of the SPIFF specification

Throughout this Recommendation | International Standard a "file" is assumed to be a representation of an array of bytes of arbitrary length. Commonly the data contained in this "file" is transmitted over a telecommunication network or stored in a computer system's hard disk storage, but the actual storage location is irrelevant to this Recommendation | International Standard. It is the purpose of this Recommendation | International Standard to impose a higher level structure on the bytes in this array and to specify the interpretation of the values in those higher level structures. All constituent parts of the file format shall be represented by byte-aligned data.

F.1.1 Constituent parts

This subclause gives a general description of each of the constituent parts of the data contained in the file format (see Figure F.1).

F.1.1.1 File header

The file header is the first data that appears in the file and serves to identify the file's contents as SPIFF data. The header also contains information about the image such as the application profile, number of components, and image dimensions.

NOTE – The definition of the header is such that SPIFF files are backward compatible, i.e. if a SPIFF file is supplied to most of the currently known commercial and public domain implementations of decoders that read compressed image data, as specified in CCITT Rec. T.81 | ISO/IEC 10918-1 and this Recommendation | International Standard, they are likely to be able to successfully decode the interchange format data stream (without using any of the other information in the SPIFF file).

F.1.1.2 Directory

The directory is a sequence of directory entries. The directory contains, or contains references to, information necessary to accurately render decoded image data, or contains, or contains references to, ancillary information accompanying the image data.

F.1.1.3 Direct and indirect data

Directory entries may contain "direct" data, or may refer to "indirect" data. Direct data is typically used if the amount of data is small and fits within the directory entry (less than 65 528 bytes). If the data for a particular directory entry is too large to fit as direct data, the entry shall contain a reference to the indirect data. This reference shall be in the format of a 32-bit unsigned integer parameter that has a value equal to the offset, in bytes, from the start of the file to the indirect data. The first byte of the file is denoted by an offset of zero.

F.1.1.4 Image data

Every SPIFF file shall contain image data and optionally may contain ancillary data associated with the image. The image may be represented by compressed or uncompressed image data. This data, in combination with some of the information contained in the directory, is what is necessary to accurately render the image on any given output device.

[2)] Users of this Recommendation | International Standard may freely reproduce the SPIFF in this annex so that it can be used for its intended purpose.

Ancillary data in the file may include one or more “thumbnail” image representations, each of which may optionally be represented by a compressed image data stream and, consequently, more than one of these data streams may be present in any given SPIFF file.

F.1.2 Application profile identifier

The SPIFF file header may contain a profile identifier which specifies the application profile required to interpret the contents of the SPIFF file. The profile ID makes it unnecessary for an application-specific implementation to support the full range of parameter values defined in this annex. The profile identifier is placed in the file header so that decoders can determine the content of the file before reading the complete directory.

F.1.2.1 Continuous-tone base profile

This profile specifies that the image is represented by a compressed data stream encoded by a subset of the baseline process of CCITT Rec. T.81 | ISO/IEC 10918-1 (JPEG), and is defined by the following:

- The compression type (“C” parameter of the file header) shall be 5 (JPEG). The compressed data stream shall be encoded with the baseline process and shall contain a single scan, i.e. if more than one component is present, the components shall be interleaved.
- The colour space (“S” parameter of the file header) shall be 3 or 8.
- The image orientation directory entry is not present.
- The use of indirect data is not allowed.

F.1.2.2 Continuous-tone progressive profile

This profile provides for low-speed communication applications, especially on low speed networks (PSTN, Mobile) in connection with conversational multimedia type of services, such as a simple Still Image Transmission Mode of Videophones. It extends the continuous-tone base profile by also supporting the following coding processes (see Table B.4):

- 8 bits, Huffman, spectral selection (capability indicator value, $CAP_0 = 6$);
- 8 bits, Huffman, full progression (capability indicator value, $CAP_0 = 8$),

in addition to the baseline sequential process (capability indicator value, $CAP_0 = 0$).

F.1.2.3 Bi-level facsimile profile

This profile is used for Group 3 and Group 4 bi-level facsimile images compressed according to Rec. T.4 (Modified Huffman – MH and Modified READ – MR), Rec. T.6 (Modified Modified READ – MMR), or Rec. T.85 (which refers to ITU-T Rec. T.82 | ISO/IEC 11544 (JBIG)), and is defined by the following:

- the compression type (“C” parameter of the file header) shall be 1, 2, 3 or 4;
- the colour space (“S” parameter of the file header) shall be 0;
- the bits per sample (“BPS” parameter of the file header) shall be 1;
- the use of indirect data is not allowed.

F.1.2.4 Continuous-tone facsimile profile

This profile applies to the representation of continuous-tone (multi-level) colour and gray-scale images for Group 3 and Group 4 facsimile as specified in Recommendations T.4, T.30, and T.503, and is defined by the following:

- the compression type (“C” parameter of the file header) shall be 5;
- the colour space (“S” parameter of the file header) shall be 14;
- the bits per sample (“BPS” parameter of the file header) shall be 8 or 12;
- the use of indirect data is not allowed.

F.1.3 Syntax description

For the purposes of this Recommendation | International Standard, the syntax specification consists of:

- the required ordering of constituent parts;
- identification of required, optional or conditional constituent parts;
- name and definition of each possible parameter and the allowed values of each parameter;
- any restrictions on the above which are specific to the contents of the contained Interchange format data stream(s).

F.1.3.1 Parameter conventions

Parameter type is identified by one of the symbols "I.", "B.", "F.", or "S." (identifying respectively: unsigned integer, byte, fixed point, and string). For type integer, the size is indicated by following the symbol "I." by a number indicating the number of bits in the parameter. This number shall be 8, 16 or 32, indicating single byte, double byte, or quadruple byte unsigned integers. Multiple byte integers are stored with the most significant byte first. Type "B." is used only for filler type fields (to guarantee alignment) and reserved fields. A number immediately following the symbol "B." indicates the number of consecutive bytes occupied by the parameter.

Parameters whose type is indicated by the symbol "F." are 4-byte parameters in "fixed point" notation. The 16 most significant bits are essentially the same as a parameter of type I.16 and indicate the integer part of this number. The 16 least significant bits are essentially the same as an I.16 parameter and contain an unsigned integer that, when divided by 65536, represents the fractional part of the fixed point number. Fixed-length string parameters, indicated by the symbol "S.", are to be interpreted as characters from ISO/IEC 8859-1. The number of bytes in the string is indicated by the number following the "S.". Variable-length string parameters are described by F.2.3.2.1.

F.2 High-level syntax

Figure F.1 specifies the order of high level constituent parts of the interchange file format. A more specific example using the image coding specified in CCITT Rec. T.81 | ISO/IEC 10918-1 is given in H.4.

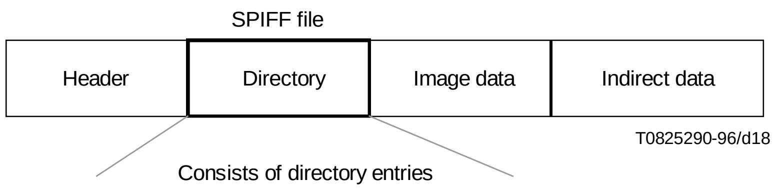

Figure F.1 – High-level syntax for the still picture interchange file format

The block labeled "indirect data" is optional and, if present, consists of one or more individual indirect data items corresponding to directory entries in the directory.

F.2.1 File header syntax

Figure F.2 specifies the syntax of the SPIFF file header, which shall be present at the start, i.e. offset zero, of every SPIFF file. This header contains some parameters that make it possible to quickly recognize a file to be a SPIFF file (by inspecting the first few bytes of the header), as well as parameters that give basic information about the image. Finally, the header contains a parameter that indicates the version of the oldest SPIFF format specification that this file conforms to.

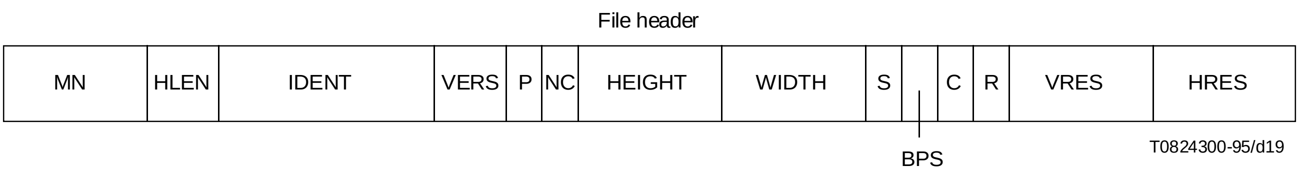

Figure F.2 – SPIFF file header syntax

The parameters shown in Figure F.2 are defined below. The size and allowed values are defined in Table F.1.

– **MN:** Magic Number – This is a number that is unique enough to distinguish the type of this file from that of many other files by just looking at these four bytes. The value of this parameter is fixed, see Table F.1.

– **HLEN:** Header Length – This parameter is the length of the file header in bytes, minus 4 (that is, MN is not included in HLEN).

– **IDENT:** An additional identifier that contributes to the uniqueness of the header – The value of this parameter is fixed and chosen to correspond to the sequence of characters "SPIFF" when interpreted using ISO/IEC 8859-1, see Table F.1.

– **VERS:** This parameter identifies the version number of this SPIFF specification that the file complies with. The parameter is defined as a two-byte integer with the most significant byte containing the major version number (currently defined as 1) and the least significant byte containing a minor revision number (currently defined as 0).

  A major version number increment (if there ever is one) represents an incompatible change in SPIFF files. Decoders should give up if they encounter an unrecognized major version number. Minor version number increments represent backwards compatible changes. Decoders should continue to process SPIFF files even if the minor version number is unrecognized.

– **P:** Profile ID – This parameter identifies the application profile which must be supported to read the SPIFF file. The allowed values are: 0 = no profile specified; 1 = continuous tone base profile; 2 = continuous tone progressive profile; 3 = bi-level facsimile profile; and 4 = continuous-tone facsimile profile.

– **NC:** Number of components – This parameter specifies the number of colour components in the image.

– **HEIGHT:** Image height – The value of this parameter indicates the number of lines in the highest component of the image.

– **WIDTH:** Image width – The value of this parameter indicates the number of samples per line in the widest component of the image.

– **S:** Colour space – This parameter specifies the colour space in which the sample values define coordinates. The order in which components are specified in the compressed image data stream shall correspond to the order established by the name of colour space. See the following subclause for a specification of the values of this parameter.

– **BPS:** Bits per sample – This parameter specifies the number of bits per sample for the components of the image. The allowed values are shown in Table F.1.

– **C:** Compression type – Specifies the compression algorithm used to compress the image data:

  0 = Uncompressed – Picture data is stored in component interleaved format, encoded at BPS per sample. When BPS is not 8, sample values shall be packed into bytes so that no bits are unused between samples. However, each scan line shall begin on a byte boundary, and padding bits having value 0 (zero) shall be inserted after the last sample of a scan line as necessary to fill out the last byte of the scan line. Sample values appear in component-interleaved order. When multiple sample values are packed into a byte, the first sample shall appear in the most significant bits of the byte. When a sample is larger than a byte, its most significant bits shall appear in earlier bytes.

  1 = Recommendation T.4, the basic algorithm commonly known as MH (Modified Huffman). This value is only permitted for bi-level images.

  2 = Recommendation T.4, commonly known as MR (Modified READ). This value is only permitted for bi-level images.

  3 = Recommendation T.6, commonly known as MMR (Modified Modified READ). This value is only permitted for bi-level images.

  4 = ITU-T Rec. T.82 | ISO/IEC 11544, commonly known as JBIG. This value is only permitted for bi-level images.

5 = CCITT Rec. T.81 | ISO/IEC 10918-1 or ITU-T Rec. T.84 | ISO/IEC 10918-3, commonly known as JPEG. The compressed image data stream shall conform to the syntax of interchange format for compressed image data as specified in the aforementioned standards. This value is only permitted for continuous-tone (grayscale or colour) images.

– **R:** Resolution units – Specifies the units in which the vertical and horizontal resolutions are expressed. Both resolutions shall be specified using the same units. A value of 1 specifies units of dots/samples per inch, a value of 2 indicates dots/samples per centimetre. A value of 0 specifies that an aspect ratio is to be defined and the values for horizontal and vertical resolutions are to be interpreted as I.32 unsigned quantities rather than fixed point number. In this case, the two numbers define the aspect ratio of the samples, i.e. the width of a sample, divided by the height of a sample.

– **VRES:** Vertical resolution – Specifies vertical resolution as a fixed point number in the units indicated by the R parameter, unless R is set to the value 0, in which case this parameter describes the numerator of a fraction that is the aspect ratio of the samples. A value for VRES of 0 is not permitted.

– **HRES:** Horizontal resolution – Specifies horizontal resolution as a fixed point number in the units indicated by the R parameter, unless R is set to the value 0, in which case this parameter describes the denominator of a fraction that is the aspect ratio of the samples. A value for HRES of 0 is not permitted.

NOTE – If vertical or horizontal resolutions are not known, R should be set to 0, and VRES and HRES both set to 1 to indicate that pixels in the image should be assumed to be square.

Table F.1 – SPIFF file header parameter sizes and values

Parameter	Type, size	Values
MN	I.32	X'FFD8FFE8'
HLEN	I.16	32
IDENT	S.6	X'535049464600'
VERS	I.16	X'0100'
P	I.8	0 - 4
NC	I.8	1 - 255
HEIGHT	I.32	1 - 4, 294, 967, 295
WIDTH	I.32	1 - 4, 294, 967, 295
S	I.8	0 - 15
BPS	I.8	1, 2, 4, 8, 12, 16
C	I.8	0 - 5
R	I.8	0 - 2
VRES	F / I.32	1 - 4, 294, 967, 295
HRES	F / I.32	1 - 4, 294, 967, 295

F.2.1.1 Allowed values for the S (colour space) parameter

This parameter identifies some well known and often used colour spaces that are perhaps not always very well defined. The values given below shall give such a definition. If an encoder does not produce/compress data in exactly one of these colour spaces, a value of 2 shall be used and applications are advised to use application specific directory entries to give further specifications.

S = 0 Bi-level – This value shall be used to indicate bi-level images. Each image sample is one bit: 0 = white and 1 = black.

S = 1 YC_bC_r(1) – This is a format often used for data that originated from a video signal. The colour space is based on Recommendation ITU-R BT.709. The valid ranges of the YC_bC_r components in this space is limited to less than the full range that could be represented given an 8-bit representation. Recommendation ITU-R BT.601-1 specifies these ranges as well as defines a 3×3 matrix transform that can be used to convert these samples into RGB.

S = 2 This value indicates that the colour space interpretation of the coded sample components is none of the interpretations specified in this subclause.

S = 3 YC_bC_r(2) – This is the most commonly used format for image data that was originally captured in RGB (uncalibrated format). The colour space is based on Recommendation ITU-R BT.601-1. The valid ranges of the YC_bC_r components in this space is [0,255] for Y, and [–128,127] for C_b and C_r (stored with an offset of 128 to convert the range to 0-255). These ranges are different from the ones defined in Recommendation ITU-R BT.601-1. Recommendation ITU-R BT.601-1 specifies a 3×3 matrix transform that can be used to convert these samples into RGB.

S = 4 YC_bC_r(3) – This is a format often used for data that originated from a video signal. The colour space is based on Recommendation ITU-R BT.601-1. The valid ranges of the YC_bC_r components in this space is limited to less than the full range that could be represented given an 8-bit representation. Recommendation ITU-R BT.601-1 specifies these ranges as well as defines a 3×3 matrix transform that can be used to convert these samples into RGB.

S = 5 Reserved.

S = 6 Reserved.

S = 7 Reserved.

S = 8 Grayscale – This is a single component sample with interpretation as grayscale value (luminance only). This value should be used for images having number of bits per sample greater than or equal to two. A value of 0 indicates minimum intensity, and a value of 2^{BPS} –1 indicates maximum intensity.

S = 9 PhotoYCC – This is the colour encoding method used in the Photo CD™ system. The colour space is based on Recommendation ITU-R BT.709 reference primaries. Recommendation ITU-R BT.709 linear RGB image signals are transformed to non-linear R'G'B' signals. Values for RGB may be either positive or negative. For positive values, the non-linear transformation corresponds to the opto-electronic transfer characteristics defined in Recommendation ITU-R BT.709. Equations for transforming R'G'B' values to YCC correspond to Recommendation ITU-R BT.601-1. Details of this encoding method can be found in Kodak Photo CD Products, *A Planning Guide for Developers*, Eastman Kodak Company, Part No. DC1200R and also in Kodak Photo CD Information Bulletin PCD045.

S = 10 RGB – The encoded data consists of samples of (uncalibrated) R, G and B data, directly suitable for display on typical RGB devices. For each component, a value of 0 indicates minimum intensity, and a value of 2^{BPS} – 1 indicates maximum intensity.

S = 11 CMY – The encoded data consists of samples of Cyan, Magenta and Yellow samples, directly suitable for printing on typical CMY devices. A value of 0 shall indicate 0% ink coverage, whereas a value of 2^{BPS} –1 shall indicate 100% ink coverage for given component sample.

S = 12 CMYK – As CMY above, except there is also a black (K) ink component. Ink coverage is defined as above.

S = 13 YCCK – This is the result of transforming original CMYK type data by computing R = (2^{BPS} – 1) – C, G = (2^{BPS} – 1) – M, and B = (2^{BPS} – 1) – Y, applying the RGB to YCC transform specified for S = 3, and then recombining the result with the unmodified K-sample value.

NOTE 1 – This transform is intended to be the same as that specified in Adobe PostScript.

S = 14 CIELab – The CIE 1976 (L* a* b*) colour space. A colour space defined by the CIE (Commission Internationale de l'Eclairage), having approximately equal visually perceptible difference between equally spaced points throughout the space. The three components are L*, or Lightness, and a* and b* in chrominance. Default version as defined in Recommendation T.42.

S = 15 Bi-level – This value shall be used to indicate bi-level images. Each image sample is one bit: 1 = white and 0 = black.

NOTE 2 – The value encoded in the S parameter does not imply that when the original samples were captured they were represented in the same colour space. Quite often encoders will decide, in order to achieve greater compression performance, to apply some colour space transformation to the samples before encoding. A good example is original data in the RGB colour space which is almost always transformed into the $YC_bC_r(2)$ colour space before encoding.

F.2.2 Directory syntax

Figure F.3 specifies the syntax of the directory format. The EOD entry is mandatory even if no other directory entries are present.

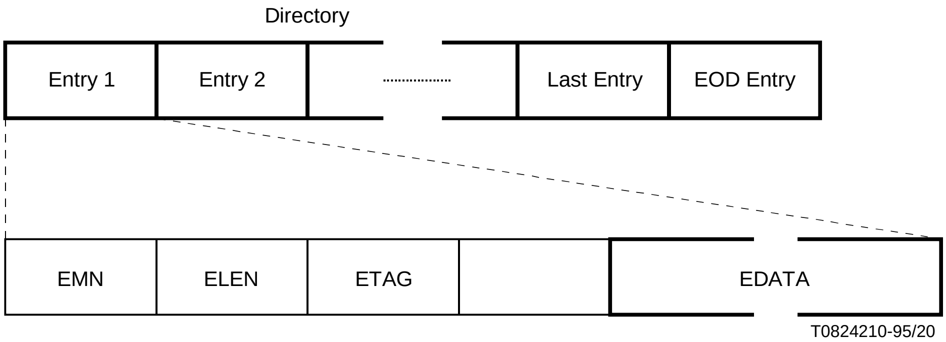

Figure F.3 – Directory syntax

The parameters shown in Figure F.3 are defined below. The size and allowed values are defined in Table F.2.

- **EMN:** Entry Magic Number – This two-byte code signals the start of a directory entry. The value of this parameter is fixed, see Table F.2.
- **ELEN:** Entry Length – This parameter is the length of the directory entry in bytes, minus 2 (that is, EMN is not included in ELEN).
- **ETAG:** Uniquely defines each set of logically related pieces of information about the image, or ancillary information, that is stored either within the entry (in the EDATA field), or is found in the indirect data part of the file. The value of this parameter is decomposed into several groups of bits. The 8 most significant bits are reserved and must be zero. Of the following 24 bits that make up the tag identification, the 3 most significant bits are used to subdivide the available range of values for tags identification into 5 separate ranges, each of which is assigned to a particular standards body or are assigned for application use, see F.2.2.2. The rest of the bits in the tag are defined by the respective standards body or application. This Recommendation | International Standard, and possible future extensions, shall only define ETAG values that use the value 0 (zero) for these 3 bits.
- **EDATA:** Contains data specific for this ETAG. This data has a format specific for the corresponding ETAG value (for specifics see the definitions of the possible ETAG values). In some cases the EDATA field will contain nothing but an offset to the “real” data stored in one of the indirect data blocks.

Table F.2 – Directory parameter sizes and values

Parameter	Type, size	Values
EMN	I.16	X’FFE8’
ELEN	I.16	8 - 65534
ETAG	I.32	0 - 16, 777, 215
EDATA	Varies	defined by ETAG

The minimum entry size is 8 bytes, for an entry with no associated data. It is required that each directory entry occupies a multiple of 4 bytes. Entries that at first do not meet this requirement should be made to adhere to this rule by adding 1, 2 or 3 filler bytes to the EDATA parameter. Alignment restrictions for individual pieces of indirect data may cause one or more "filler" bytes to exist in between two such indirect items where none of these bytes is described by, or belongs to, any particular directory entry. All bits in such bytes shall be set to zero.

Applications that decode the still picture interchange file format shall deal with all possible ETAG values. Any directory entry that is encountered with an unknown (by the decoder) ETAG value shall be ignored and skipped using the ELEN parameter value.

The term "Tag value" is used hereafter to refer to the value of the ETAG parameter.

F.2.2.1 Directory entry length specification

The ELEN parameter allows applications that do not recognize certain tags to skip directory entries and continue with a subsequent entry. A value of 'n' for the ELEN parameter indicates a directory entry with n + 2 bytes. Consequently directory entries can vary in size from 8 to 65536 (0-65528 bytes of EDATA) bytes.

Therefore directory entries are allowed to contain up to 65528 bytes of "direct" data. If more space is needed for the parameters of a particular directory entry, this must be done through the use of "indirect" data, i.e. the direct data part of the entry should contain at least one I.32 type parameter containing the offset in the file to the indirect data. This indirect data can then be defined to have any appropriate format, as there are essentially no size restrictions for indirect data.

F.2.2.2 Directory standards body specification

The 3 bits following the 8 most significant bits (bits 23:21) in the ETAG value are used to define the "originating standards body". The values are assigned as follows:

- 0 - 3 ISO/IEC and common text generic standards – All entries defined in this Recommendation | International Standard shall use this originator indication.
- 4 ISO application standards – Entries with tag values having the originator bits set to this value are defined in ISO application standards.
- 5 ITU-T – Entries with tag values having the originator bits set to this value are defined in ITU-T Recommendations.
- 6 National standards bodies – Entries with tag values having the originator bits set to this value are defined by the various national standards bodies. There shall be 10 bits immediately following these three bits that shall indicate what country is responsible according to the numeric version of the country codes as specified in ISO 3166:1993.
- 7 Other – This part of the total code space for the directory entry tag value is available for application specific use (see F.2.3.1).

F.2.2.3 End of directory

A special directory entry, the EOD entry, is used to signal the end of the directory. This entry is mandatory, even if no other directory entries are present. No additional directory entries may follow the EOD entry. The tag is followed immediately by compressed image data. Table F.3 describes the size of and allowed values for the parameters of this entry.

- **EMN:** Entry Magic Number – This two-byte code signals the start of a directory entry.
- **EODLEN:** EOD Entry Length – This parameter is the exact length of the EOD entry in bytes. Note that the EOD entry length is defined differently from the lengths of other directory entries (ELEN).
- **EODTAG:** Identifies the EOD entry.

Table F.3 – End of directory parameter sizes and values

Parameter	Type, size	Values
EMN	I.16	X'FFE8'
EODLEN	I.16	8
EODTAG	I.32	1

F.2.3 Specific directory entry definitions

This subclause lists all currently defined directory entries. Each of these entries has a unique ETAG value and each subclause defining such an entry shall also specify the format of the corresponding EDATA entry.

F.2.3.1 Application specific directory entries

In order to make this file format as flexible as possible, a provision has been made that allows specific applications to add information to a SPIFF file that could not be described using the tag values defined in this Recommendation | International Standard. It should be noted, however, that such use is application specific and other applications may not recognize these entries. Unrecognized application specific tags should be skipped over and ignored.

Application specific directory entries are those that have the 3 bits immediately following the 8 most significant bits (bits 23:21) set to all 1's. All other tag values are reserved for standards bodies (see F.2.3.2).

NOTE – It is advisable for any application that decides to use these application specific tags to make sure that the EDATA field for such entries contains a value that further uniquely identifies this use of the tag to best of the application's knowledge. Such use should reduce the probability of incorrect interpretation by other applications.

F.2.3.2 Standard directory entries

All entries with tag values other than the application specific tag values defined in F.2.3.1, are reserved for use by ISO, ITU-T or national standards bodies. Several of these entries are currently defined and their specifications can be found in the following subclauses.

F.2.3.2.1 Common representation of string parameters

All standard directory entries use a common representation for string parameters. This representation allows strings to be stored as direct or indirect data and specifies the character set used to interpret the character data. All character strings are terminated with a single byte with all bits set to zero (null byte terminated). Following the terminating byte, null bytes shall be added as necessary to pad to a 4-byte boundary. Table F.4 describes the size of and allowed values for a generic example of the parameters of this entry.

– **STRLOC:** String location – If the value of this parameter is zero, the string is stored as direct data immediately following the CHARSET parameter. If non-zero, the string is stored as indirect data and the value of the parameter is the string's starting offset.

– **CHARSET:** Specifies a character set to be used to interpret the bytes stored in any string type parameter for the purpose of display in human readable form. A value of 0 (zero) is not allowed. A value of *N* indicates interpretation using the code tables defined by the ISO/IEC 8859-*N* standard. A value of 254 indicates interpretation according to Recommendation T.51. A value of 255 indicates interpretation according to ISO/IEC 10646 (also known as Unicode), a representation that allows for international multi-byte characters. The allowed values for *N* are determined by the existence of the corresponding ISO/IEC 8859 standard. (See the examples and guidelines in Annex H).

NOTE – If the line feed character (X'0A') is encountered, it should be treated as the "new line" function. Use of all characters with a value < X'20' should be avoided.

Table F.4 – String parameter sizes and values

Parameter	Type, size	Values
STRLOC	I.32	0, or in range from EOI marker offset to 4, 294, 967, 295
CHARSET	I.8	1 to N, where N is largest existing ISO/IEC 8859-N, 254, 255

Table F.5 lists the tags, and their values, which are defined in this subclause.

Table F.5 – Tags defined in this Recommendation | International Standard

Tag name	Values
Transfer characteristics	X’00000002’
Component registration	X’00000003’
Image orientation	X’00000004’
Thumbnail	X’00000005’
Image title	X’00000006’
Image description	X’00000007’
Time stamp	X’00000008’
Version identifier	X’00000009’
Creator identification	X’0000000A’
Protection indicator	X’0000000B’
Copyright information	X’0000000C’
Contact information	X’0000000D’
Tile index	X’0000000E’
Scan index	X’0000000F’
Set reference	X’00000010’

F.2.3.2.2 Tag – Transfer characteristics

This entry describes the opto-electronic transfer characteristics of the source image. Table F.6 describes the size of and allowed values for the parameters of this entry.

– **TRANCHAR:** An 8-bit integer which describes the opto-electronic transfer characteristics (gamma correction) of the source image. If this entry is applicable for the value of the S (colour space) parameter in the file header and does not appear in the directory, a default value of 1 is assumed.

This entry shall appear at most once in the directory, and only when parameter C (compression type) in the file header has the value 5.

Table F.6 – Transfer characteristics

Transfer characteristics			Tag value: X’00000002’
Offset	Parameter	Type, size	Values
0	TRANCHAR	I.8	1 - 8
1	RESERVED	B.3	0

F.2.3.2.2.1 Allowed values for the TRANCHAR parameter

This parameter identifies well known standard transfer characteristics. The allowed values for this parameter are defined below:

– TRANCHAR = 1 Recommendation ITU-R BT.709.

– TRANCHAR = 2 Unspecified. Image characteristics are unknown.

– TRANCHAR = 3 Reserved.

– TRANCHAR = 4 Recommendation ITU-R BT.470-3 System M. Assumed display gamma = 2.2.

– TRANCHAR = 5 Recommendation ITU-R BT.470-3 System B, G. Assumed display gamma = 2.8.

– TRANCHAR = 6 SMPTE 170M.

– TRANCHAR = 7 SMPTE 240M.

– TRANCHAR = 8 Linear transfer characteristics.

F.2.3.2.3 Tag – Component registration

This entry specifies component registration, the spatial positioning of samples within components relative to the samples of other components. This entry is variable length; the number of parameters contained in this entry is given by the number of components in the image (specified by the NC parameter of the file header). Note that if the number of components is not a multiple of 4, one or more zero bytes must be appended to pad to the next 32-bit word boundary. This entry shall not be present for images having only one component. Table F.7 describes the size of and allowed values for the parameters of this entry.

- **CROFFSET$_i$:** Component registration vertical and horizontal offsets – Specifies the vertical and horizontal distances in one-half sample units to offset the current component (down and to the right). This offset is specified with respect to the grid having dimensions defined by the HEIGHT and WIDTH parameters in the file header. The vertical offset is specified in the most significant 4 bits of this parameter; the horizontal offset is specified in the least significant 4 bits. If this entry is applicable for the value of the S parameter and does not appear in the directory, a default value of 0 (zero) shall apply.

This entry shall appear at most once in the directory.

Table F.7 – Component registration

Component registration			Tag value: X'00000003'
Offset	Parameter	Type, size	Values
0	$CROFFSET_0$	I.8	0 - 255
1	$CROFFSET_1$	I.8	0 - 255
2	…		

F.2.3.2.4 Tag – Image orientation

The compressed image data commonly does not specify the order of encoding the image samples completely. For example, A.1.4 of CCITT Rec. T.81 | ISO/IEC 10918-1 mentions that encoding shall be left-to-right and top-to-bottom, but it is up to applications to define which edges of the image shall be considered left, right, top and bottom. Table F.8 describes the size of and allowed values for the parameters of this entry.

- **IMGOR:** This parameter specifies the orientation of the image, i.e. defines which edge of the image, as decoded shall be considered the top of the image for the purpose of display and rendering. The allowable values indicate a rotation in terms of multiples of 90 degrees, in clock-wise direction, that will make the image be oriented correctly after decoding and rotation. Thus, a value of 1 indicates a 90 degree rotation, 2 indicates 180 degree rotation and 3 indicates 270 degree rotation.

- **IMGFLIP:** If this parameter is set to 1 it indicates that after decoding and applying the rotation as specified by the IMGOR parameter, the image needs to have its left-to-right orientation reversed in order to be displayed correctly.

Table F.8 – Image orientation

Image orientation			Tag value: X'00000004'
Offset	Parameter	Type, size	Values
0	IMGOR	I.8	0 - 3
1	IMGFLIP	I.8	0, 1
2	RESERVED	B.2	0

If this entry is not present in the directory, the defaults shall be 0 (zero) for IMGOR and 0 (zero) for IMGFLIP, indicating that the first row of MCUs resulting from decoding shall be along the top of the resulting image. If this entry is present in the directory, it shall appear at most once.

NOTE – In most cases, images have been encoded using a fairly trivial model of this orientation issue. Generally the only issue is that of landscape versus portrait mode. Landscape mode is the most often used implementation of this orientation issue, corresponding to the normal application of typical 35 mm photography cameras. In this case, the image is larger in the horizontal dimension and IMGOR will typically be 0 (no rotation required). The other common case is portrait mode, where, using the same 35 mm camera model, this camera has been rotated 90 degree clockwise or counter-clockwise. This corresponds to IMGOR values of 1 or 3, respectively. In both scenarios above IMGFLIP would be 0.

F.2.3.2.5 Tag – Thumbnail image specification

A SPIFF file may contain a number of ancilliary images in addition to the primary compressed image data stream. All of these images shall be renditions of the primary image. The purpose of these ancilliary images is typically to supply low resolution preview images, commonly known as a “thumbnail”. Table F.9 describes the size of and allowed values for the parameters of this entry.

- **TNDATA:** This parameter specifies the offset in the file to the image data for the ancillary image. If the value of this parameter is zero, the image data for the ancillary image is stored as direct data immediately following the reserved byte at the end of the parameter list. If non-zero, the string is stored as indirect data and the value of the parameter is the string’s starting offset.

  The other parameters of this entry describe how this data is to be interpreted and used to render the ancillary image.

- **TNHEIGHT:** This parameter specifies the size of the thumbnail image in terms of the number of samples in the highest component of the thumbnail image. A value of 0 (zero) is not allowed.
- **TNWIDTH:** This parameter specifies the size of the thumbnail image in terms of the number of samples in the widest component of the thumbnail image. A value of 0 (zero) is not allowed.
- **TNS:** This parameter describes the colour space defined for the sample data comprising the thumbnail. The allowed values are identical to those defined for the S parameter in the file header. There is no requirement for the colour space defined by TNS to be the same as that defined in the file header for the primary image.
- **TNBPS:** This parameter specifies the number of bits per sample for the image components of the thumbnail image. The allowed values are shown in Table F.9. The number of bits per sample in the thumbnail image shall not be greater than the number of bits per sample defined in the file header for the primary image.
- **TNC:** This parameter specifies the compression type of the thumbnail data. The allowed values for this parameter are identical to those defined for the C parameter of the file header. There is no requirement for the compression type defined by TNC to be the same as that defined in the file header for the primary image.

Table F.9 – Thumbnail image specification

Thumbnail image specification			Tag value: X’00000005’
Offset	Parameter	Type, size	Values
0	TNDATA	I.32	Any
4	TNHEIGHT	I.16	1 - 65535
6	TNWIDTH	I.16	1 - 65535
8	TNS	I.8	0 - 14
9	TNBPS	I.8	1, 2, 4, 8, 12, 16
10	TNC	I.8	0 - 5
11	RESERVED	B.1	0
12	…		

When TNC is zero and TNBPS is not 8, sample values shall be packed into bytes so that no bits are unused between samples. However, each scan line shall begin on a byte boundary, and padding bits having value 0 (zero) shall be inserted after the last sample of a scan line as necessary to fill out the last byte of the scan line. Sample values appear in component-interleaved order. When multiple sample values are packed into a byte, the first sample shall appear in the most significant bits of the byte. When a sample is larger than a byte, its most significant bits shall appear in earlier bytes.

NOTE – It is strongly suggested that the value of TNS be set to either 3, 8, or 10. This should make it possible for applications that do not want to implement full decoders to still use thumbnails from SPIFF files.

Notice that there is no indication of the resolution (in dots per inch or centimeter) for the thumbnail. This is not necessary, as this information can be directly derived from the corresponding information for the primary image.

F.2.3.2.6 Tag – Image title

This entry describes in textual form a title for the image. Table F.10 describes the size of and allowed values for the parameters of this entry.

– **TITLELOC:** Location of a string containing textual representation of the image title – If the value of this parameter is zero, the string is stored as direct data immediately following the CHARSET parameter. If non-zero, the string is stored as indirect data and the value of the parameter is the string's starting offset.

– **CHARSET:** Specifies the character set to be used to interpret the character data (see F.2.3.2.1).

NOTE – The meaning and interpretation of the text in this entry is application specific.

Table F.10 – Image title

Image title			Tag value: X'00000006'
Offset	Parameter	Type, size	Values
0	TITLELOC	I.32	0 or in range from EOI marker offset to 4, 294, 967, 295
4	CHARSET	I.8	1 to N, where N is largest existing ISO/IEC 8859-N, 254, 255
5	...		

F.2.3.2.7 Tag – Image description

This entry refers to data in textual form containing additional descriptive information about the image contained in this file. Table F.11 describes the size of and allowed values for the parameters of this entry.

– **DESCLOC:** Location of a string containing additional descriptive material about the image – If the value of this parameter is zero, the string is stored as direct data immediately following the CHARSET parameter. If non-zero, the string is stored as indirect data and the value of the parameter is the string's starting offset.

– **CHARSET:** Specifies the character set to be used to interpret the character data (see F.2.3.2.1).

NOTE – The meaning and interpretation of the text in this entry is application specific

Table F.11 – Image description

Image description			Tag value: X'00000007'
Offset	Parameter	Type, size	Values
0	DESCLOC	I.32	0 or in range from EOI marker offset to 4, 294, 967, 295
4	CHARSET	I.8	1 to N, where N is largest existing ISO/IEC 8859-N, 254, 255
5	...		

F.2.3.2.8 Tag – Time stamp

This entry describes the date and time of the last modification of the image. The parameters of this entry are fixed-length strings which do not follow the conventions used by the string parameters of other tags. The character set used to interpret the data contained in this tag shall be that specified in ISO 8859-1. Table F.12 describes the size of and allowed values for the parameters of this entry.

- **DATE:** A string containing textual representation of the last modification date for the image – This representation is to conform to the format prescribed by the extended format of the ISO 8601 standard and is of the form YYYY-MM-DD, where YYYY specifies the year, MM specifies the month (01-12) and DD specifies the day of the month (01-31).

- **TIME:** A string containing textual representation of the last modification time for the image – This representation is to conform to the format prescribed by the ISO 8601 standard for Coordinated Universal Time (UTC) and is of the form HH:MM:SS.mmmZ. HH represents the hour (using a 24-hour time system), MM represents the minutes (00-59), and SS.mmm represents the seconds (00-59.999) to one millisecond resolution. The Z character (coded as X’5A’) indicates UTC timing.

This entry shall appear at most once in the directory.

Table F.12 – Time stamp

Time Stamp			Tag value: X’00000008’
Offset	Parameter	Type, size	Values
0	DATE	S.10	ISO 8601 format date
10	TIME	S.13	ISO 8601 format time
23	RESERVED	B.1	0 (reserved)

F.2.3.2.9 Tag – Version identifier

This entry describes in textual form a version identifier which refers to the number of revisions of the image. Table F.13 describes the size of and allowed values for the parameters of this entry.

- **VERSNLOC:** Location of a string containing textual representation of the Version identifier – If the value of this parameter is zero, the string is stored as direct data immediately following the CHARSET parameter. If non-zero, the string is stored as indirect data and the value of the parameter is the string’s starting offset.

- **CHARSET:** Specifies the character set to be used to interpret the character data (see F.2.3.2.1).

NOTE – The meaning and interpretation of the text in this entry is application specific.

Table F.13 – Version identifier

Version identifier			Tag value: X’00000009’
Offset	Parameter	Type, size	Values
0	VERSNLOC	I.32	0 or in range from EOI marker offset to 4, 294, 967, 295
4	CHARSET	I.8	1 to N, where N is largest existing ISO/IEC 8859-N, 254, 255
5	…		

F.2.3.2.10 Tag – Creator identification

This entry describes in textual form the creator of the image. The concept of what constitutes the creator of an image is application specific. Table F.14 describes the size of and allowed values for the parameters of this entry.

- **CREATLOC:** Location of a string containing textual representation of the creator identification – If the value of this parameter is zero, the string is stored as direct data immediately following the CHARSET parameter. If non-zero, the string is stored as indirect data and the value of the parameter is the string's starting offset.
- **CHARSET:** Specifies the character set to be used to interpret the character data (see F.2.3.2.1).

NOTE – The meaning and interpretation of the text in this entry is application specific.

Table F.14 – Creator identification

Creator identification			Tag value: X'0000000A'
Offset	Parameter	Type, size	Values
0	CREATLOC	I.32	0 or in range from EOI marker offset to 4, 294, 967, 295
4	CHARSET	I.8	1 to N, where N is largest existing ISO/IEC 8859-N, 254, 255
5	...		

F.2.3.2.11 Tag – Protection indicator

The presence of this entry, indicates that the image's owner has retained copyright protection and usage rights for the image. The concept of what constitutes valid copyright information is open to interpretation and this Recommendation | International Standard does not intend to attempt to resolve that question. Table F.15 describes the size of and allowed values for the parameters of this entry.

- **LEVAUT:** Indicates the "level of authenticity" assigned to the image by the owner. The allowed values for this parameter are:

  0 = indicates an unknown status;

  1 = indicates a master image;

  2 = indicates an unmodified part of a master image;

  3 = indicates that the image has been modified from the master image.
- **COPYRID:** An 8-bit copyright identifier allocated in accordance with the registration scheme defined in ISO/IEC 13818-2, Amendment 1. It identifies a work type code identifier (such as ISBN, ISSN, ISRC, etc.) whose value is defined by the Copyright Registration Authority established in accordance with ISO/IEC IS13818-2, Amendment 1. If no appropriate value has been allocated, COPYRID shall be set to X'00'.

This entry shall appear at most once in the directory.

Table F.15 – Protection indicator

Protection Indicator			Tag value: X'0000000B'
Offset	Parameter	Type, size	Values
0	LEVAUT	I.8	0 - 3
1	COPYRID	I.8	0 - 255
2	RESERVED	B.2	0 (reserved)

F.2.3.2.12 Tag – Copyright information

This entry describes in textual form copyright information for the image. The concept of what constitutes valid copyright information is open to interpretation and this Recommendation | International Standard does not intend to resolve that question. Table F.16 describes the size of and allowed values for the parameters of this entry.

– **COPYRLOC:** Location of a string containing textual representation of the copyright information – If the value of this parameter is zero, the string is stored as direct data immediately following the CHARSET parameter. If non-zero, the string is stored as indirect data and the value of the parameter is the string's starting offset.

– **CHARSET:** Specifies the character set to be used to interpret the character data (see F.2.3.2.1).

Table F.16 – Copyright information

Copyright information			Tag value: X'0000000C'
Offset	Parameter	Type, size	Values
0	COPYRLOC	I.32	0 or in range from EOI marker offset to 4, 294, 967, 295
4	CHARSET	I.8	1 to N, where N is largest existing ISO/IEC 8859-N, 254, 255
5	...		

F.2.3.2.13 Tag – Contact information

This entry describes in textual form contact information for use of the image. The contents of this entry is application specific. Table F.17 describes the size of and allowed values for the parameters of this entry.

– **REGCON:** This indicates the country of the national body responsible for allocating the contact Registration Authority identifier, REGAUT, according to the numeric version of the country codes as specified in ISO 3166:1993. The appropriate national body shall be nominated by ISO/IEC JTC1/SC29. A value of X'0000' indicates that the contact Registration Authority identifier has been directly allocated by ISO/IEC JTC1/SC29.

– **REGAUT:** An identifier, allocated by the organization indicated by REGCON, specifying a particular contact Registration Authority – A value of X'0000' is used to indicate non-registered contact information.

– **REGID:** A 32-bit registration identifier obtained from the contact Registration Authority indicated by REGAUT – If REGAUT is zero, the meaning of the registration identifier is unspecified.

– **CONTLOC:** Location of a string containing textual representation of the contact information – If the value of this parameter is zero, the string is stored as direct data immediately following the CHARSET parameter. If non-zero, the string is stored as indirect data and the value of the parameter is the string's starting offset.

– **CHARSET:** Specifies the character set to be used to interpret the character data (see F.2.3.2.1).

Table F.17 – Contact information

Contact information			Tag value: X'0000000D'
Offset	Parameter	Type, size	Values
0	REGCON	I.16	0 - 65535
2	REGAUT	I.16	0 - 65535
4	REGID	I.32	0 - 4, 294, 967, 295
8	CONTLOC	I.32	0 or in range from EOI marker offset to 4, 294, 967, 295
12	CHARSET	I.8	1 to N, where N is largest existing ISO/IEC 8859-N, 254, 255
13	...		

F.2.3.2.14 Tag – Tile index

This entry refers to data containing a list of offsets into the file. Each offset points to the X’FF’ byte of a define tile (DTT) marker present in the compressed data stream of the image. The list contains one offset for each and every DTT marker segment in the compressed data stream. Table F.18 describes the size of and allowed values for the parameters of this entry.

- **DTTINDX:** This parameter contains the offset in the file to data that contains a list of offsets into the file pointing at X’FF’ byte of define tile (DTT) marker segments, as described above. This list is sorted in ascending order. The length of the list is given by the NUMDTT parameter. If the value of this parameter is zero, the string is stored as direct data immediately following the NUMDTT parameter. If non-zero, the string is stored as indirect data and the value of the parameter is the string’s starting offset.

- **NUMDTT:** This parameter contains the total number of DTT marker segments (tiles) in the compressed data stream.

This entry shall appear at most once in the directory and only when parameter C (Compression type) of the file header is 5.

Table F.18 – Tile index

Tile index			Tag value: X’0000000E’
Offset	Parameter	Type, size	Values
0	DTTINDX	I.32	0 or in range from EOI marker offset to 4, 294, 967, 295
4	NUMDTT	I.32	2 - 4, 294, 967, 295

F.2.3.2.15 Tag – Scan index

This entry refers to data containing a list, the scan list, having a length equal to the number of scans in the compressed data stream. The scan list contains one 4-word entry for each and every scan in the compressed data stream. Table F.19 describes the size of and allowed values for the parameters of this entry.

- **SCANLIST:** This parameter contains the offset in the file to data that contains a list of 4-word entries. Each entry in the list is comprised of these four 32-bit words:

  1) SCANSTRT, the file offset to the X’FF’ byte of the SOS marker.

  2) SCANEND, the file offset of the first marker after the scan’s compressed data (not counting any RSTn markers within the scan).

  3) RSTLIST, the file offset to the start of the restart marker index list for the scan or zero (if the scan does not contain restart markers, or if the encoder chooses not to store a restart index for this scan). The restart marker index list contains offsets which point to the X’FF’ byte of each RST markers in the scan. This list is sorted in ascending order. The length of the list is given by the NUMRST parameter.

  4) NUMRST, the number of restart markers within the scan or zero (if the scan does not contain restart markers).

  Entries in the scan list shall appear in ascending order by SCANSTRT value.

- **NUMSCAN:** This parameter contains the total number of SOS marker segments in the compressed data stream.

Table F.19 – Scan index

Scan index			Tag value: X'0000000F'
Offset	Parameter	Type, size	Values
0	SCANLIST	I.32	0 or in range from EOI marker offset to 4, 294, 967, 295
4	NUMSCAN	I.32	1 - 4, 294, 967, 295

The purpose of this entry is to provide direct access to specific scan or restart interval without having to scan through the entire compressed data stream. Before a decoder can use this entry for direct access, the compressed data stream shall be processed sequentially until the first SOS marker is encountered. The decoder can then use the SCANEND file offset to skip to the end of any scan and continue decoding. When a scan index is present, the location of table and miscellaneous markers is restricted to permit random access to the data stream. A scan index may appear only if:

a) tables and miscellaneous markers appear only before the first SOS marker; or

b) all tables and miscellaneous markers are repeated before each SOS marker.

This entry shall appear at most once in the directory and only when parameter C (Compression type) of the file header is 5.

F.2.3.2.16 Tag – Set reference

This entry contains a 96-bit reference number (stored as three 32-bit parameters) intended to relate images stored in separate files. Use of this reference number is application specific. Table F.20 describes the size of and allowed values for the parameters of this entry.

- **REFNO1:** The first 32-bit reference number for the image in this file.
- **REFNO2:** The second 32-bit reference number for the image in this file.
- **REFNO3:** The third 32-bit reference number for the image in this file.

Table F.20 – Set reference

Set reference			Tag value: X'00000010'
Offset	Parameter	Type, size	Values
0	REFNO1	I.32	0 - 4, 294, 967, 295
4	REFNO2	I.32	0 - 4, 294, 967, 295
8	REFNO3	I.32	0 - 4, 294, 967, 295

Annex G

Compliance testing

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

The purpose of compliance tests is to provide designers, manufacturers, or users of a product with a set of procedures for determining whether the product meets, to some level of confidence, a specified set of requirements. There are three types of compliance tests defined herein:

- compressed data format compliance tests;
- extended encoder compliance test;
- extended decoder compliance tests.

The aim of the compressed data format compliance tests is to determine whether a particular compressed image data stream meets the interchange format or abbreviated format requirements specified by this Recommendation | International Standard. These tests are performed on the compressed data.

The compliance tests for extended encoders and decoders are procedures for testing whether embodiments of extended encoders and decoders satisfy the requirements stated in clauses 6 and 7 respectively. This Recommendation | International Standard is consistent with the philosophy of CCITT Rec. T.81 | ISO/IEC 10918-1 that imposes more requirements on extended decoders than on extended encoders. This difference is that an extended encoder need only to produce compressed images with a limited range of parameter values, but an extended decoder must handle images with broad ranges of parameters in order to facilitate interchange.

An extended decoder is required to handle either:

a) the full range and combination of the parameter values specified by its coding process (in which case it qualifies as a generic extended decoder); or

b) a subset of the same defined by some application (in which case it is an application-specific extended decoder).

This annex does not define compliance tests for application-specific extended decoders. Compliance tests for application-specific extended decoders may be constructed using the procedure specified by Annex D of ITU-T Rec. T.83 | ISO/IEC 10918-2.

G.1 Compressed data format compliance tests

A particular compressed data stream produced by an extended encoder shall satisfy the requirements of the compliance tests defined in ITU-T Rec. T.83 | ISO/IEC 10918-2 for the selected process and complies with the syntax and code assignments appropriate for the selected extension(s) as specified in Annex B.

G.2 Extended encoder compliance tests

An extended encoder is considered compliant to an encoding process used in combination with one or more encoding extensions if it satisfies the requirements stated in clause 6 of CCITT Rec. T.81 | ISO/IEC 10918-1 and clause 6 of this Recommendation | International Standard, and satisfies the requirements on accuracy for the compliance tests defined in ITU-T Rec. T.83 | ISO/IEC 10918-2 for the process.

In order to determine compliance of extended DCT-based encoders, the test procedure set forth in ITU-T Rec. T.83 | ISO/IEC 10918-2 shall be performed. No compliance tests for extended lossless encoders are defined or required.

G.3 Generic extended decoder compliance tests

A generic decoder is defined as a decoder that can handle the full range and combination of parameter values specified by at least one of the twenty-nine coding processes described in CCITT Rec. T.81 | ISO/IEC 10918-1. (These coding processes are listed in Table B.4.) To be a generic *extended* decoder requires a generic decoder that also handles the full range and combination of parameters values specified by at least one of the extensions described in this Recommendation | International Standard. (These extensions are listed in Table B.5.)

An extended decoder is considered compliant to a decoding process used in combination with one or more decoding extensions if it satisfies the requirements stated in clause 7 of CCITT Rec. T.81 | ISO/IEC 10918-1 and clause 7 of this Recommendation | International Standard, and satisfies the requirements on accuracy for the compliance tests defined in ITU-T Rec. T.83 | ISO/IEC 10918-2 for the process. The tests described here are in addition to the tests defined in ITU-T Rec. T.83 | ISO/IEC 10918-2.

Compressed test data streams required for compliance testing of extended decoders are *not* supplied as part of this Recommendation | International Standard. To test compliance of extended decoders implementors must first construct the compliance test data streams according to the procedure given in G.3.3.

G.3.1 Compliance tests for extended DCT-based decoders

In order to determine compliance of extended DCT-based decoders, the test procedure set forth in A.1.3 and A.1.4 of ITU-T Rec. T.83 | ISO/IEC 10918-2 shall be performed. An extended decoder is found to be compliant if the resulting test data, for all the tests specified for a particular process and all tests specified for the extension used, meet the requirements on accuracy specified in A.1.4 of ITU-T Rec. T.83 | ISO/IEC 10918-2.

Table G.1 lists the compliance tests defined in G.3.4 used to determine compliance of extended DCT-based decoders for each extension.

Table G.1 – Compliance tests for the extensions – Extended DCT-based decoders

Extension	Compliance test Extended DCT-based decoders Huffman coding	Compliance test Extended DCT-based decoders Arithmetic coding
Variable quantization	1	1
Hierarchical selective refinement	2	3
Progressive selective refinement	4	5
Component selective refinement	6	6
Simple tiling	7	7
Pyramidal tiling	8	8
Composite tiling	9	9

G.3.2 Compliance tests for extended lossless decoders

In order to determine compliance of extended lossless decoders, the test procedure set forth in A.2.2 of ITU-T Rec. T.83 | ISO/IEC 10918-2 shall be performed. An extended decoder is found to be compliant if the resulting test data, for all the tests specified for a particular process and all tests specified for the extension used, exactly match the decoder reference test data.

Table G.2 lists the compliance tests defined in G.3.4 used to determine compliance of extended lossless decoders for each extension.

Table G.2 – Compliance tests for the extensions – Extended lossless decoders

Extension	Compliance test Extended lossless decoders Huffman coding	Compliance test Extended lossless decoders Arithmetic coding
Hierarchical selective refinement	10	11
Component selective refinement	12	13
Simple tiling	14	15
Pyramidal tiling	16	17
Composite tiling	18	19

G.3.3 Procedure for construction of generic extended decoder compliance test data

This subclause describes the compliance test data stream structure that may be used as guidelines for construction of the compliance test data. Each test data stream structure implies a particular coding process(es) and extension(s). For any given coding process and extension, the procedure for construction of the compliance test is the following:

1) Specify specific parameters for the extension.

2) Create test image from the provided source image test data. Start by filling the first component, proceed from left-to-right, top-to-bottom, and end with the last component. If necessary, replicate the provided source image test data. Availability of source image test data is described in 4.2.1.

3) Compute the encoder reference test data by one of the following methods:

   a) For DCT-based (lossy) coding processes, apply a double precision floating point FDCT and quantize using the quantization tables specified in ITU-T Rec. T.83 | ISO/IEC 10918-2, Annex B. Then produce the test data stream for the decoding process by encoding quantized DCT coefficients using the entropy encoder.

   b) For lossless coding processes, apply a reference lossless encoder.

4) Compute the decoder reference test data by applying the reference decoder to the compressed data stream.

   For the DCT-based processes only, also apply the double precision floating point IDCT and inverse quantizer to the output. Clip the resulting output data to the range of the sample precision ([0,255] for 8-bit precision and [0,4095] for 12-bit precision). Apply the FDCT and quantizer used in step 4) to the clipped output data to produce the decoder reference test data.

5) Perform the compliance test using the procedures of ITU-T Rec. T.83 | ISO/IEC 10918-2, Annex A.

NOTE – The reference encoder and decoder which is needed to generate the compressed test data stream for generic extended compliance tests should be developed and validated by the creators of the compressed test data.

G.3.4 Compressed test data stream structure for the extensions

This subclause describes the structure of the compressed test data streams utilized by the compliance tests for extended decoders. This subclause is included for guidance only. The actual compressed test data stream structure used for compliance testing is left to the discretion of the creators of the compliance test data.

G.3.4.1 Compressed test data stream structure for the variable quantization extension – Extended DCT-based decoders

Compressed test data stream 1

SOI
VER (V = 1)
COM •
DQT (Pq = 0) quant. tables
DQS quant. scale table (Tc = 0-1)
SOF1 frame parameters (P = 8)
DHT (Th = 0-1) Huffman tables
APP0 • •
SOS scan parameters (Ns = 2)
entropy-coded data segment • •
SOS scan parameters (Ns = 2)
entropy-coded data segment • •
EOI

G.3.4.2 Compressed test data stream structure for the hierarchical selective refinement extension – Extended DCT-based decoders

Compressed test data stream 2
SOI
VER (V = 1)
COM •
DHP hierarchical parameters
DQT quant. tables
SOF1 frame parameters
DHT (Th = 0-3) Huffman tables
SOS scan parameters (Ns = Nf)
entropy-coded data segment • •
EXP expand 2:1, 2:1
SRF refinement selections
SOF5 frame parameters
DHT (Th = 0-3) Huffman tables
SOS scan parameters (Ns = Nf)
entropy-coded data segment •
• (repeat for total of 4 differential frames)
EOI

Compressed test data stream 3
SOI
VER (V = 1)
COM •
DHP hierarchical parameters
DQT quant. tables
SOF9 frame parameters
DAC (Tb = 0-3) AC cond. tables
SOS scan parameters (Ns = Nf)
entropy-coded data segment • •
EXP expand 2:1, 2:1
SRF refinement selections
SOF13 frame parameters
• • (repeat for total of 4 differential frames)
EOI

G.3.4.3 Compressed test data stream structure for the progressive selective refinement extension – Extended DCT-based decoders

Compressed test data stream 4	Compressed test data stream 5
SOI	SOI
VER (V = 1)	VER (V = 1)
COM •	COM •
DQT (Pq = 0) quant. tables	DQT (Pq = 0) quant. tables
DRI restart interval (Ri = 10)	DRI restart interval (Ri = 10)
SOF2 frame parameters (P = 8)	SOF10 frame parameters (P = 8)
DHT (Th = 0-3) Huffman tables	DAC (Tb = 0-3) AC cond. tables
SOS scan parameters (Ns = 3, Ss = Se = 0, Ah = Al = 0)	SOS scan parameters (Ns = 3, Ss = Se = 0, Ah = Al = 0)
entropy-coded data segment • •	entropy-coded data segment • •
DHT (Th = 0-3) Huffman tables	DAC (Tb = 0-3) AC cond. tables
SRS refinement selections	SRS refinement selections
SOS scan parameters (Ns = 3, Ss = Se = 1, Ah = Al = 0)	SOS scan parameters (Ns = 3, Ss = Se = 1, Ah = Al = 0)
entropy-coded data segment •	entropy-coded data segment •
• (total of 10 scans)	• (total of 10 scans)
EOI	EOI

G.3.4.4 Compressed test data stream structure for the component selective refinement extension – Extended DCT-based decoders

Compressed test data stream 6

SOI
VER (V = 1)
COM •
DQT (Pq = 0) quant. tables
SOF1 frame parameters (P = 8, Nf = 4)
DHT (Th = 0-1) Huffman tables
SOS scan parameters (Ns = 1)
entropy-coded data segment • •
SRS refinement selections
SOS scan parameters (Ns = 3)
entropy-coded data segment • •
EOI

G.3.4.5 Compressed test data stream structure for the simple tiling extension – Extended DCT-based decoders

Compressed test data stream 7

SOI
VER (V = 1)
COM •
DTI tiling parameters (TT = 0)
DQT (Pq = 0) quant. tables
DTT tile 0 parameters
SOF1 frame parameters (P = 8)
DHT (Th = 0-1) Huffman tables
SOS scan parameters (Ns = Nf)
entropy-coded data segment • •
DTT tile 1 parameters
SOF1 frame parameters (P = 8)
SOS scan parameters (Ns = Nf)
entropy-coded data segment •
• (repeat for total of 16 tiles)
EOI

G.3.4.6 Compressed test data stream structure for the pyramidal tiling extension – Extended DCT-based decoders

Compressed test data stream 8

SOI
VER (V = 1)
COM •
DTI tiling parameters (TT = 1)
DQT (Pq = 0) quant. tables
DTT tile 0 parameters (TFhs, TFvs = 4)
SOF0 frame parameters (P = 8)
DHT (Th = 0-1) Huffman tables
SOS scan parameters (Ns = Nf)
entropy-coded data segment • •
DTT tile 1 parameters (TFhs, TFvs = 2)
SOF0 frame parameters (P = 8)
SOS scan parameters (Ns = Nf)
entropy-coded data segment •
• (repeat for total of 3 levels, 21 tiles)
EOI

G.3.4.7 Compressed test data stream structure for the composite tiling extension – Extended DCT-based decoders

Compressed test data stream 9

SOI
VER (V = 1)
COM •
DTI tiling parameters (TT = 2)
DQT quant. tables
DTT tile 0 parameters (TFhs, TFvs = 1)
SOF1 frame parameters (P = 8)
DCR component registration
DHT (Th = 0-1) Huffman tables
SOS scan parameters (Ns = Nf)
entropy-coded data segment •
DTT tile 1 parameters (TFhs, TFvs = 2)
SOF1 frame parameters (P = 8)
DCR component registration
SOS scan parameters (Ns = Nf)
entropy-coded data segment •
• (repeat for total of 17 tiles)
EOI

G.3.4.8 Compressed test data stream structure for the hierarchical selective refinement extension – Extended lossless decoders

Compressed test data stream 10

SOI
VER (V = 1)
COM •
DHP hierarchical parameters
DQT quant. tables
SOF3 frame parameters
DHT (Th = 0-3) Huffman tables
SOS scan parameters (Ns = Nf)
entropy-coded data segment • •
EXP expand 2:1, 2:1
SRF refinement selections
SOF7 frame parameters
DHT (Th = 0-3) Huffman tables
SOS scan parameters (Ns = Nf)
entropy-coded data segment •
• (repeat for total of 4 differential frames)
EOI

Compressed test data stream 11

SOI
VER (V = 1)
COM •
DHP hierarchical parameters
DQT quant. tables
SOF11 frame parameters
DAC (Tb = 0-3) AC cond. tables
SOS scan parameters (Ns = Nf)
entropy-coded data segment • •
EXP expand 2:1, 2:1
SRF refinement selections
SOF15 frame parameters
• • (repeat for total of 4 differential frames)
EOI

G.3.4.9 Compressed test data stream structure for the component selective refinement extension – Extended lossless decoders

Compressed test data stream 12	Compressed test data stream 13
SOI	SOI
VER (V = 1)	VER (V = 1)
COM •	COM •
SOF3 frame parameters (P = 8, Nf = 4)	SOF11 frame parameters (P = 8, Nf = 4)
DHT (Th = 0-3) Huffman tables	DAC (Tb = 0-3) AC cond. tables
SOS scan parameters (Ns = 1)	SOS scan parameters (Ns = 1)
entropy-coded data segment • •	entropy-coded data segment • •
SRS refinement selections	SRS refinement selections
SOS scan parameters (Ns = 3)	SOS scan parameters (Ns = 3)
entropy-coded data segment • •	entropy-coded data segment • •
EOI	EOI

G.3.4.10 Compressed test data stream structure for the simple tiling extension – Extended lossless decoders

Compressed test data stream 14	Compressed test data stream 15
SOI	SOI
VER (V = 1)	VER (V = 1)
COM •	COM •
DTI tiling parameters (TT = 0)	DTI tiling parameters (TT = 0)
DTT tile 0 parameters	DTT tile 0 parameters
SOF3 frame parameters (P = 8)	SOF11 frame parameters (P = 8)
DHT (Th = 0-3) Huffman tables	DAC (Tb = 0-3) AC cond. tables
SOS scan parameters (Ns = Nf)	SOS scan parameters (Ns = Nf)
entropy-coded data segment • •	entropy-coded data segment • •
DTT tile 1 parameters	DTT tile 1 parameters
SOF3 frame parameters (P = 8)	SOF11 frame parameters (P = 8)
SOS scan parameters (Ns = Nf)	SOS scan parameters (Ns = Nf)
entropy-coded data segment •	entropy-coded data segment •
• (repeat for total of 16 tiles)	• (repeat for total of 16 tiles)
EOI	EOI

G.3.4.11 Compressed test data stream structure for the pyramidal tiling extension – Extended lossless decoders

Compressed test data stream 16	Compressed test data stream 17
SOI	SOI
VER (V = 1)	VER (V = 1)
COM •	COM •
DTI tiling parameters (TT = 1)	DTI tiling parameters (TT = 1)
DTT tile 0 parameters (TFhs, TFvs = 4)	DTT tile 0 parameters (TFhs, TFvs = 4)
SOF3 frame parameters (P = 8)	SOF11 frame parameters (P = 8)
DHT (Th = 0-3) Huffman tables	DAC (Tb = 0-3) AC cond. tables
SOS scan parameters (Ns = Nf)	SOS scan parameters (Ns = Nf)
entropy-coded data segment • •	entropy-coded data segment • •
DTT tile 1 parameters (TFhs, TFvs = 2)	DTT tile 1 parameters (TFhs, TFvs = 2)
SOF3 frame parameters (P = 8)	SOF11 frame parameters (P = 8)
SOS scan parameters (Ns = Nf)	SOS scan parameters (Ns = Nf)
entropy-coded data segment •	entropy-coded data segment •
• (repeat for total of 3 levels, 21 tiles)	• (repeat for total of 3 levels, 21 tiles)
EOI	EOI

G.3.4.12 Compressed test data stream structure for the composite tiling extension – Extended lossless decoders

Compressed test data stream 18	Compressed test data stream 19
SOI	SOI
VER (V = 1)	VER (V = 1)
COM •	COM •
DTI tiling parameters (TT = 2)	DTI tiling parameters (TT = 2)
DTT tile 0 parameters (TFhs, TFvs = 1)	DTT tile 1 parameters (TFhs, TFvs = 1)
SOF3 frame parameters (P = 8)	SOF11 frame parameters (P = 8)
DCR component registration	DCR component registration
DHT (Th = 0-3) Huffman tables	DAC (Tb = 0-3) AC cond. tables
SOS scan parameters (Ns = Nf)	SOS scan parameters (Ns = Nf)
entropy-coded data segment • •	entropy-coded data segment • •
DTT tile 1 parameters (TFhs, TFvs = 2)	DTT tile 1 parameters (TFhs, TFvs = 2)
SOF3 frame parameters (P = 8)	SOF11 frame parameters (P = 8)
DCR component registration	DCR component registration
SOS scan parameters (Ns = Nf)	SOS scan parameters (Ns = Nf)
entropy-coded data segment •	entropy-coded data segment •
• (repeat for total of 17 tiles)	• (repeat for total of 17 tiles)
EOI	EOI

Annex H

Examples and guidelines

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

This annex provides examples of applications, coding and usage of the features defined in this Recommendation | International Standard. As an example, regions of specific interest can be coded to allow their display at enhanced resolution, either automatically or under user or application control, with a bit stream coded in accordance with this Recommendation | International Standard. This annex offers implementors some guidance to factors which might affect their choice of coding options and the practical implications of particular features of this Recommendation | International Standard.

H.1 Variable quantization

Variable quantization allows several features to implementors. These include:

- transcodability from other compression file formats using variable quantization, particularly MPEG;
- increased subjective quality for particular compressed image sizes;
- adaptive coding to maximize utilization of a fixed bandwidth transmission channel;
- coding to fit to some imposed fixed maximum coded image data size.

H.1.1 Transcoding from MPEG

ISO/IEC 11172-2 (MPEG-1) and ISO/IEC 13818-2 (MPEG-2) define a coding scheme for the transmission of moving image information. Applications may require capture of a video image, and transcoding to the interchange format defined in CCITT Rec. T.81 | ISO/IEC 10918-1 and this Recommendation | International Standard (subsequently referred to as the 'JPEG compressed data stream').

MPEG uses variable quantization, amongst other features, to achieve high compression ratios. As a result, the variable quantization methods defined in this Recommendation | International Standard are needed to provide accurately transcoded images. In particular, the use of I-frames in the MPEG data stream provides a single frame of picture information which could be used (together with information on the current states of the entropy coder, and quantization tables within the MPEG data stream) to generate a data stream compliant with this Recommendation | International Standard.

H.1.2 Maintaining subjective quality

A variety of models can be used to measure the subjective quality of a decoded lossy image. This may be relatively simple (e.g. mean square error compared with original image) or more complex (e.g. based on a psycho-visual study). Also, subjective quality will depend on the target output device(s). In these cases, individual blocks or groups of blocks within an image may be used to generate a comparison between the actual decoded lossy image and the original image. The output of this process is used to control a variable quantizer (in a feedback loop). This can either be used in a single-pass or two-pass process in order to maintain the image quality within defined bounds. The process may be automatic, or could be controlled by a user specifying regions of an image which are of reduced or increased significance.

H.1.3 Constant bandwidth compression

The majority of communications channels have a maximum effective bandwidth available to an application. Variable quantization may be used to control the flow of data from the application, to avoid overflow or underflow of any buffering that the channel employs. In certain cases, applications coding data to this Recommendation | International Standard may be unable to temporarily halt their information flow. Examples include satellite based or other mobile imaging systems providing continual data flow, systems required to provide continual image capture (e.g. for vehicular traffic monitoring), or high speed scanning systems (e.g. for bank check processing).

These systems may use the flow control system provided by the communications channel or measurements of their own internal usage buffer to signal the need for reduced or increased data flow. To reduce the data flow, quantization table values are increased by raising the scale factor. To increase the data flow, quantization table values are decreased by lowering the scale factor.

H.1.4 Fixed file size compression

Many storage systems allocate storage in fixed size units, or are subject to an overall size limitation, particularly if the storage medium is removable. Applications may choose one of a number of possible algorithms to predict a compression ratio and, hence, a final image size for a particular quantization matrix.

For example, use a small number of blocks (selected at random or by some formula) across a large part of the image, compress them, and measure the compression ratio achieved. This prediction can then be used to set a target for compression size, for example at 95% of the available space. Adaptive mechanisms can then track how close the actual encoding process comes to generating the forecasted encoded image size, and control a variable quantizer to approach from below a given coded image size. While the mechanism has been found to work well in practice, implementors should note that there is a risk that the difficulty in setting bounds for a lossy coded image *may* result in a loss of image quality.

H.2 Selective refinement

A number of methods for selectively enhancing particular regions of a displayed image can be selected depending on a number of factors including the desired visual result, coding or application complexity, and external factors such as the use of particular database or transmission systems. In addition to the features described in this subclause, implementors may be able to achieve visually similar final results through the use of tiling (see H.3).

As a general comment on the selective refinement techniques outlined below, hierarchical selective refinement offers a visually improved lower resolution in low definition areas of the image when compared with progressive selective refinement, but at an expense in terms of the amount of memory storage required to implement the coding technique.

H.2.1 Hierarchical selective refinement

A good application of hierarchical selective refinement is a soft trimming of images. For example, in a medical image representation, the areas outside region of interest can be subsampled in order to reduce the total image size. Figure H.1 describes a hierarchical selective refinement of an image. It is mandatory to use integer multiples of MCU's for both offsets and sizes of partial frames (see format example in G.3.4.8).

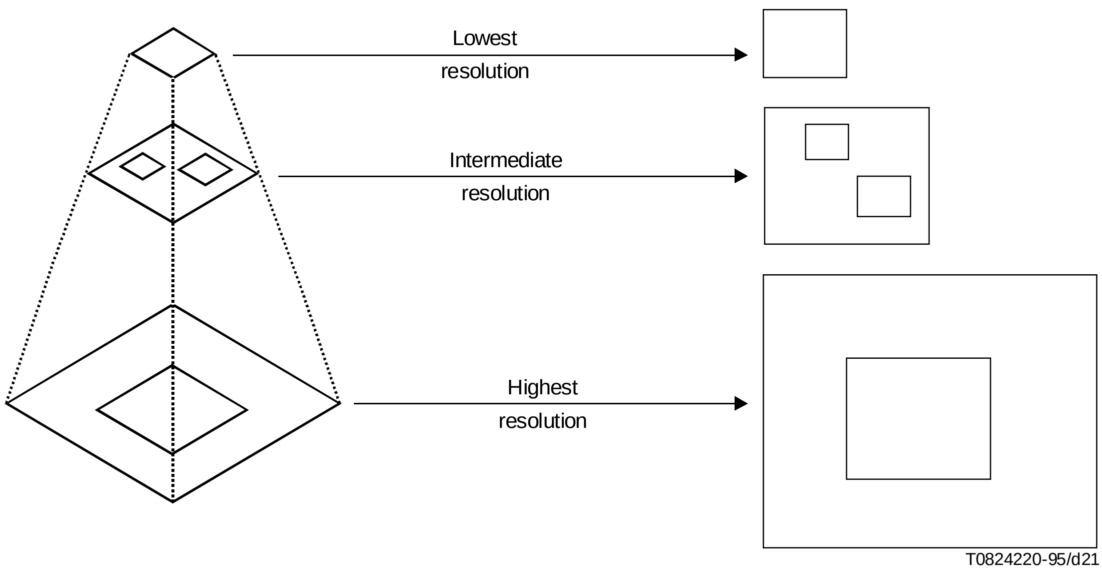

Figure H.1 – An example of hierarchical selective refinement

H.2.2 Progressive selective refinement

A good application of progressive selective refinement is a dual function usage of images. For example, in the series of scans of a traffic control camera, police may need numberplate information at maximum resolution to track vehicles, and a wider, lower resolution image to prosecute the offender. In order to achieve this, the area around bumper height would be selected for progressive refinement.

H.2.3 Component selective refinement

A good application of component progressive selective refinement is a processing of mixed images. For example, a page of a magazine might contain a colour photograph in a text area. Therefore, component selective refinement could provide better colour rendition for the photograph region.

H.3 Tiling

Tiling can be applied for several reasons:

- to divide a large image into a number of interrelated smaller sub-images;
- to provide the ability to process defined parts of an image independently;
- to address application or system specific limitations, such as the availability of display memory;
- to allow a number of images, possibly from dissimilar sources, to be linked together in a single composite image.

It is likely that the successively greater complexity of solutions based on simple, pyramidal, and composite tiling should mean that the number of commercial applications offering the ability to encode or decode images corresponding to these techniques will be restricted and increasingly application and user specific. Therefore it is recommended that the simplest tiling technique capable of offering the desired features be adopted by any implementor wishing to interchange compliant data.

H.3.1 Simple tiling

Simple tiling is useful for breaking huge images into manageable parts. One example of a huge image which needs tiling is Landsat data. Not only are the images large but also the image itself is an endless format.

H.3.2 Pyramidal tiling

In pyramidal tiling, not only is the original image tiled, but it is reduced and tiled at each of a number of different levels of resolution.

An example of the use of pyramidal tiling is the JPEG Tiled Image Pyramid (JTIP). The JTIP structure is described below.

At the top of the pyramid sits the ‘vignette’, a format that takes 1/16th of the display area, used mainly for browsing purposes. The constant size is needed to allow for consistent display in browsing applications. Immediately underneath the vignette, 4 times larger, comes the ‘imagette’, occupying 1/4 of the display area. This image size can be used for comparison of from two to four images when a selection requires a more accurate view of the content of multiple images.

Under this level is the full display area image, the largest displayable format showing the total visual content of the image.

The levels beneath the full-screen image are then tiled into pieces with a maximum size equal to the size of the screen, allowing a magnified part of the image to be set on screen by decoding at most 4 tiles. This process can be repeated to increase resolution in particular regions of interest, or across the image as a whole.

The following steps need to be taken to encode an image in this way using pyramidal coding:

1) Define the target display screen size to use as the tile size; this is, of course, an application specific decision. Example resolutions are, for instance, 640×480, 720×576, or 800×600 (horizontal $\times$ vertical picture elements).

2) Depending on whether the image orientation is portrait or landscape, compute the reduction ratio from total image to screen size, “R”.

3) Use the value of “R” as follows:

$$R = R1 \times 2^n$$

where R1 is a variable in the range 1 to 3, 2^n being a power of 2; R1 is then the first reduction ratio;; all other levels use a reduction ratio of 2.

4) Tile the original image, and compress and store all tiles in scan order from top left-to-bottom right. All tiles are the same size as the target display device, except normally for those on the right and bottom extremities of the image which finish at the image edge.

5) Reduce the image in size by applying a reduction R1 to the original image, and possibly carrying out appropriate filtering. Tile the resulting reduced image as indicated in step 4) above.

6) Apply a 2-times reduction to the resultant image from step 5) and repeat the tiling process indicated in step 4).

7) Continue reducing by a factor of 2 until the screen resolution is reached for a single tile; compress and store this as the full screen image.

8) Reduce by 2 again, compress and store as the “imagette” tile.

9) Carry out a final reduction by 2, compress and store as the “vignette” tile.

10) Build the image file, starting with a header defined by SPIFF, followed by tiles in this order:

- vignette;
- imagette;
- full screen;
- 4-tiles at high definition;
- 16-tiles at very high definition;
- ....;
- original tiled image.

Figure H.2 shows the process in diagrammatic form.

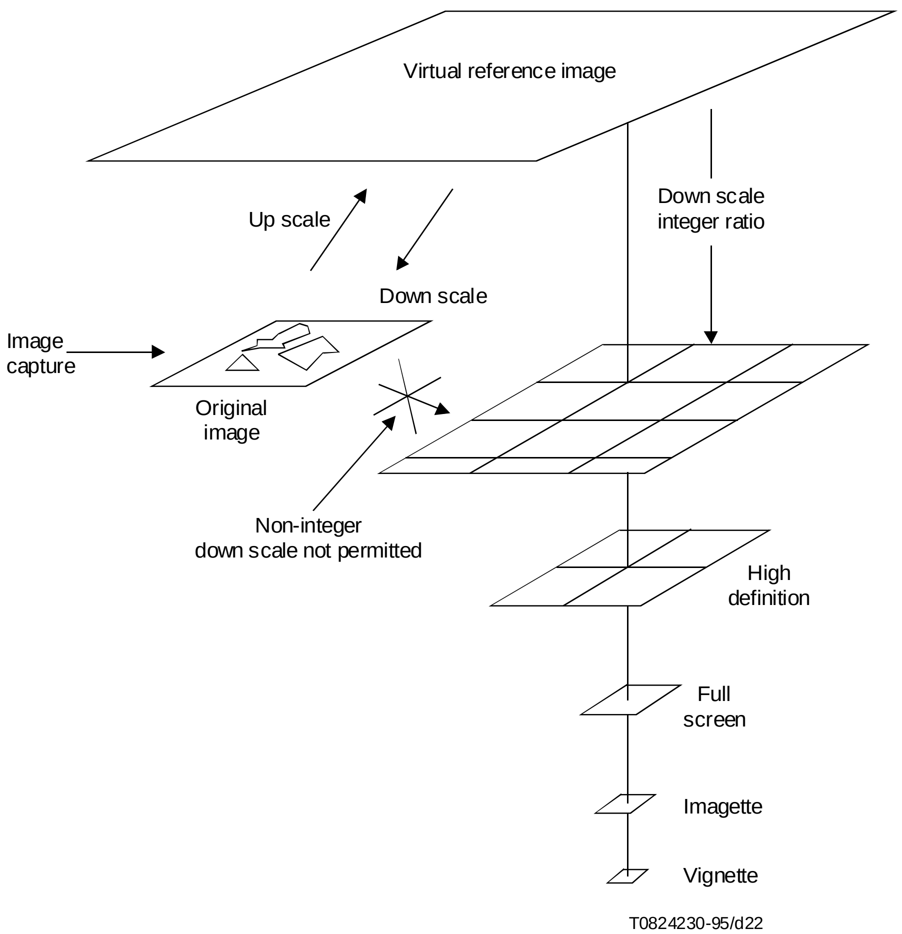

Figure H.2 – Example of JTIP coding

Some existing image coding schemes, for example Photo CD™ are pyramidal in structure, although not using JPEG image compression. These can usually be transcoded to a format compliant with this Recommendation | International Standard by either of two methods:

a) use the highest level of resolution in the original image and follow the procedures outlined above;

b) transcode each level in the original image as separate levels of a pyramid tiling scheme in accordance with this Recommendation | International Standard.

It is suggested that a) above may provide higher quality results, depending on the methods used to reduce resolution in the original pyramidal image.

This Recommendation | International Standard deals with ‘internal tiling’ in which all files form a part of a single JPEG compressed data stream. Some applications may wish to use ‘external tiling’ in which the individual tiles in an image are stored separately as discrete JPEG compressed data streams. This may allow compatibility with existing decoders, protection of parts of an image for commercial or other reasons, for instance by encryption, and for improved access times in on-line retrieval systems. This Recommendation | International Standard does not specify how external tiling should be carried out, as it is application specific.

H.3.3 Composite tiling

One example of composite tiling is overlaying of geographic data of different types. The figure of North America, for example, will be produced using a variety of image sources – satellite, aircraft, and so on. The digitized mapping information need to be presented as a component aligned composite image.

H.4 Still Picture Interchange File Format (SPIFF)

CCITT Rec. T.81 | ISO/IEC 10918-1 did not define an image interchange file format which can be used by other applications. The original intent of JPEG was to allow other applications to encapsulate JPEG data within their own composite file formats and, to some extent, this has been accomplished in both standardized (e.g. ODA, TIFF, CGM, IPI-IIF) and proprietary or de facto standards (e.g. JFIF, RTF, PostScript, etc.).

The increased interest in using JPEG compressed data streams in multimedia applications and a need for application software able to interpret these data streams prior to a full decode (for example, to make decisions about the decode process, or to provide copyright or other reference information) has led to the decision to include a specified file format for the interchange of coded data streams conforming to CCITT Rec. T.81 ISO/IEC 10918-1 and this Recommendation | International Standard (JPEG) as well as other compression standards. Although applications using JPEG compressed data streams as part of a processable image may well need additional information, the inclusion of SPIFF information is relevant for images which may need to be interchanged.

Figure H.3 shows the basic SPIFF file format when being used to contain an image coded using one of the JPEG processes. It is anticipated that in most cases indirect data would not be present.

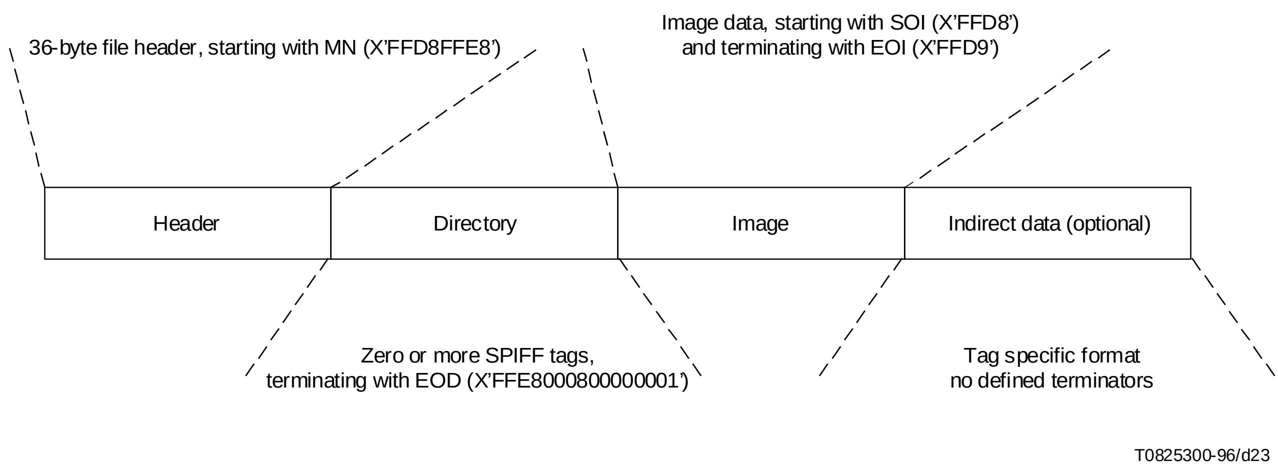

Figure H.3 – High-level syntax for still picture interchange file format with JPEG image

It is suggested that for applications supporting some or all of the features in this Recommendation | International Standard, the default file format should be SPIFF. Also, for images which may require copyright or other restrictions, the approach in H.4.2.5 is suggested.

The following list is a brief summary of the functionality provided by SPIFF.

Header (required tags)

- Profile identifier: type of image data, e.g. binary, continuous tone, etc.;
- Compression type: e.g. MH, MR, JBIG, JPEG, etc.;
- Colour space: e.g. YCbCr(), CIE Lab, etc.;
- Resolution units: e.g. dots per inch, dots per centimeter, etc.;
- Image height: number of lines;
- Image width: number of samples per line;
- Vertical resolution: resolution in resolution units;
- Horizontal resolution: resolution in resolution units.

Directory entries (optional tags; some may be used more than once)

- Transfer characteristics: gamma correction;
- Component registration: location of components with respect to each other;
- Image orientation: rotation or flip of images if required for proper display;
- Thumbnail image spec.: defines ancillary images;
- Image title: text string;
- Image description: text string;
- Time stamp: date and time;
- Version identifier: image version identifier;
- Creator identification: text string;
- Protection indicator: level of authenticity;
- Copyright information: explicit statement of copyright;
- Contact information: id number and text;
- Scan index: pointer(s) to scans (also to RST intervals);
- Tile index: pointer(s) to tiles;
- Set reference: application tag relating this file to other files, e.g. sequence, hierarchical level, etc.

H.4.1 Transcoding from other file formats

Where compatibility with applications that do not currently use SPIFF as an issue, the encapsulation of the existing JPEG compressed data format together with any existing file format embedded in the compressed data stream, is recommended. If this compatibility is not an issue, then the replacement of existing APPn marker segments by information contained in SPIFF is recommended. This is in line with the recommendation made in CCITT Rec. T.81 | ISO/IEC 10918-1, Annex B, where APPn markers should be removed from the interchange format data stream before interchange between application environments.

If non-standardized application, dependent or private information is to be interchanged, this can be added to SPIFF using the mechanisms defined in Annex F for application specific usage. Organizations wishing to standardize such usage are advised to consider approaching either ISO, ITU, or a national standard body for the allocation of specific registered ETAG information.

H.4.2 Use of character string fields

H.4.2.1 Background to character sets

The character string set used in each character string field of SPIFF may be either ISO/IEC 10646, Recommendation T.51, and ISO/IEC 8859-N, as specified in F.2.3.2.1.

H.4.2.2 Fallback to ISO 8859-1

If the application is using the character strings defined in SPIFF, for example to display additional information to a user, it may choose to interpret a subset of the possible character sets available. How this problem is handled is application dependent, but it is suggested that the character set ISO 8859-1 be used to display the information – possibly accompanied by an indication that the information shown may be inaccurate.

H.4.2.3 Use of the COM marker

For the COM marker segment, it is recommended that the ISO 8859-1 character set be used.

H.4.2.4 Recommended length of content

There is no restriction concerning the length of each character string. However, a length of no more than 255 bytes for any one string is recommended.

H.4.2.5 Copyright and other textual information

A number of fields have been provided within SPIFF to assist organizations wishing to establish formal (and possibly automated) processes for adding copyright and other information to a SPIFF encoded file. While this Recommendation | International Standard in no way mandates the meaning or usage of these fields, the intent was to provide some method to assist in providing a degree of commonality between applications.

The following describes a suggested usage for fields of directory entries within this version of SPIFF:

1) *Copyright information*

   Contains a simple copyright notice which might be displayed, for example under user request, to establish the organization or individual claiming copyright on the image.

   – Example: © International Organisation for Standarization, 1991

     International Electrotechnical Commission, 1991

2) *Contact information*

   Contains sufficient information in an internationally acceptable format to allow contact with an organization (or individual) who were aware of the copyright and status position when the image was created.

   – Example: ISO/IEC JTC1/SC29 Secretariat
     IPSJ/ITSCJ Kikai-Shinko-Kaikan Bldg
     3-5-8 Shiba-Koen, Minato-Ku, Tokyo, Japan
     nhirose@attmail.com, Fax +81-3-3431-6493

3) *Creator information*

   Identifies the individual, group, or organization responsible for the original image creation.

   – Example: JPEG Committee

4) *Title*

   Describes the image in a format which can be used in selection or communications regarding the image. It can also allow some applications to generate a displayable list of images contained within a database structure, for example to permit user selection of a particular image without the need to decode the image first. It is recommended that the image title be no more than one line (72 characters).

   – Example: Gold Hill

5) *Image description*

   Contains text which adds additional information to the ETAG Title. This is intended to provide some descriptive text which can be associated with the image as, for example, background information to a picture database.

   – Example: Gold Hill is in a small village, Shaftsbury, in the South West of England. This photograph was part of a test set used to evaluate the coding schemes proposed to the original JPEG committee.

6) Version identifier

   Contains a text string which helps to identify a particular image. This would normally be used during an editing or encoding process to assist in identifying multiple instances of the same image being used as the basis for a set of images which have some common link – for example to compare image quality after decoding.

   When associated with the title, this information might be expected to provide a reference which could be used to identify a particular instance of a SPIFF file to the organization or individual described in the ETAG Contact Information.

   – Example: V1.03

H.5 Recommended decoder recovery and encoder fallback

H.5.1 Purpose of decoder recovery and encoder fallback

When decoders encounter what they do not expect, some simple procedures may help to make the best use of the subsequent coded data stream. Such procedures are referred to as decoder recovery. Also, if in an application specific negotiation process it is found that the decoder does not support some functions, the encoder needs to eliminate such functions. Such procedures are called encoder fallback.

H.5.2 If JPGn or APPn code is present

If a decoder encounters a JPGn marker, there is no way to recover and the decoder has to quit decoding. If a decoder encounters an APPn marker, the decoder should skip the length of data indicated and continue decoding. If an encoder knows that JPGn or APPn is not understood by the decoder, the fallback procedures are given in following subclauses.

H.5.3 Encoder fallback if variable quantization options is not supported

If the encoder finds that the decoder does not support variable quantization, it should regenerate the coded image with a fixed quantization matrix.

H.5.4 Encoder fallback if selective refinement options is not supported

If the encoder finds that the decoder does not support selective refinement options, it should regenerate the coded image without selective refinement, with the quality selected by the encoder.

H.5.5 Encoder fallback if tiling options not supported

If the encoder finds that the decoder does not support tiling options, it should regenerate the coded image without tiling. The encoder may then have to split the image into several smaller images if the image size exceeds the maximum permitted size.

Annex I

Bibliography

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

I.1 General references

Hudson (G.), Yasuda (H.) and Sebestyen (I.): The International Standardization of a Still Picture Compression Technique, *Proc. of IEEE Global Telecommunications Conference*, pp. 1016-1021 (November 1988).

Leger (A.), Mitchell (J.) and Yamazaki (Y.): Still Picture Compression Algorithms Evaluated for International Standardization, *Proc. of the IEEE Global Telecommunications Conference*, pp. 1028-1032 (November 1988).

Netravali (A. N.) and Haskell (B. G.): Digital Pictures: Representation and Compression, *Plenum Press*, New York 1988.

Wallace (G. K.): The JPEG Still Picture Compression Standard, *Communications of the ACM*, Vol. 34, No. 4, pp. 31-44 (April 1991).

Leger (A.), Omachi (T.) and Wallace (G. K.): JPEG Still Picture Compression Algorithm, *Optical Engineering*, Vol. 30, No. 7, pp. 947-954 (July 1991).

Mitchell (J. L.) and Pennebaker (W. B.): Evolving JPEG Color Data Compression Standard, *Standards for Electronic Imaging Systems*, M. Nier, M. E. Courtot, Editors, SPIE Vol. CR37, pp. 68-97 (1991).

Rabbani (M.) and Jones (P.): Digital Image Compression Techniques, *Tutorial Texts in Optical Engineering*, Vol. TT7, SPIE Press (1991).

Barda (J. F.): Codage et Compression des Grandes Images, *Proc. of AFNOR Multimedia and Standardization Conference*, pp. 300-315 Vol. 1, Paris, France, March 1993.

Pennebaker (W. B.) and Mitchell (J.L.) JPEG: Still Image Data Compression Standard, *Van Nostrand Reinhold*, New York 1993.

I.2 Variable Quantization

Joseph (K.), Raychaudhuri (S. Ng. D.), Siracusa (R.), Zepski (J.), Saint-Girons (R.) and Savatier (T.): MPEG++: A Robust Compression and Transport System for Digital HDTV, *Signal Processing: Image Communication 4*, Vol. 4, Nos. 4-5, Elsevier Science Publishers, pp. 307-323, (August 1992).

I.3 Still picture interchange file format

C-Cube Microsystems: JPEG File Interchange Format (JFIF), Version 1.02 (September 1992).

Aldus Corporation: TIFF Developer's Toolkit, Revision 6.0, Seattle, WA, June 1992.

Eastman Kodak: Photo CD, A Planning Guide for Developers, Part No. DC1200R.

Avelem (S.A.): JPEG Tiled Image Pyramid (JTIP) Version 1.02, Gargilesse, France, (March 1995).