INTERNATIONAL STANDARD

ISO/IEC 14492

First edition
2001-12-15

AMENDMENT 1
2004-12-15

Information technology — Lossy/lossless coding of bi-level images

AMENDMENT 1: Encoder

Technologies de l'information — Codage avec ou sans perte des images au trait

AMENDEMENT 1: Codeur

Reference number
ISO/IEC 14492:2001/Amd.1:2004(E)

PDF disclaimer

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

Adobe is a trademark of Adobe Systems Incorporated.

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

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published by ISO in 2005

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 1 to ISO/IEC 14492:2001 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information,* in collaboration with ITU-T. The identical text is published as ITU-T Rec. T.88 (2000)/Amd.1.

Introduction

In this amendment, the following new materials are added:

a) new clauses 9, 10, and 11 to describe the required architecture and procedures for JBIG2 encoding; and

b) a new Annex J to document optional JBIG2 encoding methods.

The encoding procedures in clauses 9 and 10 are essentially the inverse of the decoding procedures already described in clauses 6 and 7 of ITU-T Rec. T.88 | ISO/IEC 14492. To simplify the required new documentation, description of each of the encoding procedures is given by referring to the corresponding decoding procedures in clauses 6 and 7, wherever applicable. Clause 11 and Annex J, however, are new material and thus contain more detailed documentation. In clause 11 (although the encoding complements that of clause 8 of ITU-T Rec. T.88 | ISO/IEC 14492), JBIG2 encoding architecture as well as its technical components are described, and their corresponding implementation methods are given by reference. In J.1, compliant example encoding methods are summarized in table form.

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

Information technology – Lossy/lossless coding of bi-level images

Amendment 1

Encoder

1) New clauses 9, 10, and 11

Add the following clauses:

9 Encoding procedures

The encoding procedures in this clause are essentially the inverse of the decoding procedures already described in clause 6 and will not be duplicated here. The inverse of generic region encoding is described in 6.2. The inverse of generic refinement encoding is described in 6.3. The inverse of text region encoding is described in 6.4. The inverse of symbol dictionary encoding is described in 6.5. The inverse of halftone region encoding is described in 6.6. The inverse of pattern dictionary encoding is described in 6.7.

10 Control encoding procedures

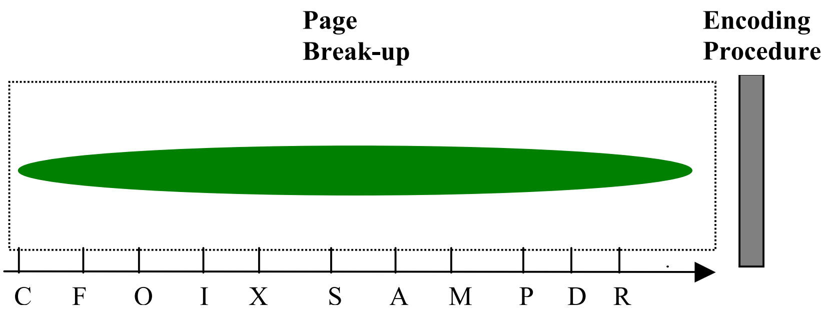
The control encoding procedures in this clause are essentially the inverse of the decoding control procedures already described in clause 7 and will not be duplicated here. The inverse of segment header syntax encoding is described in 7.2. The inverse of segment type encoding is described in 7.3. The segment types syntax for the region segment information field, symbol dictionary segment, text region segment, pattern dictionary segment, halftone region segment, generic region, generic refinement segment, end of page segment, end of stripe segment, end of file segment, profiles segment, code table segment and extension segment are described in detail in 7.4.1 to 7.4.15 respectively.

11 Page break-up

The page break-up ("Front end") procedures in this clause are conceptually the inverse of the page make-up ("Back end") procedures already described in clause 8. However, page break-up also requires additional page and document decomposition steps prior to encoding.

11.1 Page break-up architecture

This clause describes the JBIG2 encoder break-up defined by compliant, but optional, technical 'components' (with a range of 'algorithms' possible to implement each of these components). These JBIG2 page break-up components are a set of processing steps labelled: Capture, Filter, Orient (de-skew), Identify, eXtract, Screen, Align (register), Match, Post-match, Dictionary (optimize), and Refine. An example sequence of this component set is illustrated in the Architectural Components figure below as the horizontal axis with abbreviated labels C F O I X S A M P D R (leading from input on the left to a compressed data stream on the right). The vertical dimension above each label represents the range of possible algorithms that may be used to implement each component. The horizontal band illustrates an example JBIG2 compliant page break-up method, using some algorithm for each architectural component and spanning over these components.



ARCHITECTURAL COMPONENTS

A compliant JBIG2 encoder need not include all architectural components, nor use them in exactly the above sequence.

11.2 Page image decomposition

A page image is decomposed into several groups of sub-images such as marks [J2], line-arts, residues and halftones. Each group is identified and then compressed using an appropriate set of processing (architectural components) from those summarized in 11.2.1 to 11.2.12. Processing may include one or more of these component techniques prior to bitstream creation. The specific algorithm selected for each processing step is left up to the implementer but compliant examples for each processing step are provided in J.1. Implementing a full combination of these components, each using a compliant example encoding method, will result in an encoder capable of producing reasonable near-lossless quality for most 300 + dpi images.

11.2.1 Capture (rasterize)

Capture (rasterization) is a process by which an image source is converted into a two-dimensional bi-level raster image. This is done by mapping a region of the image source to a set of pixels of the raster image, and then assigning a 1-bit colour value to each pixel. In the scope of this amendment, two types of images are defined: generated and scanned images. A generated image is an image converted from a computer-generated metafile or vector graphic (e.g., a bitmap rasterized from a document created using a typical word processor), whereas a scanned image is an image obtained from a paper document by means of imaging hardware such as a scanner or facsimile.

11.2.2 Filter

In most cases, a scanning process is noisy, and the resulting scanned image may contain random pixel values not representative of the original source. These pixels or small groups of pixels are called flyspecks. It is often desireable to remove flyspecks in a scanned image to improve compression efficiency as well as visual quality of the reconstructed image. A scanned image also contains quantization errors, i.e., identical marks in the original image may be slightly different in the scanned image. Smoothing the edges of the marks helps to recover the equivalence of such identical marks in the scanned image and also improve compression efficiency. These filtering techniques are shown as a reference in J.1. Filtering is seldom required for generated images although these techniques may still be applied.

11.2.3 Orient (de-skew)

A scanned image may be skewed when it is scanned or photocopied at a slight angle, and it is often beneficial to identify and adjust any skew prior to compression. In most texts, marks (characters) are aligned in straight lines, and examining the slope of these lines that align pairs of marks yields the skew angle. Several methods of de-skewing are shown as a reference in J.1.

11.2.4 Identify

Identification of sub-image categories involves two processes: segmentation and classification. First an image is segmented into groups of sub-mages or 'regions' having similar characteristics. These regions (segments) are then classified into pre-defined categories such as textual data, line-art and halftones, to which appropriate compression methods are applied.

11.2.5 Extract

A symbol (character) is a mark consisting of black pixels. A symbol boundary is first traced by observing the connectivity of black pixels, and the adjacent black pixels are extracted to form a symbol. Although simply extracting all the pixels confined by the boundary may work in most cases, it does not handle nested marks. Several methods are shown as a reference in J.1.

11.2.6 Screen

Comparing an extracted mark against all the symbols in the dictionary is inefficient especially when the dictionary size is large, and relatively complex matching criteria as described in 11.2.8 are used. Simple methods, such as restricting comparisons to only be made between marks and dictionary symbols with similar width and height, can be used to find possible matching candidates. More detailed approaches are shown as a reference in J.1.

11.2.7 Align (register)

Symbols are often aligned (registered) in the dictionary using the same criteria selected for the screening method in 11.2.6. When distribution of black pixels is tested against symbols in a dictionary to find matching candidates, aligning symbols along their centroids can enhance the screening rate. More detailed approaches are shown as a reference in J.1.

11.2.8 Match

Marks are extracted from a region containing textual data and compared with existing symbols in a dictionary, in order to exploit any similarities between them for better compression. Basically, each mark is tested to determine whether it is similar enough to be considered a 'match' to one of the existing symbols. One way of matching is to first obtain a difference bitmap between the mark and a symbol and test the number of black pixels in the difference bitmap to a pre-defined threshold. Giving more weight to the clustered black pixels in the difference bitmap usually improves matching results. When a close match is found, a reference to the matching symbol in the dictionary is coded. When there is no close match, the extracted mark is stored as a new symbol in the dictionary.

11.2.9 Post-match

Several additional criteria and processing steps may be applied to the symbol dictionary to improve image quality. A best dictionary symbol shape may be determined by examining several similar symbols, which have already passed the matching step. Direct encoding of a symbol or alignment of symbol bottoms may also be used to improve symbol dictionary accuracy.

11.2.10 Dictionary (optimize)

After a symbol dictionary has been generated, it may be examined further to identify any 'singletons' [J2]. Singletons are symbols in the dictionary that have not been referenced by more than one mark. One may sometimes wish to remove such symbols from the dictionary and place them back into the 'residue' sub-images (which contain any residual marks). Such a residue image is compressed using a JBIG2 'generic' entropy encoder.

11.2.11 Refine

Encoded image (or symbol) bitmaps may also be subsequently 'refined' to similar (but different) bitmaps [J1], [J4]. For example, where images are first encoded in a near-lossless manner (e.g., when scanned image symbols are encoded using dictionaries), they can be subsequently 'refinement' encoded to a fully lossless representation of the original image. Also, successive dictionary symbols may be more efficiently encoded as 'refinements' of symbols encoded previously.

11.3 Multi-page document composition

An encoder may organize multi-page document segments using a sequential, random or embedded organization as described in Annex D (File Formats). Dictionary segments may be organized into one global segment, one or more segments per page or stripe, or a combination of global and page-specific dictionary segments. Dictionary segment organization dramatically effects decoder performance and should be carefully selected for a specific application.

2) New Annex J

Add the following annex:

Annex J

Compliant example encoding methods

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

A normative JBIG2 encoder should perform the following:

- lossy encoding mode is mandatory – lossless is optional;
- minimum resolution of 300 dpi (to reduce matching error and accommodate a number of encoder system simplifications, which would otherwise be required for 200-dpi images);
- two or more stripes;
- scanned or generated image source.

The following table contains compliant example references to a range of 'algorithms' useable to implement each of the optional components of a compliant JBIG2 encoding architecture as described in clause 11. These example encoding methods target expired (or expiring) patents, openly published methods and royalty-free patents. For detailed descriptions of each technique, refer to the appropriate document, patent or book. This is not an exhaustive list of algorithms and/or component methods. Selecting one of each method should result in reasonable encoder performance.

J.1 List of JBIG2 encoding components and corresponding algorithms

Architectural component			Component method	Method reference (Found in J.2)
Scanned	Capture (rasterize) – Resolution – Striping		Digitize from sensor to bi-tonal bitmap – 300 dpi – 2 or more	– None – None
	Filter – Fly-speck remove – Quantization error remove		– Isolated mismatched pixels method – Single protruding pixel method	– Figure 3 of [J1] – Page 217 of [J1]
	Orient (de-skew)		Adjacent mark-based slope detection, Hough transform	– Pages 357-372 of [J2]
	Identify (Region) – Segment – Classify		– Recursive x-y cut – Texture analyzer	– Pages 372-384 of [J2] – Pages 385-388 of [J2]
	Extract – Isolate (black shapes) – Truncate (by size) – Force (to avoid generic region encoding)		– 8 connected boundary trace/rubout-based region-fill approach – Min/max of shape w & h – Not required (default)	– Pages 320-325 of [J2], [J3] – [J3] – None
	Screen (means of speeding up match)		Comparison of height, width and/or distribution of black pixels	– Pages 332-333 of [J2]
	Align (register)		Alignment of symbols using centroid	– Pages 332-333 of [J2]
	Match – Lossy/Lossless – Lossless		– Weighted XOR (WXOR) plus CSIS (combined size-independent strategy) $^{b)}$ /CTM (compression-based template matching) $^{c)}$ – XOR (Hamming distance = 0)	– Pages 325-332 of [J2], [J3] and [J4] – None

Architectural component			Component method	Method reference (Found in J.2)
	Post-match			
	– Symbol optimize		– 'Best' dictionary symbol shape (not just simple average of the marks)	– None
	– Symbol encoding		– Direct encoding (no refinement)	– None
	– Symbol placement optimization		– Align symbol bottoms	– None
	Dictionary			
	– Singletons		– In dictionary[d] (default)	– None
	– Page spanning		– Sequentially incremental[e] (default)	– None
	Refine		Arithmetic (inverse of JBIG2 decoder process)	– [J4], ITU-T Rec. T.88 | ISO/IEC 14492
	Encode (bitstream generation)		Arithmetic or Huffman (inverse of JBIG2 decoder process)	– ITU-T Rec. T.88 | ISO/IEC 14492
Generated	Capture (rasterize) – at page or character level		Rasterize from generated to bi-tonal bitmap – at page or character level	
	– Resolution		– 300 dpi	– None
	– Striping		– 2 or more	– None
	Filter		N/A	N/A
	Orient (de-skew)		N/A	N/A
	Identify (Region)			
	– Segment		– Recursive x-y cut	– Pages 372-384 of [J2]
	– Classify		– Texture analyzer	– Pages 385-388 of [J2]
	Extract			
	– Isolate (black shapes)		– 8 connected boundary trace/rub-based region-fill approach	– Pages 320-325 of [J2], [J3]
	– Truncate (by size)		– Min/max of shape w & h	– [J3]
	– Force (to avoid generic region encoding)		– Not required (default)	– None
	Screen (means of speeding up match)		Comparison of height, width and/or distribution of black pixels	Pages 332-333 of [J2]
	Align (register)		Identical (Bounding box is identical)	None
	Match[a]		XOR (Hamming distance = 0)	None
	Post-match			
	– Symbol optimize		– N/A	– N/A
	– Symbol encoding		– Direct encoding (no refinement)	– None
	– Symbol placement optimization		– N/A	– N/A
	Dictionary			
	– Singletons		– In dictionary[d] (default)	– None
	– Page spanning		– Sequentially incremental[e] (default)	– None
	Refine		N/A	N/A
	Encode (bitstream generation)		Arithmetic or Huffman (inverse of JBIG2 decoder process)	– ITU-T Rec. T.88 | ISO/IEC 14492

[a] For some cases of JBIG2 encoding from Generated data, it may be practical to also eliminate the Extract, Screen and Align components by implementing the Match component before the Capture (rasterize) component.

[b] Add CSIS to reduce substitution errors of WXOR.

[c] Select WXOR+CSIS for less computation than CTM.

[d] Minimizing number of Singletons in the dictionary will reduce dictionary memory requirements, and optimizing the sizes (h, w) of Singletons in the dictionary may increase the compression.

[e] Random page access is enabled if, for example, one uses a single multi-page-spanning 'common' dictionary, plus multiple 'page unique' dictionaries.