INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

T.89

(09/2001)

SERIES T: TERMINALS FOR TELEMATIC SERVICES

Application profiles for Recommendation T.88 – Lossy/lossless coding of bi-level images (JBIG2) for facsimile

ITU-T Recommendation T.89

ITU-T Recommendation T.89

Application profiles for Recommendation T.88 – Lossy/lossless coding of bi-level images (JBIG2) for facsimile

Summary

This Recommendation, "Application Profiles for Recommendation T.88", specifies application profiles of the JBIG2 coding scheme, defined in ITU-T Rec. T.88 | ISO/IEC 14492, for facsimile applications. The JBIG2 Recommendation specifies a collection of standard encoder/decoder components, referenced as a tool kit, that are used in generating and decoding JBIG2 conformant data streams. JBIG2 has standardized seven profiles, and encourages definition of additional application profiles to satisfy further needs of various application environments.

Source

ITU-T Recommendation T.89 was revised by ITU-T Study Group 16 (2001-2004) and approved under the WTSA Resolution 1 procedure on 5 September 2001.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

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.

NOTE

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

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

ITU-T Recommendation T.89

Application profiles for Recommendation T.88 – Lossy/lossless coding of bi-level images (JBIG2) for facsimile

1 Scope

This Recommendation defines application profiles of ITU-T Rec. T.88 | ISO/IEC 14492 "Lossy/lossless coding of bi-level images (JBIG2)" for facsimile applications.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

- ITU-T Recommendation T.44 (1999), *Mixed raster content (MRC)*.
- ITU-T Recommendation T.4 (1999), *Standardization of Group 3 facsimile terminals for document transmission.*
- ITU-T Recommendation T.88 (2000) | ISO/IEC 14492:2001, *Information technology – Lossy/lossless coding of bi-level images.* (Commonly referred to as JBIG2 standard.)

3 Principle

This Recommendation specifies application profiles of ITU-T Rec. T.88 | ISO/IEC 14492 for facsimile applications.

JBIG2's tunable lossy/lossless compression of bi-level images is made possible by mixing and matching the various components and parameters within its tool-kit collection of components and parameters. The JBIG2 tool kit contains two basic categories of image and integer coding methods:

1) Arithmetic coding, as defined in Annex E/T.88, is used in the encoding of both image and integer data;
2) MMR coding, as defined in 6.2.6/T.88 and Huffman coding, as defined in Annex B/T.88, are used in the encoding of image and integer data respectively.

These encodings are selectively applied, using a variety of parameter values, to segmented image regions containing image types such as text, dithered (half-tone) and generic bit-map data.

This Recommendation defines a set of JBIG2 application profiles to be used in the decoding of a JBIG2 data stream. Given that the JBIG2 Recommendation (T.88) is a decoder-only standard, the profiles defined within this Recommendation do not address encoding. The profiles are categorized by image and integer coding methodology, image region types and memory constraining parameters. To insure interoperability between various implementations, this Recommendation defines a base profile, which shall be implemented by all facsimile implementations that use JBIG2. The base profile is augmented by a set of standardized optional profiles. Collectively these profiles deliver different levels of performance over a range of facsimile implementations.

Error-free or error-corrected transmission, as defined in Annex A/T.4, and the shared data structure defined in ITU-T Rec. T.44 shall be used in JBIG2 facsimile implementations. Mode 4 or higher of ITU-T Rec. T.44 and Annex H/T.4 ("Black-and-White Mixed Raster Content Profile (MRCbw") clause) shall be used when the "JBIG2 fax profiles", specified within this Recommendation, are implemented in colour and in black-and-white-only applications respectively.

4 Facsimile profiles

Multiple facsimile profiles are defined in 4.1 "JBIG2 fax profiles". These profiles are intended to accommodate applications spanning a range of implementation resource requirement from stand-alone terminals to laptop and desktop computers.

4.1 JBIG2 FAX profiles

Table 1 defines one mandatory profile – Profile 1: BASE – and four optional profiles – Profiles 2: Upper Huffman, Profile 3: Lower arithmetic, Profile 4: Medium lossy/lossless arithmetic and Profile 5: Medium lossy/lossless arithmetic/Huffman. The JBIG2 fax profiles table also contains the outline for an additional optional profile, which is provided for information as it is still under study and has not been approved for implementation. Profiles 1 through 5 have been reserved by the ITU and communicated to ISO/IEC JTC1 SC29, which has reserved profile identification numbers 0x00000100 through 0x00000FFF for ITU-T disposition. Profile identification numbers 0x00000101 through 0x00000105 are assigned to Profiles 1 through 5 above respectively. The relative complexity and working memory requirements of a profile generally increases as the value of its profile number increase for a particular base coder (i.e. arithmetic or Huffman). Accordingly, a reader supporting a profile with a higher value profile number shall be capable of also supporting a profile with a lower value profile number if it utilizes the same base coder.

The BASE profile (Profile 1 or 0x00000101) is designed to accommodate minimal implementation resources in a stand-alone application environment. It is effectively the minimum subset of the lowest level JBIG2 profile, Profile 0x00000007 (see Table F.7/T.88). Consistent with the most prevalent facsimile implementations of today, the BASE profile use MMR coding in the coding of bitmap data and Huffman coding scheme in the coding of numeric (integer) data. The main advantage of the profile is the greatly increased compression available through the use of "lossy" JBIG2 coding. The optional Upper Huffman profile (Profile 2 or 0x00000102), using MMR and Huffman coders for bitmap and numeric respectively, is based on the less constrained JBIG2 Huffman-based profile, Profile 0x00000005 (see Table F.5/T.88). Profile 2 is defined to provide enhanced performance, including specific half-tone region coding via pattern matching and provisions that accommodate use of "color tags" as defined in ITU-T Rec. T.44, to the stand-alone facsimile application environment. Use of Profiles 1 and 2 may be suitable for other low complexity and low-speed processor applications, such as high-speed printing. Definition of Profile 3 (0x00000103) "Lower arithmetic" recognizes the growing trend towards adoption of arithmetic-based coders in facsimile applications and uses arithmetic for both bitmap and numeric coding. Profile 3 is intended to provide a minimal subset of the most highly constrained JBIG2 arithmetic profile, Profile 0x00000006 (see Table F.6/T.88). The Medium lossy/lossless arithmetic profile (Profile 4 or 0x00000104) is defined to provide lossless enhancement to Profile 3. Profile 4 is a subset of the less constrained JBIG2 arithmetic profile, Profile 0x00000003 (see Table F.3/T.88). The Medium lossy/lossless arithmetic/Huffman profile (Profile 5 or 0x00000105) is defined to accommodate the flexibility of using arithmetic, Huffman and MMR base coders as appropriate, along with provisions for both lossy and lossless JBIG2 coding modes. Selectively, arithmetic or MMR may be used for bitmap, arithmetic or Huffman for numeric, and arithmetic for refinement region coding. Additionally, there is provision for specific half-tone region coding via pattern matching. The provisions that accommodate use of "colour tags", which are associated with Profiles 2 and 4, are retained in Profile 5. Profile 5 is a combined subset of the two less constrained and refinement enabled JBIG2 arithmetic and Huffman-based profiles, Profiles 0x00000003 and 0x00000004 (see Tables F.3/T.88 and F.4/T88). Profiles 1 through 3 uses the "lossy" JBIG2 coding mode while Profiles 4 and 5 accommodate both "lossy" and "lossless" modes. Use of Profiles 3 through 5 may be suitable for medium complexity and medium-speed processor applications such as high-end facsimile or other applications, such as multi-function and web-based applications. All profiles can also support "lossless" coding for generated data or if symbol coding is not used (as in JBIG-1).

Clause 4.2 provides background on the memory-related function constraints.

NOTE – For conciseness, ITU-T Rec. T.88 terminology such as "Generic region decoding procedure" has been replaced in T.89 by "direct bitmap coding", and "Generic refinement region decoding procedure" has been replaced by "refinement bitmap coding".

Table 1/T.89 – JBIG2 fax profiles

Number	Functions	Profiles (related to the profiles recommended in ITU-T Rec. T.88 | ISO/IEC 14492 Annex F)						Function Values		
		0x00000101 BASE (Note 8) (Table F.7 minimal subset)	**0x00000102 Upper Huffman (Table F.5)**	**0x00000103 Lower arithmetic (Table F.6 minimal subset)**	**0x00000104 Medium lossy/ lossless arithmetic (Table F.3 subset)**	**0x00000105 Medium lossy/ lossless arithmetic/ Huffman (Tables F.3 and F.4 subsets)**	**X+2 Future Study FULL arithmetic and Huffman (Table F.1 subset)**	**1**	**2**	**3**
1	Direct bitmap coding (Notes 1 and 2)	1	1	2	2	3	3	MMR	Arithmetic	Both
2	Direct bitmap arithmetic coding template (Note 1)	N/A	N/A	1	1	1	2	Restricted	All	
3	*Template size* (Note 3)	N/A	N/A	10 pixels	10 pixels	10, 13 pixels	10, 13, 16 pixels			
4	Direct bitmap arithmetic coding AT pixels (Note 1)	N/A	N/A	1	1	1	2	Restricted	All	
5	*AT pixel location limit* (Notes 3 and 7)	N/A	N/A	Previous 0 rows, 127 columns or nominal location (Note 7)	Previous 0 rows, 127 columns or nominal location	Previous 16 rows, 127 columns	Previous 16 rows?, ? columns			
6	Direct bitmap arithmetic coding TPGD (Note 1)	N/A	N/A	1	1	2	2	TPGD forbidden	Allowed	
7	Refinement bitmap coding (Note 1)	1	1	1	2	2	2	Forbidden	Allowed	
8	Refinement bitmap arithmetic coding template (Note 1)	N/A	N/A	N/A	1	1	2	Restricted	All	
9	*Template size* (Note 3)	N/A	N/A	N/A	10	10, 13 pixels	10, 13 pixels			
10	Refinement bitmap arithmetic coding AT pixels (Note 1)	N/A	N/A	N/A	N/A	N/A	2	Restricted	All	

Table 1/T.89 – JBIG2 fax profiles

Number	Functions	Profiles (related to the profiles recommended in ITU-T Rec. T.88 | ISO/IEC 14492 Annex F)						Function Values		
		0x00000101 BASE (Note 8) (Table F.7 minimal subset)	0x00000102 Upper Huffman (Table F.5)	0x00000103 Lower arithmetic (Table F.6 minimal subset)	0x00000104 Medium lossy/ lossless arithmetic (Table F.3 subset)	0x00000105 Medium lossy/ lossless arithmetic/ Huffman (Tables F.3 and F.4 subsets)	X+2 Future Study FULL arithmetic and Huffman (Table F.1 subset)	1	2	3
11	*AT pixel location limit* (Note 3)	N/A	N/A	N/A	N/A	N/A	Negotiable 16?			
12	Refinement bitmap arithmetic coding TPGR (Note 1)	N/A	N/A	N/A	1	2	2	Forbidden	Allowed	
13	Auxiliary buffers (Note 1)	1	1	1	1	2	2	Forbidden	Allowed, memory restriction	Allowed
14	*Memory limit* (Note 3)	N/A	N/A	N/A	N/A	1	Negotiable?	100% of resolution dependent page (or max stripe) buffer size (e.g. 1.0 Mbytes at 300 dpi and 2.0 Mbytes at 400 dpi).	Negotiable	
15	Integer coding (Numerical data) (Notes 1 and 2)	1	1	2	2	3	3	Huffman	Arithmetic	Both
16	Huffman table choices (Note 2)	1a	3	N/A	N/A	3	3	Restricted – JBIG2 7.4.2/T.88 and 7.4.3/T.88 Huffman list: a) first table (flag bits = 0) only (~3 K memory) b) all 3 defined tables (flag bit =1) (~9 K memory)	All + memory limit	All + variable, no memory restriction

Table 1/T.89 – JBIG2 fax profiles

Number	Functions	Profiles (related to the profiles recommended in ITU-T Rec. T.88 | ISO/IEC 14492 Annex F)						Function Values		
		0x00000101 BASE (Note 8) (Table F.7 minimal subset)	**0x00000102 Upper Huffman (Table F.5)**	**0x00000103 Lower arithmetic (Table F.6 minimal subset)**	**0x00000104 Medium lossy/ lossless arithmetic (Table F.3 subset)**	**0x00000105 Medium lossy/ lossless arithmetic/ Huffman (Tables F.3 and F.4 subsets)**	**X+2 Future Study FULL arithmetic and Huffman (Table F.1 subset)**	**1**	**2**	**3**
17	Symbol coding (Note 4)	2	2	2	2	2	2	Forbidden	Allowed, memory restriction	Allowed
18	*Memory limit* (Note 3)	1	1	1	1	1	TBD	Memory levels[a)] in Mbytes: Level 1 = 1.0 Level 2 = 2.0 Level 3 = unlimited[b)] [a)] All decoders shall accommodate at least Level 1. Levels 2 or 3 may be optionally supported. [b)] Consistent with host-based implementations (i.e. ≥32 Mbytes)		
19	Symbol-coding strip size (Note 4)	2	2	2	2	2	2	Restricted	All 4 stripe sizes (i.e. 1, 2, 4 and 8 pixels)	
20	Symbol aggregation (Note 4)	1	1	1	2	2	3	Forbidden	N = 1, required for symbol-by-symbol refinement	Any, appropriate for symbol build-up

Table 1/T.89 – JBIG2 fax profiles

Number	Functions	Profiles (related to the profiles recommended in ITU-T Rec. T.88 | ISO/IEC 14492 Annex F)						Function Values		
		0x00000101 BASE (Note 8) (Table F.7 minimal subset)	0x00000102 Upper Huffman (Table F.5)	0x00000103 Lower arithmetic (Table F.6 minimal subset)	0x00000104 Medium lossy/ lossless arithmetic (Table F.3 subset)	0x00000105 Medium lossy/ lossless arithmetic/ Huffman (Tables F.3 and F.4 subsets)	X+2 Future Study FULL arithmetic and Huffman (Table F.1 subset)	1	2	3
21	Half-tone coding (Note 5)	1	2	1	1	2	2	Forbidden	Allowed, memory restriction	Allowed
22	*Memory limit* (Note 3)	N/A	1	N/A	N/A	1	3	Approximately 110% of resolution-dependent page buffer size (i.e. 1.0 Mbytes at 300 dpi and 2.0 Mbytes at 400 dpi). No skip mask	Approxi-mately 110% of resolution dependent page buffer size (i.e. 1.0 Mbytes at 300 dpi and 2.0 Mbytes at 400 dpi).	negotiable
23	Half-tone grid orientation (Note 5)	N/A	2	N/A	N/A	2	2	0 degrees	Any	
24	Half-tone grid cell size (Note 5)	N/A	2	N/A	N/A	2	2	Integer	Fractional	
25	Transposition (Note 6)	1	2	1	2	2	2	Forbidden – non-transposed only	Allowed	
26	Reference corner (Note 6)	1	2	1	2	2	2	Restricted – LOWERLEFT only	All	

Table 1/T.89 – JBIG2 fax profiles

Number	Functions	Profiles (related to the profiles recommended in ITU-T Rec. T.88 | ISO/IEC 14492 Annex F)						Function Values		
		0x00000101 BASE (Note 8) (Table F.7 minimal subset)	0x00000102 Upper Huffman (Table F.5)	0x00000103 Lower arithmetic (Table F.6 minimal subset)	0x00000104 Medium lossy/ lossless arithmetic (Table F.3 subset)	0x00000105 Medium lossy/ lossless arithmetic/ Huffman (Tables F.3 and F.4 subsets)	X+2 Future Study FULL arithmetic and Huffman (Table F.1 subset)	1	2	3
27	Striping (Note 6)	1a & b	1a	1a & b	1a	1a	2	Required: a) minimum of 2 stripes per page b) stripes containing text region segments shall not contain other region segments such as half-tone or generic	Required	
28	*Stripe size* (Note 3)	1	1	1	1	1	2	default = 1 K lines, almost (Note 9) full page max	default = page, full page max	
29	Page default pixel value (Note 6)	1	1	1	1	1	2	0 only	0 or 1	
30	Text region default pixel value (SBDEFPIXEL) (Note 4)	1	1	1	1	1	2	0 only	0 or 1	
31	Half-tone region default pixel value (HDEFPIXEL) (Note 5)	N/A	1	N/A	N/A	1	2	0 only	0 or 1	
32	Region external combination operator (Note 6)	1	1	1	1	1	2	OR, XOR only	All	

Table 1/T.89 – JBIG2 fax profiles

Number	Functions	Profiles (related to the profiles recommended in ITU-T Rec. T.88 | ISO/IEC 14492 Annex F)						Function Values		
		0x00000101 BASE (Note 8) (Table F.7 minimal subset)	**0x00000102 Upper Huffman (Table F.5)**	**0x00000103 Lower arithmetic (Table F.6 minimal subset)**	**0x00000104 Medium lossy/ lossless arithmetic (Table F.3 subset)**	**0x00000105 Medium lossy/ lossless arithmetic/ Huffman (Tables F.3 and F.4 subsets)**	**X+2 Future Study FULL arithmetic and Huffman (Table F.1 subset)**	**1**	**2**	**3**
33	Text region internal combination operator (SBCOMBOP) (Note 4)	1	1	1	1	1	2	OR, XOR only	All	
34	Half-tone region internal combination operator (HCOMBOP) (Note 5)	N/A	1	N/A	N/A	1	2	OR, XOR only	All	

NOTE 1 – Arithmetic coder-related function.

NOTE 2 – MMR/Huffman coder-related function.

NOTE 3 – Memory-related function constraint.

NOTE 4 – Symbol region-related function.

NOTE 5 – Half-tone region-related function.

NOTE 6 – Commonly applied function.

NOTE 7 – The AT pixel may be placed at its nominal location, or (if it is moved from there) placed anywhere in the previous n rows, m columns, which means that the previous m pixels in the current row, and the pixels within +/–m in the n – 1 previous rows may be used for AT pixels.

NOTE 8 – The parameters n and m are non-negative integers.

NOTE 9 – The number of scan lines within 'almost full page' is the number of scan lines in a full page minus one.

All JBIG2 T.89 implementations shall include support for this profile.

Table 1 is read by selecting one of the profiles and traversing down the associated column to the various value entries. The values are interpreted by looking to the left along the associated row to determine the function name, listed in the function column. If the function name contains a (Note 3) designation then it is a function constraint, (see 4.2 for background descriptions), of the function in the row directly above. The values of function constraints are self-explanatory and require no further interpretation. Further interpretation of function values is obtained by looking to the right, along the associated row, to identify the value interpretation that is listed in the value column corresponding to the value number.

4.2 Function constraints

The objective in introducing memory or other function constraints for these profiles is to prevent a JBIG2 T.89 encoder from overrunning a decoder. Overrun may occur just by sending many dictionaries in succession; this is clearly not appropriate for fax. It is desirable for the encoder to know not to make dictionaries too large, because this can result in decoder failures. For these reasons, it is necessary to have some targets for implementers to aim for. These proposed FAX profiles establish constraints so that there are a few fixed points (e.g. the decoder has 2 M of memory besides its page buffer) that implementers know are out there and can target their encoders at.

Function constraints

1) *Direct bitmap arithmetic coding – Template size*

   This specifies how large a template is used when doing arithmetic coding of pixels. Basically, an encoder looks at the surrounding N pixels (where $N = 10$, 13, or 16) and from that it can learn statistics of whether the current pixel, given the value of those N pixels, is going to be a 0 or a 1, and use those statistics to gain compression. If it's highly likely to be a 0, then the encoder can transmit the information "it was a 0" in very little space (a small fraction of a bit).

   Memory requirement is 2^N bytes (i.e. 1 K-64 K).

   Larger templates (larger N) are also more expensive to implement in hardware: more on-chip buffers required, more memory operations to schedule, etc.

2) *Direct bitmap arithmetic coding – AT pixel location limit*

   Of the surrounding N pixels, an encoder is allowed to specify the location of 1 or 4 of them (1 if $N = 10$ or 13, 4 if $N = 16$). However, if it is stated that "the template pixel is 55 rows above the current pixel", the encoder must then be able to buffer at least the previous 55 rows; in a hardware implementation, that buffer might have to be on-chip. JBIG2 allows the pixel to be up to 127 rows above the current row; it might be desirable to restrict that to a smaller number to reduce the number of rows required to buffer.

3) *Refinement bitmap arithmetic coding – Template size*

   When a lossy to lossless refinement is performed, the encoder is essentially transmitting the lossless version of each pixel (in some box), given all the information it knows so far. This information may include the lossy version of that same pixel, the values of surrounding lossy pixels, and the values of surrounding lossless pixels. The number of pixels that get drawn into this process, again in order to learn statistics, is $N = 10$ or 13; 2^N bytes of memory (i.e. 1 K-8 K) is needed.

4) *Refinement bitmap arithmetic coding – AT pixel location limit*

Similar to number 3: a larger number means more buffer memory required in the bitmap encoder and decoder. The $N = 10$ pixel template for refinement does not have any AT pixels at all, so if constrained to use $N = 10$, then this does not apply.

5) *Auxiliary buffers – Memory limit*

A JBIG2 T.89 encoder can instruct a decoder implementation to decode a region, such as a position block, and put the decoded bitmap into "off-screen" memory: do not draw it into the page buffer yet (it gets drawn in later after being refined). If there is only a page (or stripe) buffer, then this cannot be done. This is true even if there is some extra memory available for use, the implementation will still want to limit it. A reasonable page that uses this feature will probably need as much auxiliary buffer memory as it does page (or stripe) buffer memory.

6) *Huffman table memory*

This specifies how much memory space transmitted Huffman tables may consume in their uncompressed form. Typically 1 kbyte is all that is needed per table. Four kbytes are usually adequate to accommodate all Huffman tables.

The number of Huffman tables used to decode a symbol dictionary may vary based on the symbol region and whether refinement is being used. For symbol dictionaries:

a) when no refinement is present, up to 3 custom Huffman tables can be used: one for delta width, one for delta height, and one for transmitting the size of the MMR-coded bitmaps;

b) when refinement is present, up to 4 custom Huffman tables might be used.

For text regions:

i) when no refinement is present, up to 3 custom Huffman tables can be used, to transmit first S, delta S, delta T;

ii) when refinement is present, up to 8 custom Huffman tables can be used.

There are three types of values that are needed to decode a text region. The "First S" Huffman table is used to transmit the X coordinate (basically, if transposition is on, then it is the Y coordinate) of the first symbol in each line of text. The "Delta S" Huffman table is used to transmit the spacing between characters within a line of text. The "Delta T" Huffman table is used to transmit the spacing between lines of text.

7) *Symbol coding – Memory limit*

This limits the total amount of decoded symbol dictionary information a decoder will accommodate in memory at one time to decode a file. This limit includes two components: a fixed and a per-symbol component. The fixed component does not depend on the number of symbols, while the per-symbol component does depend on the number of symbols. The fixed component includes template size dependent variables and a constant. The per-symbol component includes the space required to accommodate the uncompressed symbol bitmaps and the overhead, such as stored width and height information and symbol ID Huffman table memory. Note that the total symbol dictionary memory "MSD" is the sum of the fixed component and all the outstanding decoded symbol dictionaries (i.e. those for which their "scope" has not elapsed or the "forget" command has not been issued).

The MSD does have dependency on whether Huffman (see Note 1) or arithmetic (see Note 2) coding is used and whether the dictionaries contain symbols or half-tone patterns (see Note 3).

The decoder symbol dictionary memory requirement shall be determined as follows:

MSD = fixed component + per-symbol component

Fixed component = 2^{direct coding template size}

+ 2^{refinement coding template size} + 8 K

$$\text{Per-symbol component} = \sum \frac{32 + R\left(W(i) \times H(i)\right)}{8} \text{ over i, i = 1 to N}$$

where:

MSD	symbol dictionary memory (in bytes)
I	index (ith symbol in dictionary)
N	number of symbols in dictionary
(W(i))	symbol width
(H(i))	symbol height
32 bytes per symbol	symbol overhead

The overhead items here are things such as: width of symbol, height of symbol, symbol ID Huffman code, length of symbol ID Huffman code, and pointer to memory where symbol bitmap resides.

R(W(i)) rounded width

W(i) rounded up to the next multiple of 32 bits (e.g. 33 rounds to 64, 128 rounds to 128).

This means that for each symbol there are 32 bytes overhead, plus H(i) rows of bitmap data, each of which is R(W(i))/8 bytes.

NOTE 1 – For Huffman coding there are no templates, so the fixed component is about 8 kbytes. The fixed component can in fact be zero if custom Huffman tables are not used.

NOTE 2 – For arithmetic coding the per-symbol component is the same. The amount of memory needed to store the decoded dictionary bitmaps (i.e. the (R(W(i)) × H(i))/8 component)) is unchanged. Differences occur in the 32 bytes per-symbol overhead component. The width, height and pointer fractions of the overhead still apply; however, the Huffman code parts do not apply. There are, however, context tables for symbol ID probability modeling that take the place of the Huffman code parts. Bottom line, 32 bytes is also a reasonable per-symbol overhead for arithmetic coding. The template options, documented in the JBIG2 FAX profiles in Table 1, range from a 10-pixel direct bitmap template with no refinement bitmap coding to a 16-pixel direct bitmap template with 13 pixels refinement bitmap template. Given this range of templates, the fixed component will range from 9 to 80 kbytes.

NOTE 3 – The same expression holds for pattern dictionaries of half-tone image regions since pattern dictionaries are similar to symbol dictionaries but contain half-tone patterns. The pattern dictionaries, however, tend to be small relative to symbol dictionaries since the pattern count is frequently low. This is only a few kbytes of memory. It is the space required by a decoder to hold the half-tone bit-planes that is of significance and determines the memory limit. This memory requirement is documented in the JBIG2 FAX profiles in Table 1, typically 110% of the resolution-dependent page buffer size (i.e. 1.0 Mbytes at 300 dpi and 2.0 Mbytes at 400 dpi).

8) *Half-tone coding – Memory limit*

Half-tones also need dictionaries, which take memory to store – but the dictionaries tend to be small, so a few kbytes is probably fine.

However, when decoding a half-tone, a decoder needs temporary space to hold the bit-planes. Say the half-tone cell is 4 × 4 pixels, 8-bit grayscale, and the half-tone covers a whole 300-dpi page. Then a decoder would need 0.5 Mbytes memory, plus its page buffer.

9) *Striping – stripe size (height limit)*

If a decoder cannot afford a full-page buffer, then this specifies how much of a stripe buffer it can afford.

Note that the use of "page" in JBIG2 translates to "stripe" within the context of MRC.

10) *Combination operators*

The "Region external combination operator" is used to indicate how this region interacts with other regions on the same page: how the overlapping pixels are combined.

The "Text region internal combination operator (SBCOMBOP)" is used to indicate how the different symbols within a text region are combined: if two symbols in a text region overlap, how the overlapping pixels are combined.

The "Half tone region internal combination operator (HCOMBOP)" is used to indicate how the different patterns within a half-tone region are combined: if two patterns in a half-tone region overlap, how the overlapping pixels are combined.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	**Terminals for telematic services**
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2002