INTERNATIONAL STANDARD

ISO/IEC 11544

First edition
1993-12-15

Information technology — Coded representation of picture and audio information — Progressive bi-level image compression

Technologies de l'information — Représentation codée de l'image et du son — Compression de l'image progressive à deux niveaux

Reference number
ISO/IEC 11544:1993(E)

Contents

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Annexes

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

International Standard ISO/IEC 11544 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 CCITT. The identical text is published as CCITT Recommendation T.82.

Annexes A, B, C, D, E and F of this International Standard are for information only.

Patents

During the preparation of this International Standard, information was gathered on patents upon which application of the standard might depend. Relevant patents were identified as belonging to the patent holders listed in annex E. However, ISO/IEC cannot give authoritative or comprehensive information about evidence, validity or scope of patent and like rights. The patent holders have stated that licences will be granted under reasonable terms. Communications on this subject should be addressed to the patent holders (see annex E).

Introduction

This Recommendation | International Standard was prepared by the Joint Bi-level Image experts Group (JBIG) of ISO/IEC JTC1/SC29/WG9 and CCITT SGVIII. The JBIG experts group was formed in 1988 to establish a standard for the progressive encoding of bi-level images.

A progressive encoding system transmits a compressed image by first sending the compressed data for a reduced-resolution version of the image and then enhancing it as needed by transmitting additional compressed data, which builds on that already transmitted. This Recommendation | International Standard defines a coding method having progressive, progressive-compatible sequential, and single-progression sequential modes and suggests a method to obtain any needed low-resolution renditions. It has been found possible to effectively use the defined coding and resolution-reduction algorithms for the lossless coding of greyscale and color images as well as bi-level images.

0.1 General characteristics

This Specification defines a method for lossless compression encoding of a bi-level image (that is, an image that, like a black-and-white image, has only two colors). The defined method can also be used for coding greyscale and color images. Being adaptive to image characteristics, it is robust over image type. On scanned images of printed characters, observed compression ratios have been from 1,1 to 1,5 times as great as those achieved by the MMR encoding algorithm (which is less complex) described in Recommendations T.4 (G3) and T.6 (G4). On computer generated images of printed characters, observed compression ratios have been as much as 5 times as great. On images with greyscale rendered by halftoning or dithering, observed compression ratios have been from 2 to 30 times as great.

The method is bit-preserving, which means that it, like Recommendations T.4 and T.6, is distortionless and that the final decoded image is identical to the original.

The method also has "progressive" capability. When decoding a progressively coded image, a low-resolution rendition of the original image is made available first with subsequent doublings of resolution as more data is decoded. Note that resolution reduction is performed from the higher to lower resolution layers, while decoding is performed from the lower to higher resolution layers. The lowest resolution image sent in a progressive sequence is a sequentially coded image. In a single-progression sequential coding application, this is the only image sent.

Progressive encodings have two distinct benefits. One is that with them it is possible to design an application with one common database that can efficiently serve output devices with widely different resolution capabilities. Only that portion of the compressed image file required for reconstruction to the resolution capability of the particular output device has to be sent and decoded. Also, if additional resolution enhancement is desired, for say, a paper copy of an image already on a CRT screen, only the needed resolution-enhancing information has to be sent.

The other benefit of progressive encodings is that they can provide subjectively superior image browsing (on a CRT) for an application using low-rate and medium-rate communication links. A low-resolution rendition is transmitted and displayed rapidly, and then followed by as much resolution enhancement as desired. Each stage of resolution enhancement builds on the image already available. Progressive encoding can make it easier for a user to quickly recognize the image as it is being built up, which in turn allows the user to interrupt the transmission of the image.

Let D denote the number of doublings in resolution (called differential layers) provided by the progressive coding. Let I_D denote the highest resolution image and let its horizontal and vertical dimensions in pixels be X_D and Y_D. Let R_D denote the sampling resolution of the image I_D.

This Specification imposes almost no restrictions on the parameters R_D, X_D, Y_D, or D. Choices such as 400 or 200 dpi (dots-per-inch) for the resolution R_D of the highest resolution layer result in a hierarchy of resolutions commensurate with current facsimile standards. Choosing R_D as 600 or 300 dpi gives a progressive hierarchy more compatible with the laser printer resolutions available as of the writing of this Specification.

It is anticipated that D will typically be chosen so that the lowest resolution is roughly 10 to 25 dpi. Typical bi-level images when reduced to such a resolution are not legible, but nonetheless such low-resolution renditions are still quite useful and function as automatically generated icons. Page layout is usually apparent and recognition of particular pages that have been seen before at higher resolution is often possible.

As mentioned above, this Specification does not restrict the number D of resolution doublings. It can be set to 0 if progressive coding is of no utility, as is the case, for example, in hardcopy facsimile. Doing so retains JBIG's compression advantage over MMR (and in fact usually increases it somewhat), while eliminating the need for any buffering and simplifying the algorithm. Single-progression sequential JBIG coding has potential applications identical to those of MMR coding. Images compressed by a single-progression sequential encoder will be readable by decoders capable of progressive decoding, although only the lowest resolution version of a progressively encoded image will be decodable by a single-progression sequential decoder.

It is possible to use this Specification for the lossless coding of greyscale and color images by coding bit-planes independently as though each were itself a bi-level image. This approach to the coding of greyscale and color images can be used as an alternative to the photographic encoding specification CCITT Rec. T.81 | ISO/IEC 10918-1 (JPEG) in its lossless mode. Preliminary experimental results have shown that JBIG has a compression advantage over JPEG in its lossless mode for greyscale images up to 6 bits-per-pixel. For 6 to 8 bits-per-pixel the compression results have been similar for both JBIG and JPEG. This Specification makes provision for images with more than one bit plane, but makes no recommendation on how to map greyscale or color intensities to bit-planes. Experimentally, it has been found that for greyscale images a mapping via Gray-coding of intensity is superior to a mapping via simple weighted-binary coding of intensity.

0.2 Stripes and data ordering

When it is necessary to distinguish progressive coding from the more traditional form of image coding in which the image is coded at full resolution from left to right and top to bottom, this older form of coding will be referred to as "sequential". The advantage of sequential coding over progressive coding is that no page (frame) buffer is required. Progressive coding does require a page buffer at the next-to-highest resolution because lower resolution images are used in coding higher resolution images.

It is possible to create a JBIG datastream with only a lowest resolution layer and this can be named single-progression sequential coding. In such coding, a full-resolution image is coded without reference to any differential resolution layers. The parameters D (mentioned in 0.1) is set equal to zero. It should be noted that in a progressive encoding of an image, the lowest resolution layer is actually encoded in single-progression sequential coding. If a full-resolution image is encoded using single-progression sequential coding, it will not be possible to decode the image progressively.

Coding in the progressive-compatible sequential mode is said to be "compatible" with coding in the progressive mode because the datastreams created (encoder) or read (decoder) in either mode carry exactly the same information. All that changes with a switch from progressive to progressive-compatible sequential encoding is the order in which parts of the compressed data are created by the encoder. All that changes with a switch from progressive to progressive-compatible sequential decoding is the order in which these parts are used by the decoder.

This compatibility is achieved by breaking an image into smaller parts before compression. These parts are created by dividing the image in each of its resolution "layers" into horizontal bands called "stripes." Progressive-compatible sequential coding does require a "stripe" buffer (much smaller than a page buffer) and additional individual "state" memory used for adaptive entropy coding of each resolution layer and bit plane.

Figure Intro. 1 shows such a decomposition when there are three resolution layers, three stripes per layer, and only one bit plane. Table Intro. 1 shows defined ways to sequence through the nine stripes.

Notice that in addition to the progressive-versus-sequential distinction that is carried by the **SEQ** bit, there is also a resolution-order distinction that is carried by the **HITOLO** bit. Encoders work from high resolution downward and so most naturally encode the stripes in **HITOLO** order. Decoders must build up the image from low resolution and so most naturally process stripes in the opposite order. When an application uses an encoder that sends progressively coded data directly to a decoder, one or the other must buffer to invert the order. When an application includes a database, the database (with appropriate set-up) can be used to buffer and invert the order (including setting **HITOLO** correctly) thereby removing this requirement from the encoder and decoder.

A stripe has a vertical size that is typically much smaller than that of the entire image. The number L_0 of lines per stripe at the lowest layer is another free parameter. As an example, L_0 might be chosen so that a stripe is about 8 mm. If such a choice is made, the number S of stripes in an image of a business-letter-sized sheet of paper will be about 35.

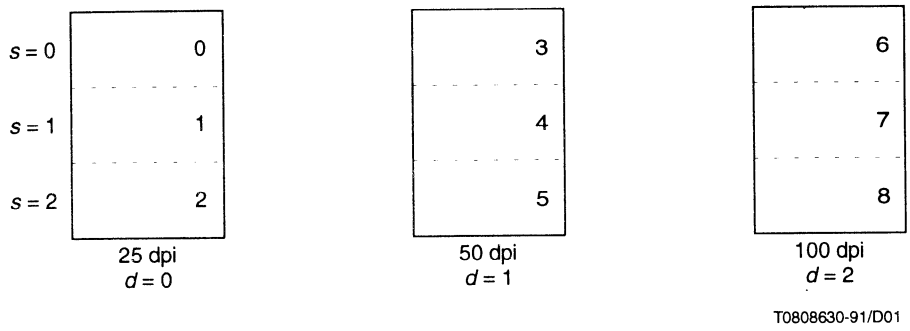

Figure 0.1 – Decomposition in the special case of 3 layers, 3 stripes, and 1 bit plane

Table 0.1 – Possible bi-level data orderings

HITOLO	SEQ	Example order
0	0	0,1,2 3,4,5 6,7,8
0	1	0,3,6 1,4,7 2,5,8
1	0	6,7,8 3,4,5 0,1,2
1	1	6,3,0 7,4,1 8,5,2

When there is more than one bit plane, as in Figure 0.2, there are twelve defined stripe orderings. Table 0.2 lists them. As before, the **HITOLO** bit carries the resolution-order distinction, and the **SEQ** bit carries the progressive-versus-sequential distinction. When the **ILEAVE** bit is 1, it indicates the interleaving of multiple bit planes. When the **SMID** bit is 1, it indicates *s*, the index over the stripe, is in the middle as shown more clearly in Table 11 of 6.2.4.

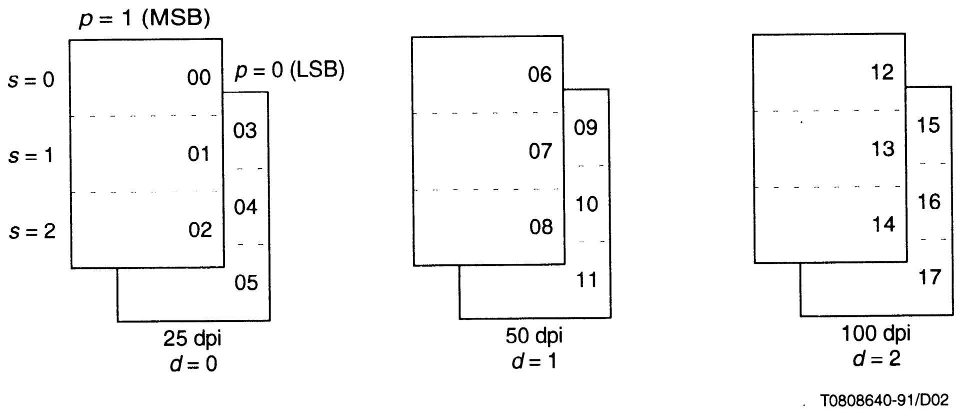

Figure 0.2 – Decomposition in the special case of 3 layers, 3 stripes, and 2 bit planes

Table 0.2 – Possible multi-plane data orderings

HITOLO	**SEQ**	**ILEAVE**	**SMID**	Example order
0	0	0	0	(00,01,02 06,07,08 12,13,14) (03,04,05 09,10,11 15,16,17)
0	0	1	0	(00,01,02 03,04,05) (06,07,08 09,10,11) (12,13,14 15,16,17)
0	0	1	1	(00,03 01,04 02,05) (06,09 07,10 08,11) (12,15 13,16 14,17)
0	1	0	0	(00,06,12 03,09,15) (01,07,13 04,10,16) (02,08,14 05,11,17)
0	1	0	1	(00,06,12 01,07,13 02,08,14) (03,09,15 04,10,16 05,11,17)
0	1	1	0	(00,03 06,09 12,15) (01,04 07,10 13,16) (02,05 08,11 14,17)
1	0	0	0	(12,13,14 06,07,08 00,01,02) (15,16,17 09,10,11 03,04,05)
1	0	1	0	(12,13,14 15,16,17) (06,07,08 09,10,11) (00,01,02 03,04,05)
1	0	1	1	(12,15 13,16 14,17) (06,09 07,10 08,11) (00,03 01,04 02,05)
1	1	0	0	(12,06,00 15,09,03) (13,07,01 16,10,04) (14,08,02 17,11,05)
1	1	0	1	(12,06,00 13,07,01 14,08,02) (15,09,03 16,10,04 17,11,05)
1	1	1	0	(12,15 06,09 00,03) (13,16 07,10 01,04) (14,17 08,11 02,05)

The two new variables **ILEAVE** and **SMID** plus the two earlier variables **HITOLO** and **SEQ** make it possible to index all twelve of these orders. The other four of the sixteen possible combinations for these four binary variables have no stripe ordering associated with them. If there is only one plane, stripe order is not dependent on **ILEAVE** and **SMID** and their values are inconsequential.

The compressed data $C_{s, d, p}$ for stripe *s* of resolution layer *d* of bit-plane *p* is independent of stripe ordering. All that changes as **HITOLO**, **SEQ**, **ILEAVE** and **SMID** vary is the order in which the data is concatenated onto a datastream. This is the compatibility feature noted earlier.

For simplicity, the remainder of this introduction will assume there is only one bit plane and the subscript *p* denoting bit plane will be dropped from $C_{s, d, p}$.

0.3 Encoder functional blocks

An encoder can be decomposed as shown in Figure 0.3. (In single-progression sequential coding only the lowest-resolution-layer encoder would be used.)

Although conceptually there are *D* algorithmically identical differential-layer encoders as shown in Figure 0.3, some implementations may choose to recursively use only one physical differential-layer encoder.

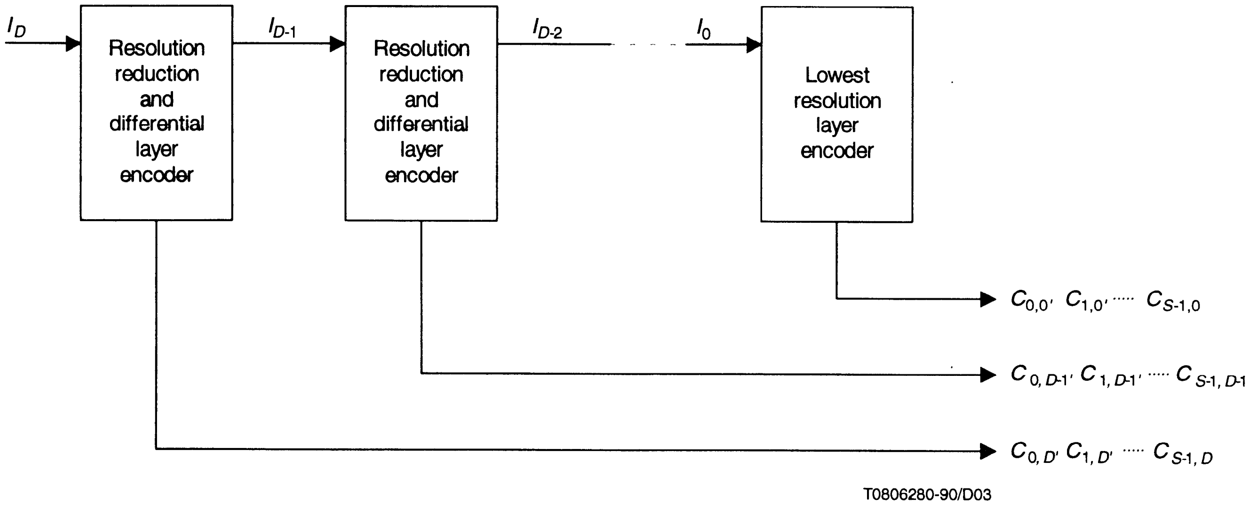
I_D
Resolution reduction and differential layer encoder
I_{D-1}
Resolution reduction and differential layer encoder
I_{D-2}
I_0
Lowest resolution layer encoder
$C_{0,0}, C_{1,0}, \cdots C_{S-1,0}$
$C_{0,D-1}, C_{1,D-1}, \cdots C_{S-1,D-1}$
$C_{0,D}, C_{1,D}, \cdots C_{S-1,D}$
T0806280-90/D03

Figure 0.3 – Decomposition of encoder

0.3.1 Resolution reduction and differential layer encoder

Each of the resolution-reduction-and-differential-layer-encoder blocks of Figure 0.3 is identical in function, hence only a description of the operation at one layer is needed. For such a description there are only two resolution layers involved. For simplicity in the remainder of this subclause, the incoming image will be referred to as the "high-resolution" image and the outgoing image, as the "low-resolution" image. Note though that the "high" and "low" resolution images of any particular resolution-reduction-and-differential-layer-encoder block in Figure 0.3 are not in general the highest and lowest resolution images of the entire system.

A resolution-reduction-and-differential-layer-encoder block of Figure 0.3 can itself be decomposed into sub-blocks as shown in Figure 0.4. Not all sub-blocks need be used in all systems. Refer to the tables in clause 4 for a definition of signal names.

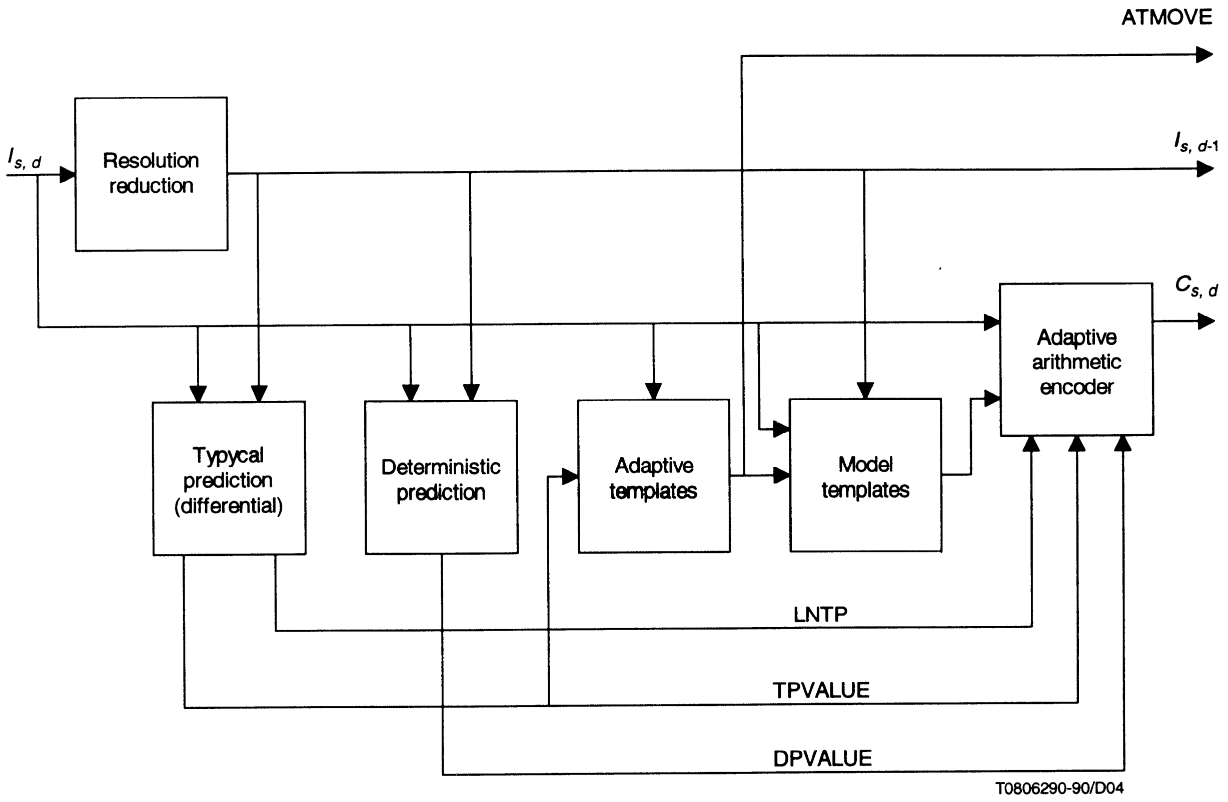

Figure 0.4 – Resolution reduction and differential layer encoder

Acronyms for the processing blocks of this figure and some others to be discussed in this introductory clause are given in Table 0.3.

Table 0.3 – Acronyms for processing blocks

Acronym	Meaning
AAD	Adaptive Arithmetic Decoder
AAE	Adaptive Arithmetic Encoder
AT	Adaptive Templates
DP	Deterministic Prediction
MT	Model Templates
RR	Resolution Reduction
TPB	Typical Prediction (Bottom)
TPD	Typical Prediction (Differential)

0.3.1.1 Resolution reduction

The resolution-reduction (RR) block performs resolution reduction. This block accepts a high-resolution image and creates a low-resolution image with, as nearly as possible, half as many rows and half as many columns as the original.

An obvious way to reduce the resolution of a given image by a factor of two in each dimension is to subsample it by taking every other row and every other column. Subsampling is simple, but creates images of poor subjective quality, especially when the input image is bi-level.

For bi-level images containing text and line drawings, subsampling is poor because it frequently deletes thin lines. For bi-level images that contain halftoning or ordered dithering to render greyscale, subsampling is poor because greyness is not well preserved, especially if the dithering period is a power of two, as is frequently the case.

This Specification suggests a resolution reduction method. This particular method has been carefully designed, extensively tested, and found to achieve excellent results for text, line art, dithered greyscale, halftoned greyscale, and error-diffused greyscale.

0.3.1.2 Differential layer typical prediction

The differential-layer typical prediction (TP) block provides some coding gain, but its primary purpose is to speed implementations. Differential-layer TP looks for regions of solid color and when it finds that a given current high-resolution pixel for coding is in such a region, none of the processing normally done in the DP, AT, MT, and AAE blocks is needed. On text or line-art images, differential-layer TP usually makes it possible to avoid coding over 95% of the pixels. On bi-level images rendering greyscale, processing savings are significantly smaller.

0.3.1.3 Deterministic prediction

The purpose of the deterministic-prediction (DP) block is to provide coding gain. On the set of test images used in the development of this Specification it provided a 7% gain, and such a gain is thought to be typical.

When images are reduced in resolution by a particular resolution reduction algorithm, it sometimes happens that the value of a particular current high-resolution pixel to be coded is inferable from the pixels already known to both the encoder and decoder, that is, all the pixels in the low-resolution image and those in the high-resolution image that are causally related (in a raster sense) to the current pixel. When this occurs, the current pixel is said to be deterministically predictable. The DP block flags any such pixels and inhibits their coding by the arithmetic coder.

DP is a table driven algorithm. The values of particular surrounding pixels in the low-resolution image and causal high-resolution image are used to index into a table to check for determinicity and, when it is present, obtain the deterministic prediction. DP tables are highly dependent on the particular resolution reduction method used. Provision is made for an encoder to download DP tables to a decoder if it is using a private resolution reduction algorithm. If an application requires default DP, decoders need to always have the default DP tables and no DP tables need be sent. Hence, if the suggested resolution reduction algorithm is used, no DP table need ever be sent.

0.3.1.4 Model templates

For each high-resolution pixel to be coded, the model-templates (MT) block provides the arithmetic coder with an integer called the context. This integer is determined by the colors (binary levels) of particular pixels in the causal high-resolution image, by particular pixels in the already available low-resolution image, and by the spatial phase of the pixel being coded. "Spatial phase" describes the orientation of the high-resolution pixel with respect to its corresponding low-resolution pixel.

The arithmetic coder maintains for each context an estimate of the conditional probability of the symbol given that context. The greatest coding gain is achieved when this probability estimate is both accurate and close to 0 or 1. Thus, good templates have good predictive value so that when the values of the pixels in the template are known, the value of the pixel to be coded is highly predictable.

0.3.1.5 Adaptive templates

The adaptive-templates (AT) block provides substantial coding gain (sometimes as much as 80%) on images rendering greyscale with halftoning. AT looks for periodicity in the image and on finding it changes the template so that the pixel preceding the current pixel by this periodicity is incorporated into the template. Such a pixel has excellent predictive value.

Such changes are infrequent, and when one occurs, a control sequence (indicated symbolically by **ATMOVE** in Figure 0.4) is added to the output datastream. Hence, decoders need not do any processing to search for the correct setting for AT.

0.3.1.6 Adaptive arithmetic encoder

The adaptive-arithmetic-encoder (AAE) block is an entropy coder. It notes the outputs of the TP and DP blocks to determine if it is even necessary to code a given pixel. Assuming it is, it then notes the context and uses its internal probability estimator to estimate the conditional probability that the current pixel will be a given color. Often the pixel is highly predictable from the context so that the conditional probability is very close to 0 or 1 and a large entropy coding gain can be realized.

Maintaining probability estimates for each of the contexts is a non-trivial statistical problem. A balance must be struck between obtaining extremely accurate estimates and the conflicting need of adapting quickly to changing underlying statistics.

0.3.2 Lowest resolution layer encoder

Figure 0.5 shows a lowest-resolution-layer encoder. It is conceptually simpler than the differential-layer encoder because the RR and DP blocks are not applicable and the PT, AT, and MT blocks are different since there is no lower resolution layer to be used as input. Refer to the tables in clause 4 for a definition of signal names. (Not all sub-blocks need to be used in all systems.)

Lowest-resolution-layer TP like differential-layer TP is primarily intended to speed processing. The algorithms used for the two versions of TP are quite different, however, and it is not possible to skip as high a percentage of pixels with lowest-resolution-layer TP as it is with differential-layer TP. On images with text and line art, lowest-resolution-layer TP allows skipping about 40% of the pixels.

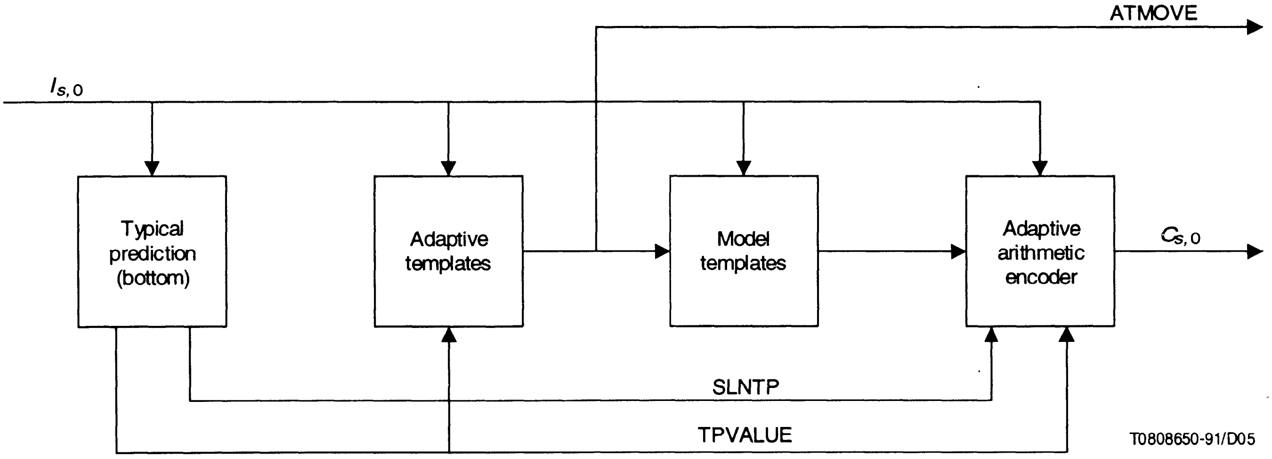

Figure 0.5 – Lowest-resolution-layer encoder

0.4 Decoder functional blocks

Figures 0.6, 0.7 and 0.8 are analogous to Figures 0.3, 0.4 and 0.5 but show decoding rather than encoding. Note that the RR and AT blocks do not appear in the decoder. Refer to the tables in clause 4 for a definition of signal names. In single-progression sequential coding only lowest-resolution-layer-decoder block of Figure 0.6 would be used. Not all sub-blocks in Figures 0.7 and 0.8 need be used in all systems.

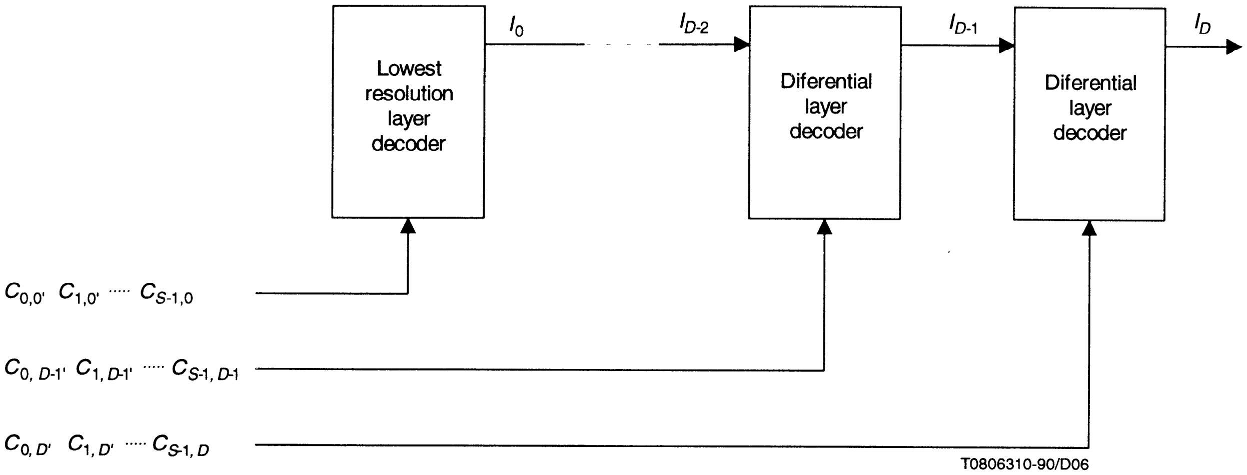
Lowest resolution layer decoder
Diferential layer decoder
Diferential layer decoder
I_0
I_{D-2}
I_{D-1}
I_D
$C_{0,0}, C_{1,0}, \cdots C_{S-1,0}$
$C_{0,D-1}, C_{1,D-1}, \cdots C_{S-1,D-1}$
$C_{0,D}, C_{1,D}, \cdots C_{S-1,D}$
T0806310-90/D06

Figure 0.6 – Decomposition of decoder

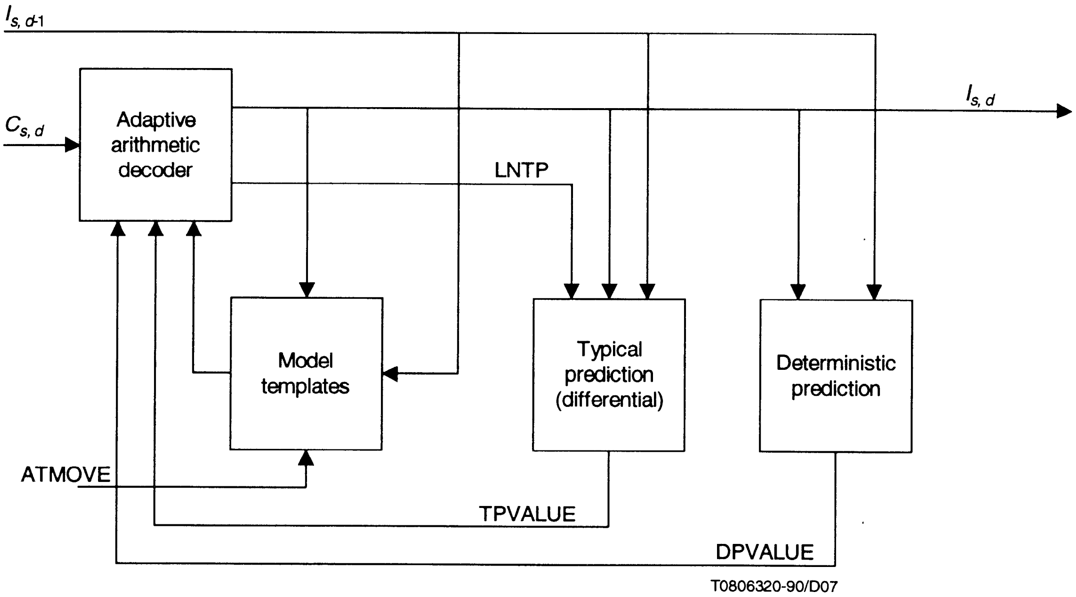

Figure 0.7 – Differential layer decoder

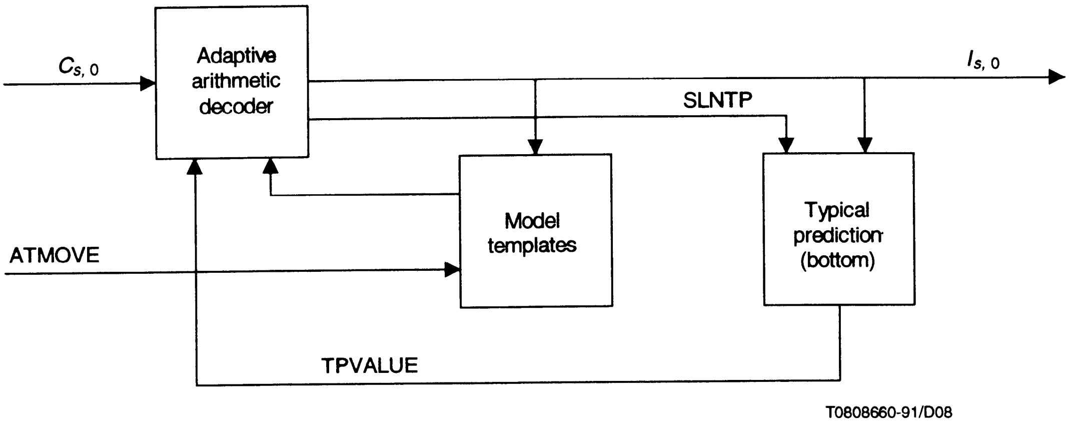

Figure 0.8 – Lowest-resolution-layer decoder

INTERNATIONAL STANDARD

CCITT RECOMMENDATION

INFORMATION TECHNOLOGY – CODED REPRESENTATION OF PICTURE AND AUDIO INFORMATION – PROGRESSIVE BI-LEVEL IMAGE COMPRESSION

1 Scope

This Recommendation | International Standard defines a bit-preserving (lossless) compression method for coding image bit-planes and is particularly suitable for bi-level (two-tone, including black-white) images.

2 Normative references

There are no normative references. Informative references to standards and to the technical literature are listed in Annex F.