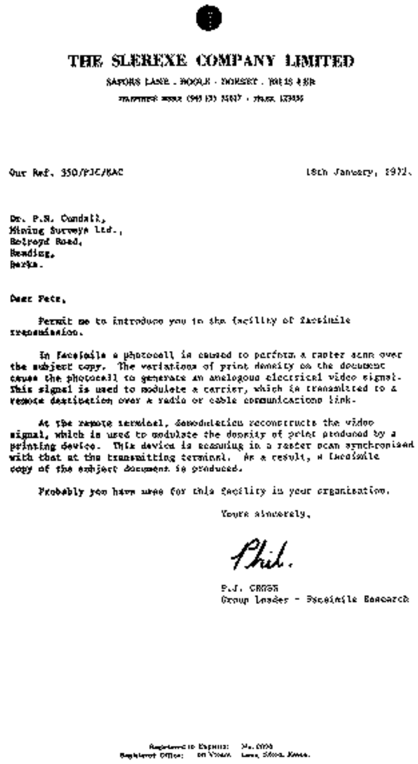
Facsimile

Nasir Memon - Northern Illinois University
Khalid Sayood - University of Nebraska-Lincoln

Introduction

A **facsimile** (**fax**) image is formed when a document is raster scanned by a light sensitive electronic device which generates an electrical signal with a strong pulse corresponding to a dark dot on the scan line and a weak pulse for a white dot. In digital fax machines, the electrical signal is subsequently digitized to two levels and processed, before transmission over a telephone line. Modern digital fax machines partition a page into 2376 scan lines, with each scan line comprising of 1728 dots. A fax document can, therefore, be viewed as a two-level image of size 2376×1728, which corresponds to 4,105,728 bits of data. The time required to transmit this raw data over a 4800 bits/sec telephone channel would be more than 14 minutes! Transmitting a 12 page document would require almost three hours. Clearly this is unacceptable. In order to reduce the bit rates some form of compression technique is required. Imposing the more realistic constraint of one minute of transmission time per page, leads us to the requirement of encoding a fax image at 0.07 bits per pixel, for a **compression ratio** of almost 15:1. Fortunately, fax images contain sufficient redundancies and even higher then 15:1 compression can be achieved by state of the art compression techniques.

Facsimile image compression provides one of the finest examples of the importance of the development of efficient compression technology in modern day communication. The field of facsimile image transmission has seen explosive growth in the last decade. One of the key factors behind this proliferation of fax machines has been the development and standardization of effective compression techniques. In the rest of this section we describe the different approaches that have been developed for the compression of fax data. For the purpose of discussion we classify the compression techniques into five different categories and give one or two representative schemes for each. We then describe international standards for facsimile encoding, the development of which have played a key role in the establishment of



THE SLEREXE COMPANY LIMITED

Our Ref. 350/PJC/EAC

18th January, 1972.

Dr. P.N. Cundall,
Mining Surveys Ltd.,
Holroyd Road,
Reading,
Berks.

Dear Pete,

Permit me to introduce you to the facility of facsimile transmission.

In facsimile a photocell is caused to perform a raster scan over the subject copy. The variations of print density on the document cause the photocell to generate an analogous electrical video signal. This signal is used to modulate a carrier, which is transmitted to a remote destination over a radio or cable communications link.

At the remote terminal, demodulation reconstructs the video signal, which is used to modulate the density of print produced by a printing device. This device is scanning in a raster scan synchronised with that at the transmitting terminal. As a result, a facsimile copy of the subject document is produced.

Probably you have uses for this facility in your organisation.

Yours sincerely,

Phil.

P.J. CROSS
Group Leader - Facsimile Research

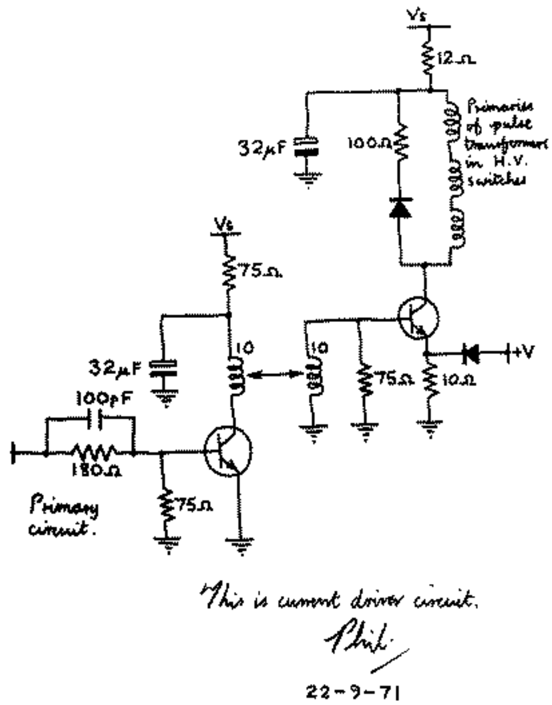


Figure 1: Example documents from the CCITT group 3 test images

facsimile transmission as we know it today. Finally, we conclude with recent progress and anticipated future developments.

Facsimile Compression Techniques

Over the last three decades numerous different techniques have been developed for the compression of facsimile image data. For the purpose of discussion we classify such compression techniques into five different categories - 1) One-dimensional coding, 2) Two-dimensional techniques, 3) Multi-level techniques, 4) Lossy techniques and 5) Pattern Matching techniques . We discuss each approach in a separate sub-section and describe one or two representative schemes.

One-dimensional coding

In figure 1 we show as examples two documents that are typically transmitted by a fax machine. One property that clearly stands out is the clustered nature of black (b) and white (w) pixels. The b and w pixels occur in bursts. It is precisely this property that is exploited by most facsimile compression techniques. A natural way to exploit this property is by **Run Length Coding**, a technique used in some form or the other by a majority of the earlier schemes for facsimile image coding.

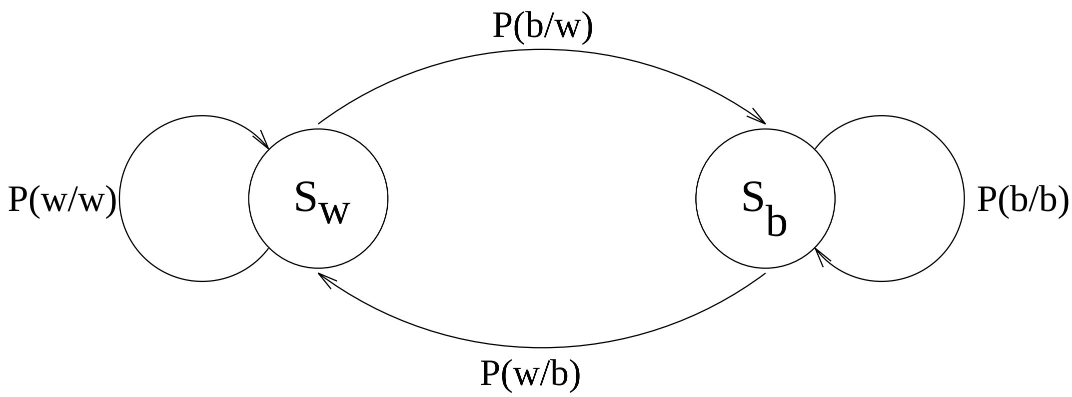


Figure 2: The Capon model for binary images.

In run length coding, instead of coding individual pixels, the lengths of the runs of pixels of the same color are encoded, following an encoding of the color itself. With a two-level image just encoding alternating runs is sufficient. In order to efficiently encode the run lengths we need an appropriate model. A simple way to obtain a model for the black and white runs is by regarding each scan line as being generated by the first order Markov process shown in figure 2, known as the *Capon model* [Capon, 59] for binary images. The two states, S_w and S_b shown in the figure represent the events that the current pixel is a white pixel or a black pixel respectively. $P(w/b)$ and $P(b/w)$ represent *transition probabilities.* $P(w/b)$ is the probability of the next pixel being a white pixel when the current pixel is black and $P(b/w)$ is the vice-versa. If we denote the probabilities $P(w/b)$ and $P(b/w)$ by t_w and t_b respectively, then the probability of a run of length r_k in a state s is given by

$$P(r_k|s) = t_s(1-t_s)^{r_k-1} \qquad s \in \{S_w, S_b\}$$

which gives us a *geometric distribution* for the run lengths. The expected run length of black and white runs then turns out to be $\frac{1}{t_b}$ and $\frac{1}{t_w}$ respectively. The geometric distribution has been found to be an appropriate model for the run lengths encountered in special classes of facsimile images like weather maps [Kunt and Johnsen 80]. However, for more structured documents like letters that contain printed text, it turns out to be inadequate. Getting analytical models for run lengths of structured documents is difficult. In practice, models are obtained empirically by analyzing a set of typical images and optimal variable length codes are then constructed based on the statistics of run lengths in this set. Usually two distinct sets of codewords are constructed for the black and white runs as the statistics for the two are found to be significantly different. The extra cost involved in maintaining two separate code tables is worth the improvement in compression obtained.

Two-dimensional coding schemes

The amount of compression obtained by one-dimensional coding schemes described in the previous sub-section is usually quite limited. This is because such schemes do not take into account vertical correlations, that is the correlation between adjacent scan lines, typically found in image data. Vertical correlations are especially prominent in high resolution images that contain twice the number of scan lines per page. There have been many schemes proposed for taking vertical correlations into account. Below we discuss a few that are representative.

One way to take vertical correlations into account is by encoding pixels belonging to k successive lines simultaneously. Many different techniques of this nature have been proposed in the literature, including *block coding*, *cascade division coding*, *quad-tree encoding* etc (for a review see [Kunt and Johnsen 80, Yasuda 80]). However, such techniques invariably fail to utilize correlations that occur across the boundaries of the blocks or bundles of lines that are being encoded simultaneously. A better way to exploit vertical correlations is to process pixels line by line as in one dimensional coding, and make use of the information encountered in previous scan lines in order to encode the current pixel or sequence of pixels. Below we list three such techniques that have proven to be very successful.

READ Coding Since two adjacent scan lines of a fax image are highly correlated, so are their corresponding runs of white and black pixels. Hence the run lengths of one scan line can be encoded with respect to the run lengths of the previous scan line. A number of schemes based on this approach were developed in the late 1970's. Perhaps the best known among them is the *Relative Element Address Designate (READ)* coding technique that was a part of Japan's response to a call for proposals for an international standard [Yasuda 80]. In READ coding, prior to encoding a run length, we locate five *reference pixels* on the current and previous scan line. These pixels are denoted by a_0, a_1, a_2, b_1 and b_2 respectively, and are identified as follows:

a_0 This is the last pixel whose value is known to both encoder and decoder. At the beginning of encoding each line a_0 refers to an imaginary white pixel to the left of the first actual pixel. While it often is a **transition pixel**, it does not have to be one.

a_1 This is the first transition pixel to the right of a_0.

a_2 This is the second transition pixel to the right of a_0.

b_1 This is the first transition pixel on the line above the line currently being encoded to the right of a_0 whose color is the opposite of the color of a_0.

b_2 This is the first transition pixel to the right of b_1 and on the same line as b_1.

For example, if the second row is the one being currently encoded, and we have encoded the pixels up to the second pixel, then the assignment of the different pixels is shown in Figure 3 below. Note that while both the transmitter (encoder) and receiver (decoder) know the positions a_0, b_1 and b_2, the positions a_1 and a_2 are known only to the encoder.

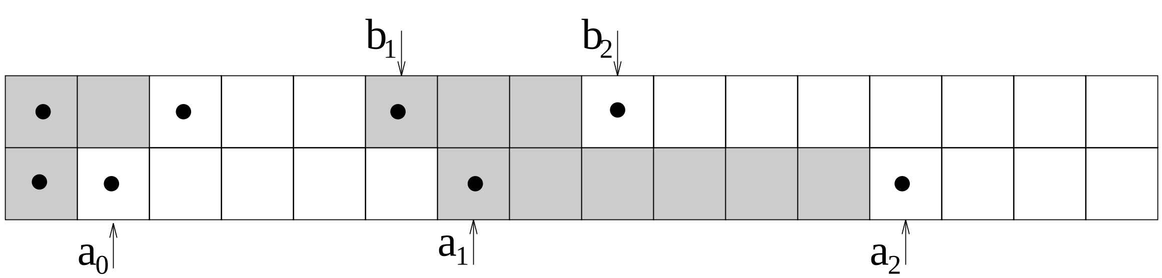


Figure 3: Two rows of an image, the transition pixels are marked with a dot.

Coding is done in one of three modes depending on the relative positions of these pixels. If the run lengths on the current and previous line are similar then the distance between a_1 and b_1 would typically be much smaller than the distance between a_0 and a_1. Hence the current length can be specified by encoding the distance (a_1, b_1). This is called *vertical mode* coding. However, when the distance between a_1 and b_1 is large, that is if there is no similar run on the previous line, then it is better to encode the runs (a_0, a_1) and (a_1, a_2) using one-dimensional run length coding. This type of encoding is known as *horizontal mode* coding. A third type of coding known as *pass mode* is performed when the condition $a_0 \leq b - 1 < b_2 < a_1$ occurs. That is we go through two runs in the previous line before completing the current run on the current line. In this case we simply advance the next pixel to be encoded to $a_0^{'}$ which is the pixel on the current line that is exactly under b_2. Before sending any run lengths, a codeword specifying the mode being used is transmitted. Additional details, including the specific codewords to be used, are given in [Yasuda 80].

Two Dimensional Predictive Coding In **predictive coding**, the image is scanned in some fixed order and a prediction is made of the current pixel based on the values of previously transmitted pixels. If the neighborhood employed to perform prediction contains pixels from both the previous and current scan lines then the technique is referred to as two-dimensional prediction. Since prediction is being made on the basis of pixels known to the receiver, only the prediction error needs to be transmitted. With binary images, the prediction error sequence is again binary, with a 0 indicating no error and a 1 indicating that an error in prediction was made. If the prediction scheme is effective then the prediction error sequence will contain many more zeroes than ones and hence can be coded more efficiently.

If we fix the neighborhood used for prediction, then given a specific image, the optimum prediction function that minimizes the probability of prediction error can be computed. However, such an optimum function varies from image to image. This fact limits the practical utility of predictive schemes. However, prediction when used as a pre-processing step can often enhance the performance of other facsimile compression techniques like run length coding [Yasuda 80].

Model Based Coding If we impose an $n'th$ order Markov model on a binary source then it's entropy is given by

$$\sum_{k=1}^{n} P(s_k)(P(x=0|s_k) \cdot \log_2 P(x=0|s_k) + P(x=1|s_k) \cdot \log_2 P(x=1|s_k))$$

where $s_1, \ldots, s_n$ are the states and x is the current pixel. When coding binary images the states s_i are simply taken to be the different bit patterns that can occur in a particular neighborhood of n pixels that occur prior to x. Given the conditional probabilities above, the source can be optimally encoded by using **arithmetic coding** [Rissanen and Langdon 79]. Note that since **Huffman coding** uses an integral number of bits to encode each source symbol. it is of little utility for encoding a binary source unless some form of alphabet extension is performed that blocks individual bits to build an extended alphabet set. Hence model based coding was not used for binary images till the early 1980's until the development of sophisticated arithmetic coding techniques that enable us to encode a source at rates arbitrarily close to it's entropy. In fact, it has been proven that model based arithmetic coding is essentially superior to any other scheme that may encode more than one bit at a time [Langdon and Rissanen 81]. In practice, however, we do not have the exact conditional probabilities needed by the model. An estimate of these can be adaptively maintained by keeping track of the counts of black and white pixels encountered so far corresponding to every state. The recently finalized JBIG standard [Hampel and Arps 92] uses model based arithmetic coding and significantly out performs the previous standards for facsimile image compression for a wide variety of test images. The compression ratio obtained is especially superior when encoding half-tone images or mixed documents that contain graphics and text [Arps and Truong 94].

Multilevel Facsimile coding

The techniques we have discussed so far can also be applied to facsimile images that have been digitized using more than two amplitude levels. An image containing 2^n gray levels, with $n \geq 2$, can be decomposed into n different bit planes each of which can then be compressed by any two-level compression technique. Better compression can be obtained if pixel intensities are expressed by using a **Gray code** representation as compared to the standard binary number representation. This is because the Gray code representation guarantees that two numbers that differ in magnitude by one will differ in their representations in only a single bit position.

The bit-plane approach for coding multi-level images can be taken to its extreme by constructing a two-level bit plane for each of the 2^n gray levels in the image. The 2^n resulting *level planes* can then be compressed by some two-level compression technique. Among the 2^n different level planes, one arbitrary one need not be encoded as it can be completely determined by the remaining $2^n - 1$ level planes. A comparison of level plane and bit plane coding has been made and it appears that level plane coding performs better than bit plane coding for images that contain a relatively small number of gray levels (typically, 2 to 4 bits per pixel) [Yasuda et. al. 85].

Another approach to coding multi-level facsimile images is to use one of the many techniques that have been developed for encoding gray scale video images. These techniques have been described in the previous section under the topic of lossless image compression. Such techniques typically perform better than bit-plane encoding and level-plane encoding when the number of gray levels present is relatively large (more than 6 bits per pixel).

Compression ratios achieved by lossless techniques are usually very modest. Typical state of the art lossless compression techniques can only achieve between 2 to 1 and 3 to 1 compression for images that have been acquired by a camera or some similar sensory device. Hence, it is quite common to use *lossy* or non-information preserving compression techniques for multi-level images. State of the art lossy techniques can easily achieve more than 15 to 1 compression while preserving excellent visual fidelity. A description of lossy techniques for multi-level images is given in a later section of this chapter.

Lossy techniques

Besides multi-level facsimile images, lossy techniques can also be used for two-level images. Two types of lossy techniques have been used on two-level images. The first type consists of a large number of pre- and post-processing techniques that are primarily used for enhancing subsequent lossless compression of two-level images. The scanning and spatial sampling process inherent in digital facsimile systems invariably leads to a high degree of jaggedness in the boundaries between black and white pixels. This jaggedness, besides reducing the visual quality of the reconstructed document also severely effects the compression ratios that can be obtained by breaking up long runs of uniform color. Hence pre-processing techniques that filter out 'noise' would not only improve picture quality but also reduce transmission time. Various such pre-processing techniques have been developed, a survey of which is given in [Yasuda 80].

A simple pre-processing technique is to remove isolated black points and bridge small gaps of white pixels between a sequence of black pixels. More sophisticated techniques employ morphological operators to modify local patterns such that subsequent compression is increased. Such techniques, however, introduce significant degradations in the image and hence require post-processing of the reconstructed image at the receiving end. This fact limits their utility in commercial systems as they require the facsimile equipment at the receiving end be equipped with circuitry to perform post-processing.

An alternative approach to reduce jaggedness in a facsimile image is by modifying the **quantizer** that is used to obtain a two-level image from electrical impulses generated while scanning a document. One such quantizer called the *notch-less bi-level quantizer* has been proposed [Yasuda 80] which adaptively adjusts the quantization level on the basis of preceding pixels. It has been shown that images obtained by using the notch-less quantizer have considerably lower entropy and better visual quality.

The second class of lossy compression techniques for facsimile image data attempt to approximate the input image by replacing patterns extracted from the image with appropriate patterns from a library. Such schemes form an important special class of facsimile image compression techniques and are discussed in the next sub-section.

Pattern Matching Techniques

Since digitized images used in facsimile transmission often contain mostly text, one way of compressing such images is to perform optical character recognition (OCR) and encode characters by their ASCII code along with an encoding of their position. Unfortunately, the large variety of fonts that may be encountered, not to mention hand written documents, makes character recognition very unreliable. Furthermore, such an approach limits documents that can be transmitted to specific languages making international communication difficult. However, an adaptive scheme that develops a library of patterns as the document is being scanned circumvents the problems mentioned above. Given the potentially high compression that could be obtained with such a technique, many different algorithms based on this approach have been proposed and continue to be investigated [Pratt et. al. 80, Johnson et. al. 83, Witten et. al. 94]

Techniques based on pattern matching usually contain a *pattern isolater* that extracts patterns from the document while scanning it in raster order. A pattern is defined to be a connected group of black pixels. This pattern is then matched with the a library of patterns that has been accumulated thus far. If no close match is formed then an encoding of the pattern is transmitted and the pattern is added to the library. The library is empty at the beginning of coding and gradually builds up as encoding progresses.

If a close match for the current pattern is found in the library then the index of the library symbol is transmitted followed by an encoding of an offset with respect to the previous pattern that is needed to spatially locate the current pattern in the document. Since the match need not be exact, the *residue*, which represents the difference between the current pattern and its matching library symbol also needs to be transmitted, if lossless compression is required. However, if the transmission need not be information preserving, then the residue can be discarded. Most practical schemes discard at least part of the residue in order to obtain high compression ratios.

Although the steps outlined above represent the basic approach there are a number of details that need to be taken care of for any specific implementation. Such details include the algorithm used for isolating and matching patterns, the encoding technique used for the patterns that do not find a close match in the library, algorithms for fast identification of the closest pattern in the library, distortion measures for closeness of match between patterns, heuristics for organizing and limiting the size of the library, etc. The different techniques reported in the literature differ in the way they tackle the issues listed above. For a good survey of such techniques, the reader is referred to [Witten et. al. 94].

A real time coder based on pattern matching was proposed by AT&T to CCITT for incorporation into the international standard [Johnsen et. al. 83]. The coder gave three times the compression given by the existing standard. The higher compression though, came at the cost of loss in quality as the scheme proposed was not information preserving.

International Standards

Several standards for facsimile transmission have been developed over the past few decades. These include specific standards for compression. The requirements on how fast the facsimile of an A4 document (210×297 mm) is transmitted has changed over the last two decades, and the the Consultative Committee on International Telephone and Telegraph (CCITT) which is a committee of the the International Telecommunications Union (ITU) of the United Nations has issued a number of recommendations based on the speed requirements at a given time. The CCITT classifies the apparatus for facsimile transmission into four groups. While several considerations are used in this classification, if we only consider the time to transmit an A4 size document over the phone lines, the four groups are described as follows:

- **Group 1** This apparatus is capable of transmitting an A4 size document in about six minutes over the phone lines using an analog scheme. The apparatus is standardized in Recommendation T.2.

- **Group 2** This apparatus is capable of transmitting an A4 document over the phone lines in about three minutes. Group 2 apparatus also use an analog scheme and therefore do not use data compression. The apparatus is standardized in Recommendation T.3.

- **Group 3** This apparatus uses a digitized binary representation of the facsimile. As it is a digital scheme it can, and does, use data compression and is capable of transmitting an A4 size document in about a minute. The apparatus is standardized in Recommendation T.4.

- **Group 4** The speed requirement is the same as Group 3. The apparatus is standardized in Recommendations T.6, T.503, T.521, and T.563.

CCITT Group 3 and 4 - Recommendations T.4 and T.6

The recommendations for Group 3 facsimile include two coding schemes; a one dimensional scheme and a two dimensional scheme. In the 1-D coding mode a run length coding scheme is used to encode alternating white and black runs on each scan line. The first run is always a white run. If the first pixel is a black pixel, then we assume that we have a white run of length zero. A special end-of-line (EOL) code is transmitted at the end of every line. Separate Huffman codes are used for the black and white runs. Since the the number of run lengths is high, instead of generating a Huffman code for each run length r_l, the run length is expressed in the form

$$r_l = 64 * m + t \qquad \text{for } t = 0, 1 \ldots, 63, \text{ and } m = 1, 2, \ldots, 27 \tag{1}$$

A run length r_l is then represented by the codes for m and t. The codes for t are called the **terminating codes**, and the codes for m are called the **make-up codes**. If $r_l < 63$ then only a terminating code needs to be used. Otherwise both a make-up code and a terminating code are used. This coding scheme is generally referred to as a **Modified Huffman (MH)** scheme. The specific codewords to be used are prescribed by the standard and can be found in a variety of sources including [Hunter and Robinson 80]. One special property of the codewords is that a sequence of six zeroes cannot result no matter how they are concatenated. Hence the codeword 0000001 is used to indicate end-of-line.

For the range of m and t given above, lengths of up to 1728 can be represented, which is the number of pixels per scan line in an A4 size document. However, if the document is wider, the recommendations provide for those with an optional set of thirteen codes. The optional codes are the same for both black and white runs.

The 2-D encoding scheme specified in the Group 3 standard is known as the Modified READ (MR) coding. It is essentially a simplification of the READ scheme described earlier. In modified READ the decision to use the horizontal mode or the vertical mode is made based on the distance a_1b_1. If $|a_1b_1| \leq 3$ then the vertical mode is used, else the horizontal mode is used. The codec also specifies a k-factor that no more than $k - 1$ successive lines are two-dimensionally encoded. k is 2 for documents scanned at low resolution and 4 for high resolution documents. This prevents vertical propagation of bit errors to no more than k lines.

The Group 4 encoding algorithm as standardized in CCITT recommendation T.6, is identical to the two dimensional encoding algorithm in recommendation T.4. The main difference between T.6 and T.4 from the compression point of view is that T.6 does not have a one dimensional coding algorithm, which means that the restriction specified by the k-factor as described in the previous paragraph is also not present. This slight modification of the modified READ algorithm has earned it the name *Modified Modified READ (MMR)*! Besides, the Group 4 encoding algorithm also does away with the end-of-line code which was intended to be a form of redundancy to avoid image degradation due to bit errors. Another difference in the Group 4 algorithm is the ability to encode lines having more than 2623 pixels. Such run lengths are encoded by using a mark-up code(s) of length 2560 and a terminating code of length less than 2560. The terminating code itself may consist of mark-up and terminating codes as specified by the Group 3 technique.

Handling transmission errors If facsimile images are transmitted over the existing switched telephone network, techniques for handling transmission errors are needed. This is because an erroneous bit causes the receiver to interpret the remaining bits in a different manner. With the 1-D Modified Huffman coding scheme, re-synchronization can quickly occur. Extensive studies of the 're-synchronization period' for the Group 3 1-D coding schemes have been made. It was shown that in most cases the Huffman code specified resynchronizes quickly, with the number of lost pixels typically being less than 50. For a document scanned at high resolution this corresponds to a length of 6.2 mm on a scan line.

To handle transmission errors, CCITT has defined an optional error limiting mode and an error correcting mode. In the error limiting mode, which is used only with MH coding, each line of 1728 pixels is divided into 12 groups of 144 pixels each. A 12 bit header is then constructed for the line indicating an all white group with a 0 and a nonwhite group with a one. The all white groups are not encoded and the nonwhite groups are encoded separately by using MH. This technique limits the effect of bit errors from propagating through an entire scan line.

The error correction mode breaks up the coded data stream into packets and attaches an error detecting code to each packet. Packets received in error are re-transmitted as requested by the receiver but only after the entire page has first been transmitted. The number of re-transmissions for any packet is restricted to not exceed four.

The JBIG Standard

The Joint Bi-Level Image Processing Group (JBIG) is a joint experts group of the International Standards Organization (ISO), International Electro-technical Commission (IEC) and the CCITT. This experts group was jointly formed in 1988 to establish a standard for the progressive encoding of bi-level images. The JBIG standard can be viewed as a combination of two algorithms, a **progressive transmission** algorithm, and a lossless compression algorithm. Each of these can be understood independently of the other.

Lossless Compression The lossless compression algorithm uses a simple **context model** to capture the structure in the data. A particular arithmetic coder is then selected for each pixel based on its **context**. The context is made up of neighboring pixels. For example, in Figure 4 the pixel to be coded is marked **X** while the pixels to be used as the context are marked **O** or **A**. The **A** and **O** pixels are previously encoded pixels and are available to both encoder and decoder. The **A** pixel can be moved around in order to better capture any structure that might exist in the image. This is especially useful in half-toned images in which the **A** pixels are used to capture the periodic structure. The location and movement of the **A** pixel is transmitted to the decoder as side information.

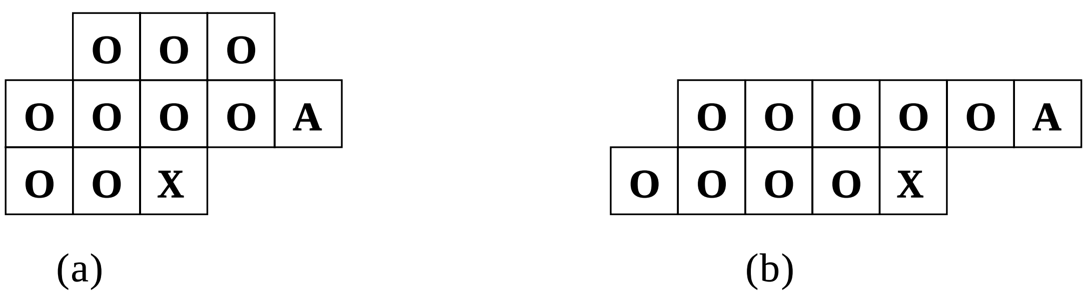


Figure 4: (a) Three line and (b) Two line model template for lowest resolution layer

The Arithmetic coders specified in the JBIG standard is a special binary adaptive Arithmetic coder known as the QM coder. The QM coder is a modification of an adaptive binary Arithmetic coder called the Q coder [Pennebaker and Mitchell 88], which in turn is an extension of another binary adaptive Arithmetic coder called the *skew* coder [Langdon and Rissanen 1981] Instead of dealing directly with the 0s and 1s put out by the source, the QM coder maps them into a More Probable Symbol (MPS) and Less Probable Symbol (LPS).

If 1 represents black pixels, and 0 represents white pixels, then in a mostly black image, 1 will be the MPS, while in an image with mostly white regions 0 will be the MPS. In order to make the implementation simple, the JBIG committee recommended several deviations from the standard Arithmetic coding algorithm. The update equations in arithmetic coding that keep track of the sub-interval to be used for representing the current string of symbols involve multiplications which are expensive in both hardware and software. In the QM coder expensive multiplications are avoided and re-scalings of the interval take the form of repeated doubling, which corresponds to a left shift in the binary representation. The probability q_c of the LPS for context C is updated each time a rescaling takes place and the context C is active. A ordered list of values for q_c is kept in a table. Every time a rescaling occurs, the value of q_c is changed to the next lower or next higher value in the table, depending on whether the rescaling was caused by the occurrence of an LPS or MPS. In a non-stationary situation, it may happen that the symbol assigned to LPS actually occurs more often than the symbol assigned to MPS. In this situation, the assignments are reversed; the symbol assigned the LPS label is assigned the MPS label and vice versa. The test is conducted every time a rescaling takes place. The decoder for the QM coder operates in much the same way as the encoder, by mimicking the encoder operation.

Progressive Transmission In progressive transmission of an image a low resolution representation of the image is first sent. This low resolution representations requires very few bits to encode. The image is then updated, or refined, to the desired fidelity by transmitting more and more information. In order to encode an image for progressive transmission, we need to create a sequence of progressively lower resolution images from the original higher resolution image. The JBIG specification recommends generating one lower resolution pixel for each two by two block in the higher resolution image. The number of lower resolution images (called layers) is not specified by JBIG. However, there is a suggestion that the lowest resolution image is roughly 10 to 25 dpi. There are a variety of ways in which the lower resolution image can be obtained from a higher resolution image, including sampling and filtering. The JBIG specification contains a recommendation against the use of sampling. The specification provides a table based method for resolution reduction. The table is indexed by the neighboring pixels shown in Figure 5 in which the circles represent the lower resolution layer pixels and the squares represent the higher resolution layer pixels.

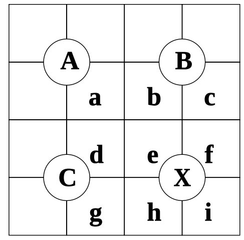


Figure 5: Pixels used to determine value of lower level pixel.

Each pixel contributes a bit to the index. The table was formed by computing the expression

$$4e + 2(b + d + f + h) + (a + c + g + i) - 3(B + C) - A$$

If the value of this expression is greater than 4.5 the pixel X is tentatively declared to be 1. The table has certain exceptions to this rule to reduce the amount of edge smearing, generally encountered in a filtering operation. There are also exceptions that preserve periodic patterns and dither patterns.

When the progressive mode is used for transmission, information from lower resolution layers can be used to improve compression. This is done by including pixels from lower resolution layers in the context used to encode a pixel in the current layer. The contexts used for coding the lowest resolution layer are those shown in Figure 4. The contexts used in coding the higher resolution layer are shown in Figure 6. Ten pixels are used in each context. If we include the two bits required to indicate which context template is being used, twelve bits will be used to indicate the context. This means that we can have 4096 different contexts.

The standard does not impose any restrictions on, D, the number of resolution layers that are constructed. Indeed, D can be set to zero if progressive coding is of no utility. In this case, coding is said to be *single-progression sequential*, or just *sequential*. The algorithm allows some degree of compatibility between the progressive and sequential modes. Images that have been encoded in a progressive manner can be decoded sequentially, that is, as just one layer. Images that have been encoded sequentially, however, cannot be decoded progressively. This compatibility between progressive and sequential modes is achieved by partitioning an image into *stripes*, with each stripe representing a sequence of image rows with user defined height. If the image has multiple bit-planes then stripes from each bit-

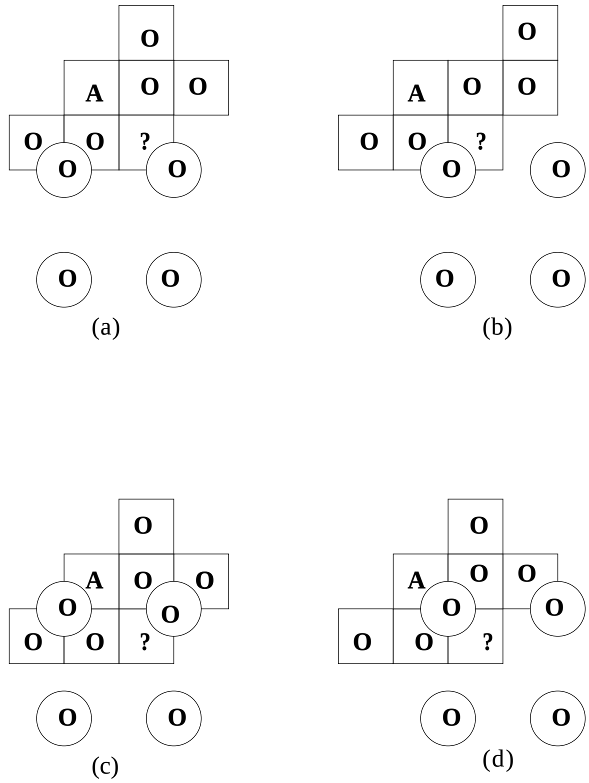


Figure 6: Contexts used in the coding of higher resolution layers

plane can be interleaved. Each stripe is separately encoded, with the user defining the order in which these stripes are concatenated into the output data stream.

Comparison of MH, MR, MMR, and JBIG

In the previous sub-section we have seen four different facsimile coding algorithms that are part of different international standards. As we might expect the JBIG algorithm performs better than the MMR algorithm which performs better than the MR algorithm, which in turn performs better than the MH algorithm. The level of complexity also follows the same trend, though one could argue that MMR is actually less complex than MR. A comparison of the schemes for a some facsimile sources is shown in Table 1.

The Modified READ algorithm was used with $K = 4$, while the JBIG algorithm was used with an adaptive 3 line template and adaptive arithmetic coder to obtain the results in this table. As we go from the one dimensional MH coder to the two dimensional MMR coder we get a factor of two reduction in file size for the sparse text sources. We get even further

Source Description	Original Size (pixels)	MH (bytes)	MR (bytes)	MMR (bytes)	JBIG (bytes)
Letter	4352×3072	20605	14290	8531	6682
Sparse Text	4352×3072	26155	16676	9956	7696
Dense Text	4352×3072	135705	105684	92100	70703

Table 1: Comparison of binary image coding schemes (Arps, 1994)

reduction when we use an adaptive coder and an adaptive model as is true for the JBIG coder. When we come to the dense text, the advantage of the 2 dimensional MMR over the one dimensional MH is not as significant, as the amount of two dimensional correlation becomes substantially less.

The compression schemes specified in T.4 and T.6 break down when we try to use them to encode half tone images. This is to be expected as the model that was used to develop these coding schemes is not valid for half-tone images. The JBIG algorithm, with its adaptive model and coder suffers from no such drawbacks, and performs well for half-tone images as well [Arps and Truong 94].

Future trends

The next decade will see continued progress in the development of facsimile technology. Future developments anticipated include proliferation of color facsimile, integration of facsimile equipment with personal computers, penetration of fax machines into the home market, telepublishing and even distribution of newspapers through fax machines. Compression technology is expected to play a key role in these developments as new techniques have to be designed and incorporated into international standards.

For example, an immediate challenge that stands before the international community is the establishment of a compression standard for color facsimile. Although few color fax machines exist today, technological developments in printing technology are expected to bring their price down to reasonable levels.

References

[1] Ronald Arps and Thomas Truong. Comparison of international standards for lossless still image compression. *Proceedings of the IEEE*, 82(6):889–899, June 1994.

[2] J. Capon. A probablistic model for run-length coding of pictures. *IRE Transactions on Information Theory*, pages 157–163, December 1959.

[3] H. Hampel and R. B. Arps et. al. Technical features of the JBIG standard for progressive bi-level image compression. *Signal Processing*, 4(2):103–111, 1992.

[4] R. Hunter and A. H. Robinson. International digital facsimile standards. *Proceedings of the IEEE*, 68(7):855–865, 1980.

[5] O. Johnsen, J. Segen, and G. L. Cash. Coding of two-level picturs by pattern matching and substitution. *Bell Systens Technical Journal*, 62(8):2513–2545, 1983.

[6] G. G. Langdon Jr. and J. Rissanen. Compression of black-white images with arithmetic coding. *IEEE Transactions on Communications*, COM-29(6):858–867, 1981.

[7] Murat Kunt and Ottar Johnsen. Block coding of graphics: A tutorial review. *Proceedings of the IEEE*, 68(7):770–786, July 1980.

[8] W. Pratt, P. Capitant, W. Chen, E. Hamilton, and R. Wallis. Combined symbol matching facsimile data compression system. *Proceedings of the IEEE*, 68(7):786–796, 1980.

[9] J. J. Rissanen and G. G. Langdon. Arithmetic coding. *IBM J. Res. Dev.*, 23(2):149–162, 1979.

[10] I. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Van Nostrand Reinhold, 1994.

[11] Y. Yasuda, Y. Yamakazi, T. Kamae, and K. Kobayashi. Advances in fax. *Proceedings of the IEEE*, 73(4):707–731, 1985.

[12] Yasuhiko Yasuda. Overview of digital facsimile coding techniques in japan. *Proceedings of the IEEE*, 68(7):830–845, 1980.

Further Reading

FAX - Facsimile Technology and Applications Handbook, 2'nd edition by K. McConnell, D. Bodson and R. Schaphorst published by Artech House, 685 Canton Street, Norwood, MA 02062, USA, is an excellent single source on various aspects of facsimile technology, including compression.

Two comprehensive surveys by Yasuhiko Yasuda et. al. on coding techniques for facsimile have appeared in the Proceedings of the IEEE in 1980 and 1985 (references 11 and 12 above) respectively. These surveys summarize most of the research that has been conducted on facsimile coding and contain an extensive list of references. Besides, the two issues that they appear in, July 1980 and April 1985 are both special issues on facsimile coding.

For a description of the CCITT standards, the best sources are the original documents containing the recommendations

1. Standardization of Group 3 Facsimile Apparatus for Document Transmission, Recommendation T.4, 1980.
2. Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus, Recommendation T.6, 1984.
3. Progressive Bi-level Image Compression, Recommendation T.81, 1992. Also appears as ISO/IEC International Standard 11544: 1993

These documents can be ordered from ITU, The International Telecommunication Union, Place Des Nations 1211, Geneva 20, Switzerland. They are also available from Omnicom, Phillips Business Information, 1201 Seven Locks Road, Suite 300, Potomac, Maryland 20854, U.S.A., Fax: 1-800-666-4266.

A more recent survey by Arps and Huang (reference 1 above) compares the performance of different standards.

Defining Terms

Compression Ratio Size of Original Data / Size of Compressed Data.

Facsimile The process by which a document is optically scanned, and converted to electrical signals.

Facsimile Image The quantized digital image corresponding to the document that has been input to a facsimile machine.

Fax Abbreviation for facsimile.

Gray code A binary code for integers in which two integers that differ in magnitude by one differ in only one bit position.

Group 3 Facsimile apparatus capable of transmitting an A4 size document in about a minute. The apparatus is standardized in Recommendation T.4.

Group 4 Facsimile apparatus for sending a document over public data networks with virtually error-free reception. Standardized in Recommendations T.6, T.503, T.521, and T.563.

JBIG The Joint Bi-Level Image Processing Group of the International Standards Organization (ISO), International Electro-technical Commission (IEC) and the CCITT. This experts group was jointly formed in 1988 to establish a standard for the progressive encoding of bi-level images. The term JBIG is also used to refer to the coding algorithm proposed by this committee.

Modified Huffman code (MH) One-dimensional coding scheme used by Group 3 equipment.

Modified READ code (MR) Two-dimensional coding scheme used by Group 3 equipment.

Modified Modified READ code (MMR) Two-dimensional coding scheme used by Group 4 equipment.

Predictive coding A form of coding where a prediction is made for the current event based on previous events and the error in prediction is transmitted.

Progressive transmission A form of transmission in which a low resolution representation of the image is first sent. The image is then updated, or refined, to the desired fidelity by transmitting more and more information.

Quantizer The process of converting analog data to digital form.