THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 8, October 1983
Printed in U.S.A.

Coding of Two-Level Pictures by Pattern Matching and Substitution

By O. JOHNSEN,* J. SEGEN,* and G. L. CASH*

(Manuscript received September 28, 1982)

A pattern matching approach is proposed for coding of two-level pictures. Patterns, which are either symbols such as characters, or fractions of black regions, such as line segments, are extracted from the facsimile. They are compared and matched to already transmitted patterns, called library patterns. If a correct match is detected, only the position of the pattern and the identification of the matching library pattern are transmitted. If a pattern does not match any library pattern, it is added to the library and its binary description is transmitted. Compared to conventional two-dimensional codes, the compression is often doubled and is sometimes 4.5 times higher. Compared to a symbol-matching coding technique,[2] the compression has increased by 20 to 80 percent, depending upon the document.

I. INTRODUCTION

Conventional two-level picture coding techniques are based on the statistical dependence between neighboring picture elements (pels).[1] The calculation of entropies, according to a local source model, gives the maximum achievable bit rates. Run length or predictive coding techniques or a combination of them takes advantage of the statistical dependence between neighboring pels and leads to bit rates close to the entropy. Each exploits what can be called the microscopic (pel) properties of a facsimile.

Pattern-recognition coding techniques exploit macroscopic proper-

* Bell Laboratories.

ties of the facsimiles. The image source is a source of patterns such as characters, lines, and black spaces. We can code the facsimile more efficiently, since the description is closer to the perceptual level. We can consider two kinds of pattern-recognition coding techniques. The first technique is pattern (or image) understanding. It recognizes a certain pattern, for example a letter, that possibly includes some font information. The second technique is pattern matching. Here, a pattern is not recognized, but is simply matched with already transmitted patterns, and if a correct match is detected, it is replaced by the matching pattern. It does not use the image-understanding level. The image-understanding approach has the potential advantage of a very high compression, but the often important aesthetic details of the documents can be lost, and there is a risk of errors at the present level of such techniques. The matching approach yields lower compression, but keeps more of the original pictorial information. There are also lower risks of errors, since matching allows only slight modifications in the pattern shapes. Naturally, neither of the pattern-recognition techniques is lossless, since they modify the picture content.

Ascher and Nagy[3] and Pratt et al.[2] have already proposed facsimile coding techniques using matching techniques. In the system presented here, not only the symbols, as in Pratt's case, but also graphical elements such as line segments and black regions are matched. The patterns are efficiently coded and updated, leading to significantly higher compressions.

II. SYSTEM DESCRIPTION

Figure 1 shows the block diagram of the system. The pattern locator examines the facsimile line by line. When it locates a black pel, the pattern isolator picks up a pattern. The pattern is either a symbol (defined as a set of black pels completely surrounded by white pels) or, when no symbol can be extracted, a fraction of the black region. Therefore, contrary to Ref. 2, there is no residue to be coded, since all black pels belong to a pattern.

The matcher makes a template matching of the incoming pattern, with existing library patterns to determine whether the incoming pattern is similar to an already transmitted pattern. The system screens the library patterns to reduce the time-consuming template matching. Thus, we consider only the patterns that might match the incoming pattern. We screen by comparing features of the library patterns with those of the incoming pattern. We apply a very efficient and simple two-pass screening. If a correct match is detected, the matcher sends the information about the position of the pattern and its library identification to the coder. If no match has occurred, the incoming pattern is added to the pattern library. The pattern library

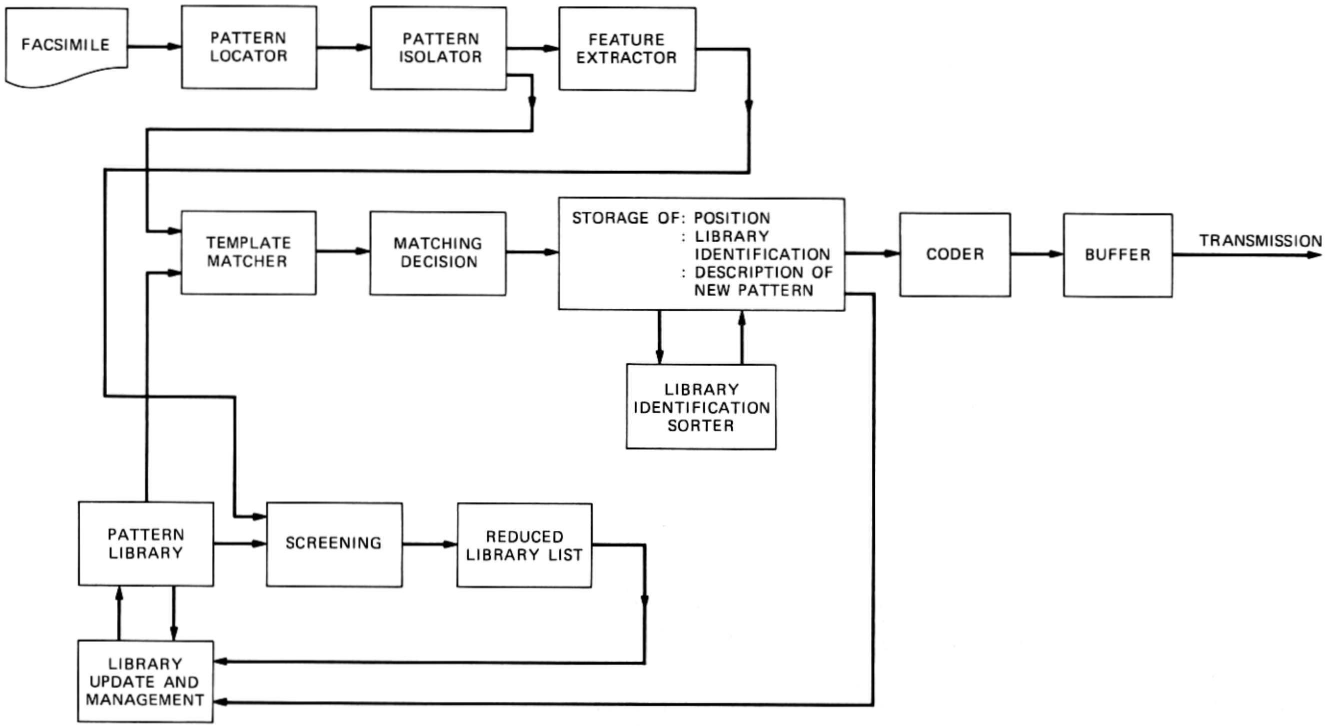


Fig. 1—Block diagram of facsimile coder by pattern matching.

is empty at the beginning of the coding and is gradually built up by the incoming library patterns. The matcher then also sends the information about the position and description of the new library patterns to the coder.

A library update and management unit takes care of the addition and deletion of library patterns and organizes them for the quickest possible match and most efficient coding. All the patterns isolated along one line are stored in the coder. When the end of the line is reached, we sort the patterns, which allows a more efficient coding.

III. LOCATION AND ISOLATION OF PATTERNS

Patterns, in the present context, are the primitive elements of the coding process. They are isolated, and sent to the matching block sequentially, in a raster order. We distinguish two classes of patterns, relative to a square window of a predetermined size, W.

1. A symbol is defined as a connected region consisting of black pels and completely surrounded by white pels, such that it can completely fit into the window.

2. A nonsymbol is defined as a windowed portion of a black connected region that is larger than the window.

Usually, characters and small graphics elements can be represented as symbols, while lines and larger figures can be decomposed into nonsymbols. The decomposed figures can be later reconstructed by taking the union of the nonsymbols. The nonsymbols do not have to be disjoint, and a better compression may sometimes result from a decomposition into overlapping symbols.

Decomposing large figures into nonsymbols allows us to use matching techniques to compress graphical information, as well as text. A figure can be decomposed in many ways, and the compression that results from grouping similar nonsymbols usually depends on the decomposition. The final compression, or the number of different classes of nonsymbols, can be used as a measure of quality of the decomposition, and one may try to find the best decomposition in respect to such measures. Finding the optimal decomposition, however, may be computationally quite complex (we do not know of any related study) and it would certainly require many passes through a figure. At present, we use a one-pass isolation procedure, which allows us to keep the computation within reasonable bounds.

The isolation procedure repeatedly isolates and removes the upper-left portion of a black region, up to a maximum size allowed by the window. If the isolated pattern has no black pel extensions, then it is a symbol; otherwise it is a nonsymbol.

The isolation algorithm operates on a two-dimensional one-bit array containing the original picture. The picture memory is scanned line by line from the upper-left element. When a black pel is found, the procedure attempts to trace the boundary of a black region, clockwise. The tracing algorithm is a standard one; however, we describe it here for further reference. Let us call the first black pel (x_1,y_1). The neighbors (adjacent pels in eight directions of (x_1,y_1) are being examined, beginning at (x_1+1,y_1) and searching clockwise around (x_1,y_1) up to (x_1-1,y_1+1). If a black pel is found, it becomes the second pel of the contour — (x_2,y_2); otherwise (x_1,y_1) is erased from the picture memory (single pels are neglected) and the scan continues. Each subsequent pel of the contour is found by searching around the current pel (x_i,y_i), beginning two steps clockwise from the previous pel (x_{i-1},y_{i-1}) (Fig. 2). The contour trace ends when it returns to the first pel in such a way that the next pel would by (x_2,y_2). The tracing algorithm checks for the limits of the picture array and it maintains a window. Pels beyond the limits of the picture array and those outside of the current window are always treated as white (0 valued). The purpose of the window is to restrict the maximal size of isolated pattern to $W \times W$. The window is initially set to a size $2W \times W$, and positioned in such a way that (x_1,y_1) is in the center of its upper edge. When the traced part of the boundary reaches a width of W, the window is reset to a size $W \times W$, and it is placed over the boundary part that has been traced, such that (x_1y_1) is still at the upper edge of the window (Fig. 3).

The tracing of the boundary is recorded in a two-dimensional one-bit array S in the following way. When the search around the current boundary pel (x_i,y_i) goes past the pel (x_i+1,y_i), a 1 is put in $S(x_i+1,y_i)$. If the search goes past the element (x_i-1,y_i) then a 1 is put in $S(x_i,y_i)$. All the elements of S are initially set to 0. The information in S (Fig. 4), after the trace termination, completely represents the boundary (it is a form of run-length code). The pattern now can be isolated by copying and erasing the portion of the picture that is enclosed by the boundary (including the boundary). This is accomplished using the

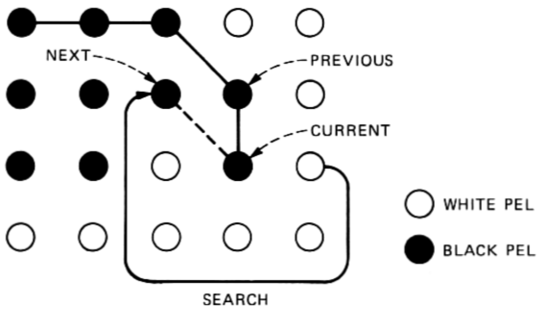


Fig. 2—Contour tracing.

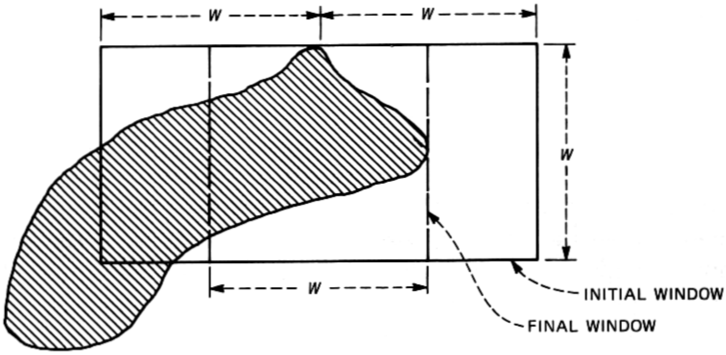


Fig. 3—Window positioning for isolation.

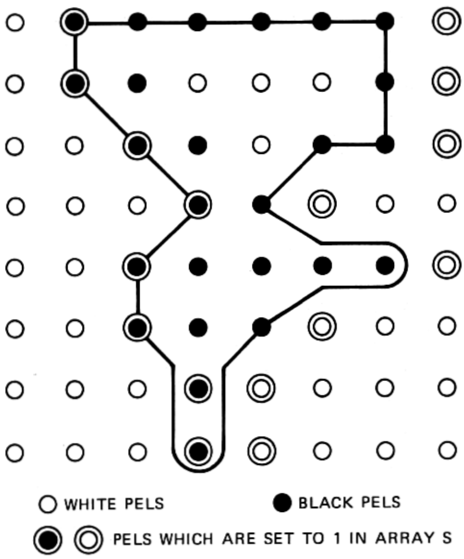


Fig. 4—Contour encoding in array S.

information in the array S. For any row of S, let $S_1, S_2 \cdots S_n$ be the position (x-coordinates) of 1-valued elements in a row. The number n is always even, which is a property of the boundary encoding that we use. For every row of S, the pixels of a corresponding row of the picture memory between S_1 and S_2, S_3 and S_4, etc., are copied to another array, and set to 0 in the picture memory, including $S_1, S_3 \cdots$ and excluding $S_2, S_4, \cdots$. The pattern is now isolated and erased from the picture memory. While the isolation algorithm described above always works correctly, i.e., it isolates symbols and completely decomposes large figures into nonsymbols, it does not attempt in any way to optimize the decomposition, so the results are not always pleasing.

To improve the decomposition in cases commonly occurring in graphics, we have added two extensions to the basic isolation scheme:

1. L-pattern suppression

L-pattern suppression improves the segmentation of large blobs that otherwise may generate many dissimilar nonsymbols (Fig. 5). This extension is implemented in the tracing phase of the isolation algorithm as follows: If the beginning part of the traced boundary goes straight down from either first or second pel over more than k (currently $k = 10$) pels, then an attempt to turn immediately to the right resets the lower edge of the window to the last pel before the right turn, so the boundary is forced to turn left (see Fig. 6).

2. Cross decomposition

If the isolated pattern can be represented as an intersection of a horizontal and a vertical line segment (a cross), then each segment becomes a separate pattern. This is implemented by comparing each isolated pattern (with the matching technique described in Section 4) to a cross formed by secting this pattern with vertical and horizontal lines one pel from the edges of the final window (Fig. 7). If a sufficiently close match is found, then one of the line segments from the cross is

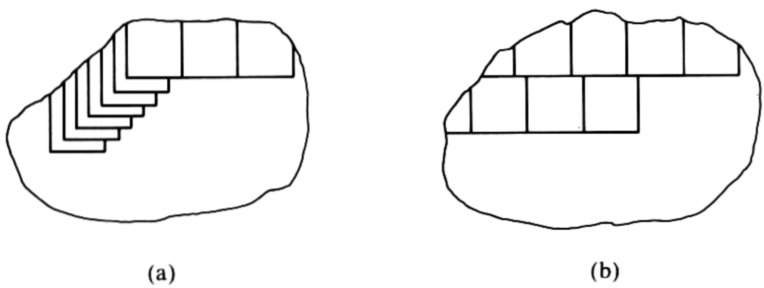


Fig. 5—Improvement in segmentation due to L-pattern suppression. (a) Before suppression. (b) After suppression.

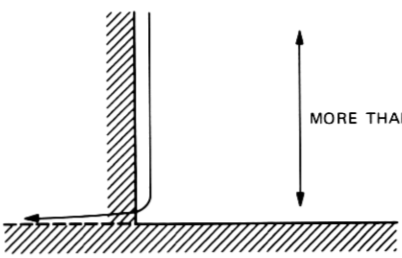


Fig. 6—L-pattern suppression. When tracing reaches the corner, it is forced to follow the dashed line.

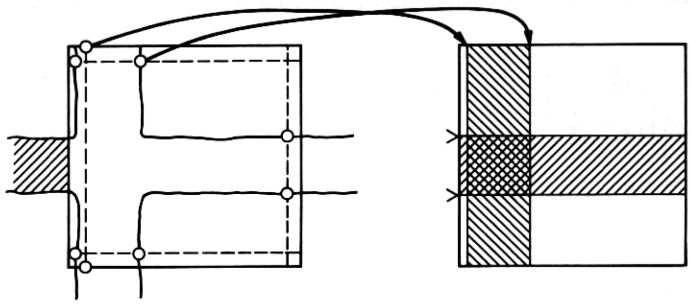
returned to the picture memory, while the other replaces the isolated pattern. This extension reduces the number of patterns generated by line crossings in grids and tables (Fig. 8).

The basic isolation algorithm is similar to the region extraction method of Dudani[4], but in contrast to the latter it does not need to store and process a list of boundary points, and it extracts regions containing holes in one pass. This algorithm can be shown to work correctly in every case and it is well suitable for a hardware implementation. The extensions of the basic algorithm are heuristic in nature, but they improve considerably the decomposition of large regions. Examples of such improvements are shown in Fig. 5 and 8. Additional improvements may be possible at some increase of the computational cost.

IV. MATCHING

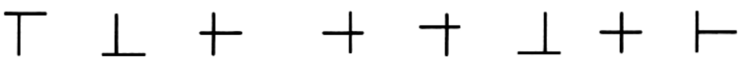
The matching includes all the processes necessary to know whether an incoming pattern matches any of the library patterns. In this system, we divide the matching into three parts.

1. The screening unit makes a selection of the library patterns, and



(a) (b)

Fig. 7—Forming an intersection pattern.



(a)

(b)

Fig. 8—Patterns resulting from grid segmentation. (a) Before cross decomposition. (b) After cross decomposition.

directs for template matching only those library patterns that might match.

2. The template matcher creates a new binary picture called error picture, containing black pels or 1's in the locations where the two template-matched patterns are dissimilar.

3. The matching decision process uses the error pictures and other information to decide whether a correct match has occurred.

4.1 Screening

The purpose of the screening is to reduce the time-consuming task of the matcher. It should direct to the template matcher only the library patterns that might match the incoming (unknown) pattern. The screening is obtained by measuring some characteristics of the patterns, called features, and comparing them. The features must be easy to compute and compare, and also must form an easily classifiable space. The digitization of a facsimile adds much noise to a pattern. To get an efficient screening, the features must also be relatively noise independent. Four features were chosen for the screening. Two of them are obvious: the pattern length and the pattern height. The two others are the number of horizontal and the number of vertical white runs enclosed in the pattern. They are characteristics of the inside of a pattern, separating, for example, c from e or o. The chosen features are shown in Fig. 9. The straightforward feature "number of black pels" was found to be of little use because of its high variability and dependency upon the other features.

The screening process also must decide in which order to send the library patterns to the matcher. The most probable match should be sent first, to reduce the number of matches. The probability of a match between patterns depends not only on the similarity of their features, but also on the probability of occurrence of a library pattern. For example, an incoming pattern having the same feature distance to an O and a Q is much more likely to match the O than the Q since O is much more frequent than Q. The screening takes into account both the feature similarity and the probability of occurrence of a library pattern. We consider the probability of occurrence by sorting the library patterns according to the number of times they have matched (see Section 5.2.1). We take the feature distance into account by allowing for each feature only a fixed margin between the two patterns. The margin must be wide enough not to preclude any correct match and tight enough to reduce the number of template matches. A two-pass screening was found very efficient. In the first screening, only library patterns with features very similar to those of the incoming patterns are sent to the template matcher. A second, much looser, screening is applied only in the few cases where no match occurred.

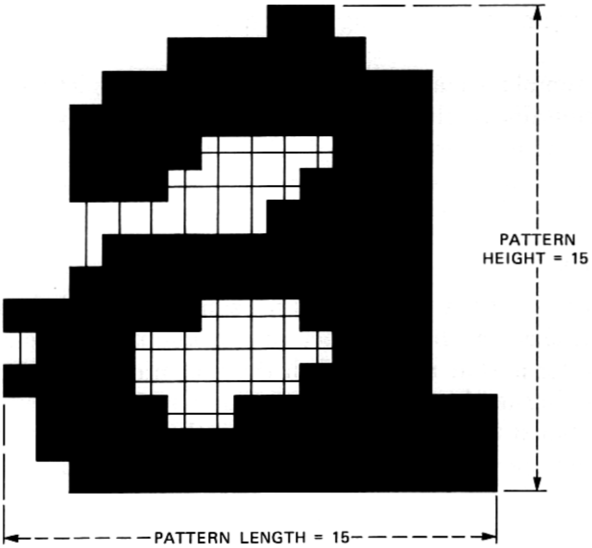


Fig. 9—Features chosen for screening. Horizontal lines indicate which runs are included in count of horizontal runs. Vertical lines indicate which runs are included in count of vertical runs. Horizontal run count is six and vertical run count 15.

The screening and the sorting are very efficient in reducing the number of matches. For example, for a typewritten document, the average number of matches per incoming pattern is reduced to 2.5, compared to 25 without screening and sorting.

4.2 Template matching

The template matcher creates a new picture called error picture, which contains 1's in the locations where the two patterns are different. The error picture is obtained simply by superimposing the two patterns and making "exclusive or" of the corresponding pels. Figure 10 is an example of matching two patterns of the same character, while Fig. 11 shows the matching of two unlike patterns. Two patterns are always matched nine times, allowing the displacement of one pattern compared to the other by ±1 in both the horizontal and vertical directions.

4.3 Matching decision

The matching decision unit must process the error picture to detect whether there is a correct match, and to decide which relative position of the library pattern gives the best match.

The straightforward approach is to count the number of errors (or 1's) in the error picture and to threshold it to make the decision. Such

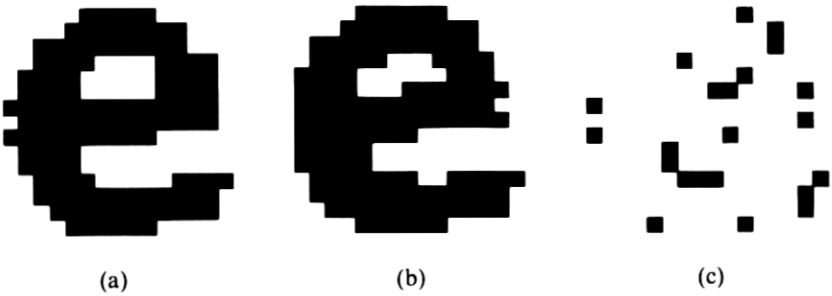


Fig. 10—Template matching of two similar patterns, with (a) and (b) original patterns and (c) error picture.

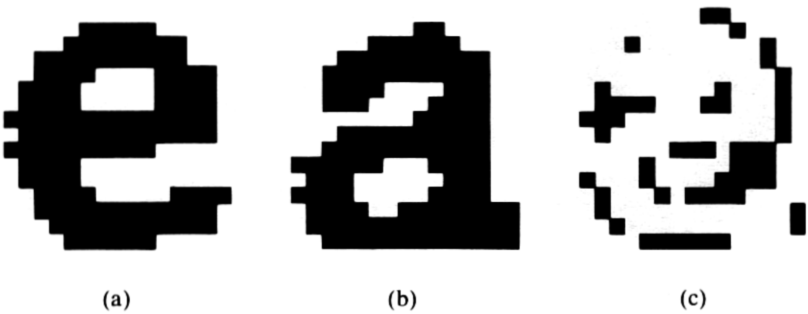
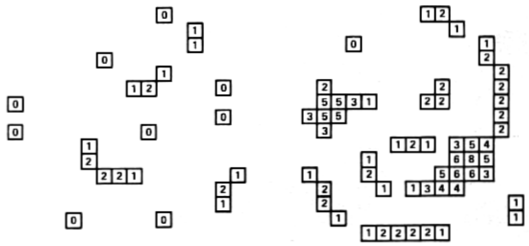


Fig. 11—Template matching of two different patterns, with (a) and (b) original pattern and (c) error picture.

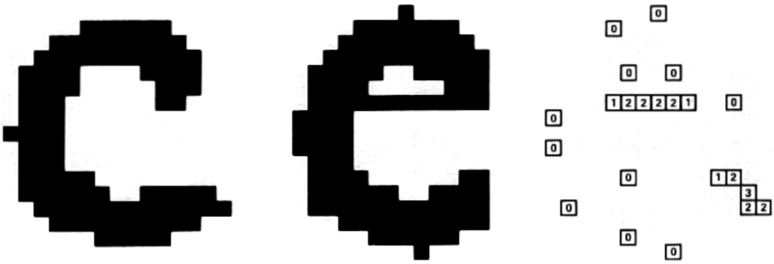
a technique would lead to many mismatches or many undetected matches, since, as shown in Ref. 2, the error count for two patterns corresponding to the same character is sometimes higher than the count for two patterns corresponding to different characters. This is caused by the digitization noise. Figure 10 shows that the template matching of two patterns of the same character gives relatively randomly distributed errors. Figure 11 shows that in the case of patterns of different characters, a cluster of errors appears where there are morphological differences between patterns.

As Ref. 2 shows, we could apply a weighted error count where the weight of an error is equal to the number of error pels among its eight neighbors. Single errors are erased and the maximum weight is eight. Figure 12 gives the weighted error pictures from the error pictures of Figs. 10 and 11. The weighted error count is not sufficient for the matching decision, as shown by Fig. 13. We must look at local error patterns to make the decision. The reason is that it is the local characteristics of the pattern that indicate whether two patterns are the same. Therefore, any decision made upon a count or integration



(a) (b)

Fig. 12—Weighted error pictures. (a) Weighted error count is 18 in Fig. 10 and (b) 144 in Fig. 11.



(a) (b) (c)

Fig. 13—A weighted error count matching criterion lead to a mismatch, with (a) and (b) original patterns, and (c) weighted error picture.

may be incorrect. The matching decision described below uses only local measures and is also made locally with the simple rule that the match is considered correct if no local rejections are detected during a template matching.

The following rule of decision is made. A match is rejected if:

Condition 1: An error pel has a weight of 4 or more, or

Condition 2: (a) an error pel has a weight of 2 or more, (b) at least two of its neighboring error pels are not connected, and (c) one of the two pels from the patterns used to obtain the error pel has a weight of 0 or 8 (corresponding to 0 or 8 surrounding black pels).

Most mismatches are detected by Condition 1, but Condition 2 is necessary in order to reject, for example, the possible match of an e and a c shown in Fig. 13. It is easy to see that Condition 2a is not necessary since it is included in 2b, but Condition 2a reduces the computation.

With these matching criterion, no visible mismatches have been detected, except slight distortion in line drawings. It is important to notice that a rejection can often be detected after processing a small fraction of the error picture. A matching decision made at the same time as the template matching would lead to an early abortion of template matchings and thus reduce the computation.

When a correct match is detected, several relative positions sometimes give a correct match. The chosen relative position will be the one with the lowest error count. The best relative position will decide where the library pattern will be put to replace the incoming pattern.

V. CODING

Contrary to many conventional facsimile coding techniques, we must code several different kinds of events and design several separate code books. The code for a pattern includes the position and the description of the pattern. The description is usually its library identification, or in the case of a new pattern, its complete description. The coding procedure is described here for the size of the International Telegraph and Telephone Consulative Committee (CCITT) test facsimiles having 1728 pels per line and 2376 lines, but it can easily be modified for other cases.

5.1 Coding of the position of the pattern

To obtain a good-quality reproduction with pattern matching, we must position the patterns accurately. Considering the CCITT test documents, 23 bits are necessary for an absolute fixed length coding (11 bits horizontally, 12 bits vertically). We choose to transmit the horizontal position uncoded (11 bits) because variable-length run-length coding would lead only to slightly smaller coding length (typically 1 to 1.5 less bits/pattern) since the horizontal distance between patterns is large. Also since the absolute horizontal position is coded, the patterns can be transmitted in a nonsequential order, which, as shown later, leads to a significant decrease in the average coding length for the library identification code words. It should be noted that with 1728 pels/line and an 11-bit code word, the code words starting with 111 are not used and therefore can be used as special code words.

We code the vertical position of the patterns in the following way:

1. A mode bit is sent at the beginning of each line to indicate whether there are any patterns starting on that line.

2. If there are no patterns on the line, operation 1 is repeated on next line.

3. If there are patterns on a line, they are all coded. The special horizontal code word 111 indicates that there are no more patterns on the line and that the next line can be considered.

4. When a pattern is replaced by a library pattern, the position of the library pattern might be moved up or down by one line. Therefore,

after the library identification has been coded, the code words 10 and 11 are used to position the library pattern up or down, while code word 0 is used to indicate no vertical displacement. No vertical displacement code word is sent with a new library pattern, since there are no changes in vertical position.

Figure 14 shows examples of the message format for the pattern positioning.

5.2 Coding of the pattern identification

The coder must send a pattern identification word with each pattern. We can transmit the pattern number uncoded. It requires, for example, seven bits in the case of a library size of 128 and nine bits in the case of a library size of 512. The coding procedure used here will lead to an average coding length of the pattern identification of fewer than five bits/pattern. It will be obtained by a continuous library updating and by variable-length coding.

5.2.1 Library updating and management

The library management and updating is done for the following purposes:

1. Accept new library patterns, and if necessary, delete a seldom used library pattern to make room for the new one.
2. Organize the library for the fastest match, taking into account the screening and matching procedures.
3. Organize the library for minimum average library identification coding length.

All three require the same processing: to keep track of the number of times each library pattern is used. By ordering the library pattern in order of decreasing usage, the correct match will be obtained rapidly,

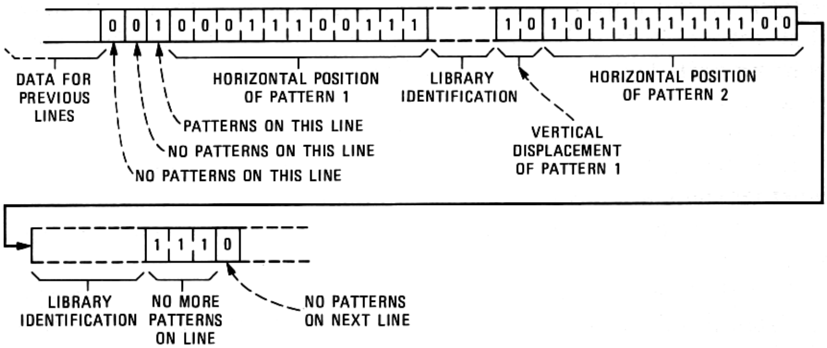


Fig. 14—Coding of positions of patterns. Two lines have no patterns, then a line has three patterns; the first on position 231 is replaced by a library pattern, the second on position 1532 is a new library pattern, There are no patterns on next line.

since the most used library patterns will be accessed first. An efficient coding of the patterns' identification is obtained by giving short code words to the first patterns in the list. The last pattern in the list, which is one of the least used patterns, can be deleted to make room for a new one.

The updating must be deterministic and use no future information, since the receiver must make the same updating to decode correctly.

The updating rule of the patterns in the library is as follows:

1. When a pattern matches a library pattern number K, that library pattern is moved to number $K/2$ and all the pattern numbers from $K/2$ to K-1 are increased by 1.

2. When a new pattern is added to the library, it gets number $N/2$ where N is the total number of library patterns. The patterns with numbers from $N/2$ to N will be increased by 1, and if N is equal to the maximum number of library patterns M, the library pattern with number $N + 1$ is dropped.

This updating procedure was found to efficiently give low identification numbers to often used patterns and high numbers to seldom used patterns. If M is the maximum number of library patterns, it guarantees that a new library pattern will stay in library for at least $M/2$ matches, but generally for many more.

5.2.2 Pattern identification coding table

The pattern identification coding table includes two special code words: "new pattern" and "same pattern." They are added to increase the coding efficiency. The "new pattern" code word is chosen because it is not necessary for a new library pattern to send an identification number, since the decoder uses the same rule as the coder to assign the identification number to the new pattern. The "same pattern" code word indicates that the transmitted pattern is the same as the previously transmitted pattern. It is useful particularly for typewritten text where the line-by-line search for a pattern often detects the same pattern (character).

The coding table for the pattern identification is given in Table I for a pattern library with a maximum of 512 patterns.

This code leads to an average library identification length of fewer than seven, compared to nine with a fixed-length code. The next section shows a more efficient coding procedure.

5.2.3 Pattern identification coding by sorting

Since an absolute code gives the horizontal position of a pattern, it is possible to transmit the patterns detected along a line in any order. The only condition is that the library updating be done at the end of the line. The average coding length of the library identification is

Table I—Coding table for identification of library patterns

Symbol	Code Word	Code Word Length
Same pattern	000	3
Library pattern 1–16	1XXXX	5
New pattern	00100	5
Library pattern 17–32	010XXXX	7
Library pattern 33–64	0011XXXXX	9
Library pattern 65–128	00101XXXXXX	11
Library pattern 129–512	011XXXXXXXXX	12

reduced to fewer than five bits, by sorting the patterns on a line according to their library number. That is because:

1. Many of the patterns are the same.
2. The library pattern identification number is run-length coded (only the increase compared to the previous identification number is coded).
3. The new library patterns are sent at the end of the line; therefore, the new pattern code word is sent only once, since any more patterns are automatically new patterns.

This can be illustrated by an example. Let a line have the following pattern: pattern 23, new pattern; pattern 28, same; pattern 23, new pattern. By looking at Table I, the coding length is 7 + 5 + 7 + 3 + 7 + 5 = 34 bits. With sorting, the patterns become: pattern 23, same; pattern 28, same; new pattern; new pattern. The coding length is 7 + 3 + 5 + 3 + 5 + 0 = 23 bits. It should be noted in this example that pattern 28 is coded as pattern 5 since only the increase in identification number compared to the previous pattern is coded.

The library updating is done at the end of each line. This creates problems when accepting new library patterns. They must be added immediately to the top of the library, since the position of the other patterns should not be changed. It is also not possible to delete patterns to make room for the new ones. For that reason, before scanning a line, enough library patterns should be deleted to avoid an overflow of the pattern library.

5.3 Coding of the library pattern description

The size of a pattern is limited to 32 × 32 bits. The description starts with a 5-bit word, which indicates the height, H, of a pattern in binary. The length of a pattern is extended to 32 pels by filling the right end with 0's. Therefore, there are $32 \times H$ pels to code. For coding efficiency, one white pel (0) is added at the beginning. A coding line is made of the $32 \times H + 1$ pels considered in the raster scan order. The reference line is similar to the coded line except that all the pels are shifted to the right by 32 pels (one line). Therefore, a line is coded using the previous line as the reference. The line is then coded by the CCITT two-dimensional code,[5] with the only modification that the first code word, which is always the horizontal mode code word, is deleted, since it doesn't give any information. For coding efficiency, it is chosen to allow switching between two modes for the coding of the library pattern description. The first mode is as described above and called "horizontal coding." The other is called "vertical coding" and is the same as above except that the pattern is coded column after column from top to bottom. Therefore, in the vertical mode the description starts with a 5-bit word indicating the length of a pattern. A header bit indicates which mode is chosen, with a 0 for horizontal mode and a 1 for vertical mode. We could also code the pattern description using a code better matched to the source. This would reduce the coding length, but at the expense of requiring a specific code in place of a standard code.

5.4 Coding summary

The coding procedure can be summarized in the following way:

1. All the patterns isolated along a scan line are matched.

2. At the end of the line, the matched patterns are sorted in order of increasing pattern identification number. The new library patterns are added at the end in sequential order.

3. The patterns are coded and transmitted with the information sent in the following order:

   a. Horizontal position of pattern.
   b. Pattern identification. If it is a new pattern, the identification is sent only for the first new pattern on the line.
   c. A 1- or 2-bit code word to specify the vertical shift of a pattern, except if it is a new library pattern.
   d. For a new library pattern the following bits are sent: (1) a header bit indicating whether the horizontal or vertical coding mode is chosen, (2) a 5-bit word indicating the number of lines of the pattern to be coded, and (3) the CCITT two-dimensional coding of the pattern (see 5.2).
   e. After all patterns on a line have been sent, the special horizontal code word 111 indicates the end of the line.
   f. The library update is made according to 5.2.3. The patterns are updated in order to increasing identification number. After updating, all patterns with a number greater than 480 are deleted, thus allowing for at least 32 new library patterns to be added on the next line.

Figure 15 is an example of message transmission. The different code words are summarized in Table II.

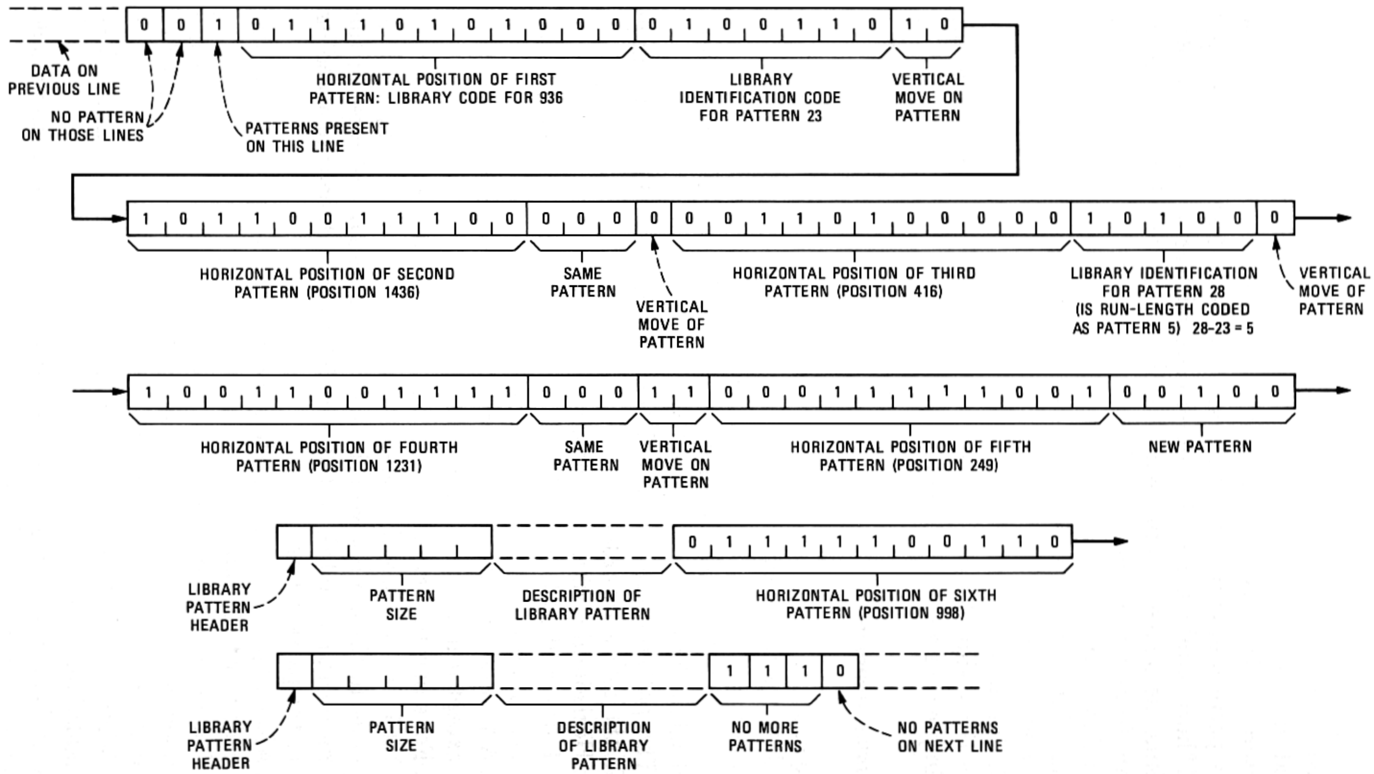


Fig. 15—Message transmission. There are two lines without patterns, next pattern 23 is in position 936, same pattern is in position 1436, pattern 28 is in position 416, same pattern is in position 1231; two new patterns are in position 249 and 998, and there are no patterns on next line.

Table II—Description of the code words for pattern matching coding

Code Definition	Word Size	Description
Mode bit	1	Indicates whether there are any patterns on the line.
Horizontal position	11	Gives in binary the absolute position of a pattern.
No more pattern	3	Indicates that there are no more patterns on the line (this code word: 111 is a special horizontal position code word).
Vertical move of pattern	1 or 2	Indicates whether the pattern must be moved up or down by one line or is not moved.
Library identification code	Variable	Defines which library pattern is transmitted.
Library pattern description header	1	Indicates whether the library pattern is coded in horizontal or vertical mode.
Library pattern size	5	
Library pattern description	Variable	Slightly modified CCITT two-dimensional code.

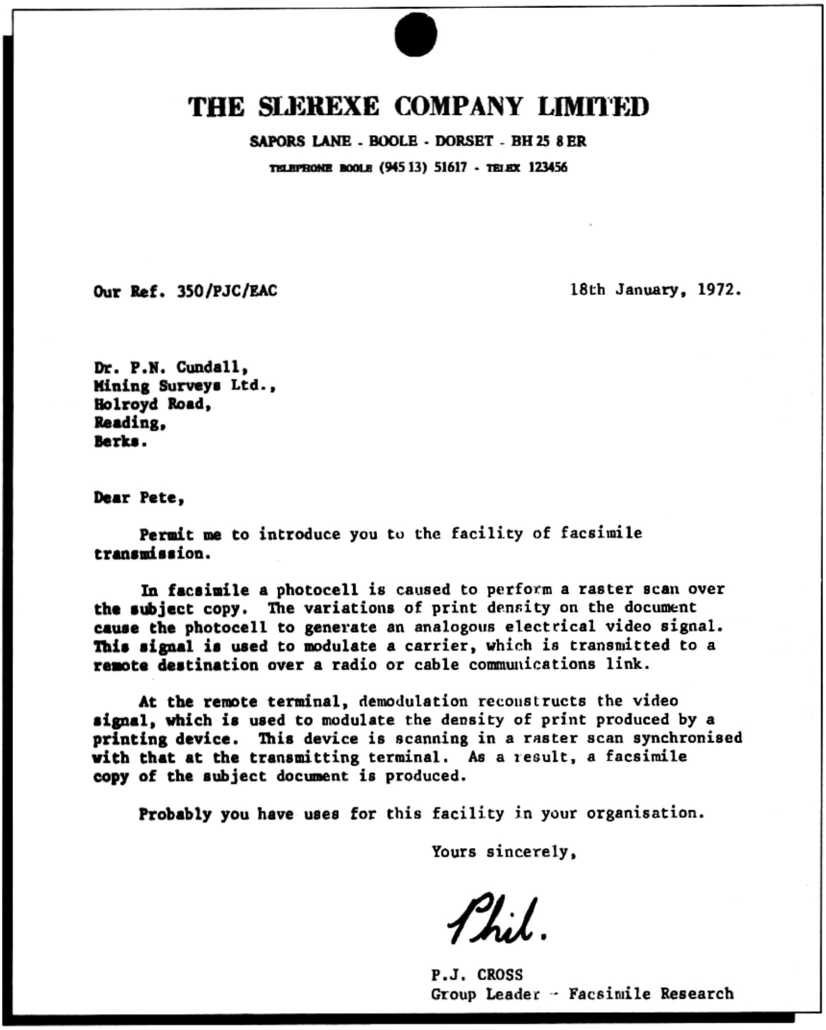
VI. SIMULATION RESULTS

The important criteria are the compression and the quality of the received documents. For that purpose, the set of eight CCITT facsimile documents are used. Their resolution is 7.7 pels/mm (200 pels/in.) in both the horizontal and vertical directions. They have 1728 pels/line and 2876 lines. Documents one, two, four and five are shown in Fig. 16. All eight documents are shown in Ref. 5. For accurate comparison with the matching technique by Pratt et al.,[2] the simulations were also made with an older nonofficial version of the CCITT documents, which is similar except each document has 1728 pels/line and 2128 lines.

6.1 Facsimile quality

In order to improve the quality of the decoded picture, a local filtering using a 3 × 3 window is applied. In addition, large library patterns are slightly expanded on their borders. This operation erases artifacts in large black regions.

The encoding scheme modifies the binary picture. We must therefore verify that the alterations are not visible or at least not annoying. We can consider three picture alterations: wrong matches, matches with a slightly distorted pattern, and wrong positioning. In the case of a wrong match, a pattern is replaced by a different pattern. The only detected wrong matches are such as between 0 and O, dot and comma, I and 1, which even people cannot recognize correctly without using the context. Therefore, it can be considered that the system has practically no wrong matches. A match with a slightly distorted pattern can occur with characters. A character might match a same character of a different font. Or a character might match a same but thinned or



THE SLEREXE COMPANY LIMITED

SAPORS LANE - BOOLE - DORSET - BH 25 8 ER

TELEPHONE BOOLE (945 13) 51617 - TELEX 123456

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,

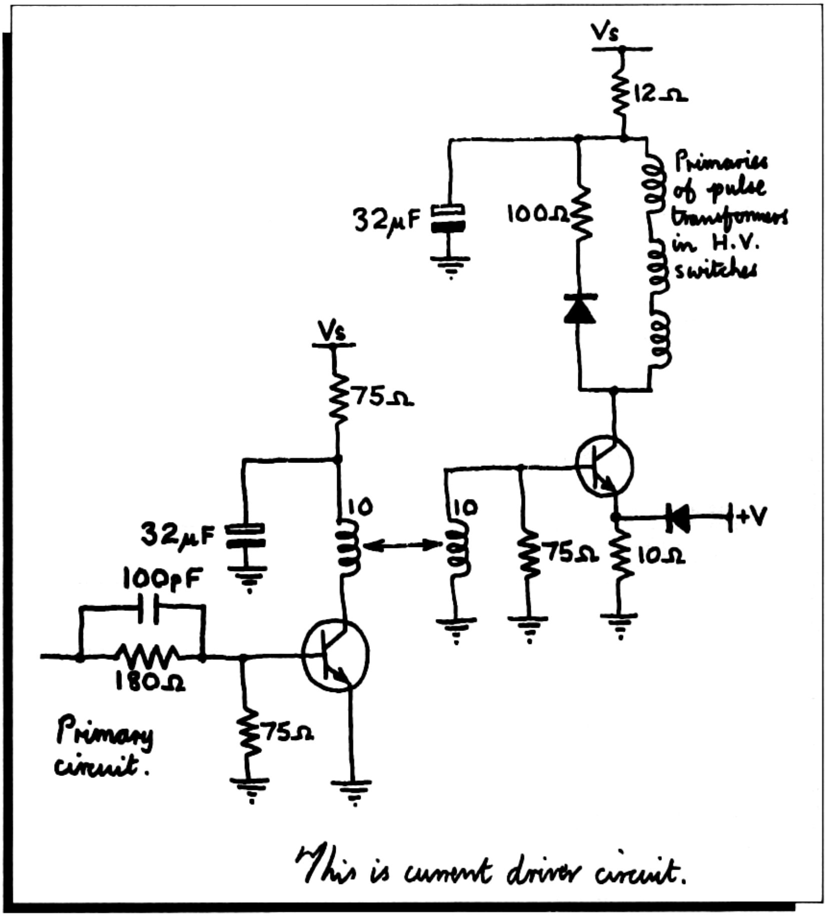
Phil.

P.J. CROSS
Group Leader - Facsimile Research

(a)

Fig. 16(a)—Original CCITT document one (first 2000 lines).

thickened character. Such matches, contrary to wrong matches, are tolerable if they don't appear too often. Such distorted matches appear when two slightly different fonts are used on a same page or when characters of a page come from a low-quality typewriter or scanner. The wrong positioning of a pattern decreases the quality of the received facsimile. No noticeable wrong positioning for patterns such as characters or other symbols is observed. Some visible wrong positionings are observed for nonsymbol patterns such as line segments, where the



(b)

Fig. 16(b)—Original CCITT document two (first 2000 lines).

successive patterns make the lines slightly jagged. Figure 17 shows the same CCITT facsimiles as Fig. 16, but after transmission by pattern matching. It can be seen that there are no significant degradations. There are some slight irregularities in line drawings, as for example in Fig. 17d. A few distorted matches appear on CCITT document one (Fig. 17a).

6.2 Compression

To make an accurate comparison with both the symbol matching and two-dimensional coding techniques, the coding simulations have been made with both the official set of CCITT facsimile documents

- 34 -

L'ordre de lancement et de réalisation des applications fait l'objet de décisions au plus haut niveau de la Direction Générale des Télécommunications. Il n'est certes pas question de construire ce système intégré "en bloc" mais bien au contraire de procéder par étapes, par paliers successifs. Certaines applications, dont la rentabilité ne pourra être assurée, ne seront pas entreprises. Actuellement, sur trente applications qui ont pu être globalement définies, six en sont au stade de l'exploitation, six autres se sont vu donner la priorité pour leur réalisation.
Chaque application est confiée à un "chef de projet", responsable successivement de sa conception, de son analyse-programmation et de sa mise en oeuvre dans une région-pilote. La généralisation ultérieure de l'application réalisée dans cette région-pilote dépend des résultats obtenus et fait l'objet d'une décision de la Direction Générale. Néanmoins, le chef de projet doit dès le départ considérer que son activité a une vocation nationale donc refuser tout particularisme régional. Il est aidé d'une équipe d'analystes-programmeurs et entouré d'un "groupe de conception" chargé de rédiger le document de "définition des objectifs globaux" puis le "cahier des charges" de l'application, qui sont adressés pour avis à tous les services utilisateurs potentiels et aux chefs de projet des autres applications. Le groupe de conception comprend 6 à 10 personnes représentant les services les plus divers concernés par le projet,et comporte obligatoirement un bon analyste attaché à l'application.

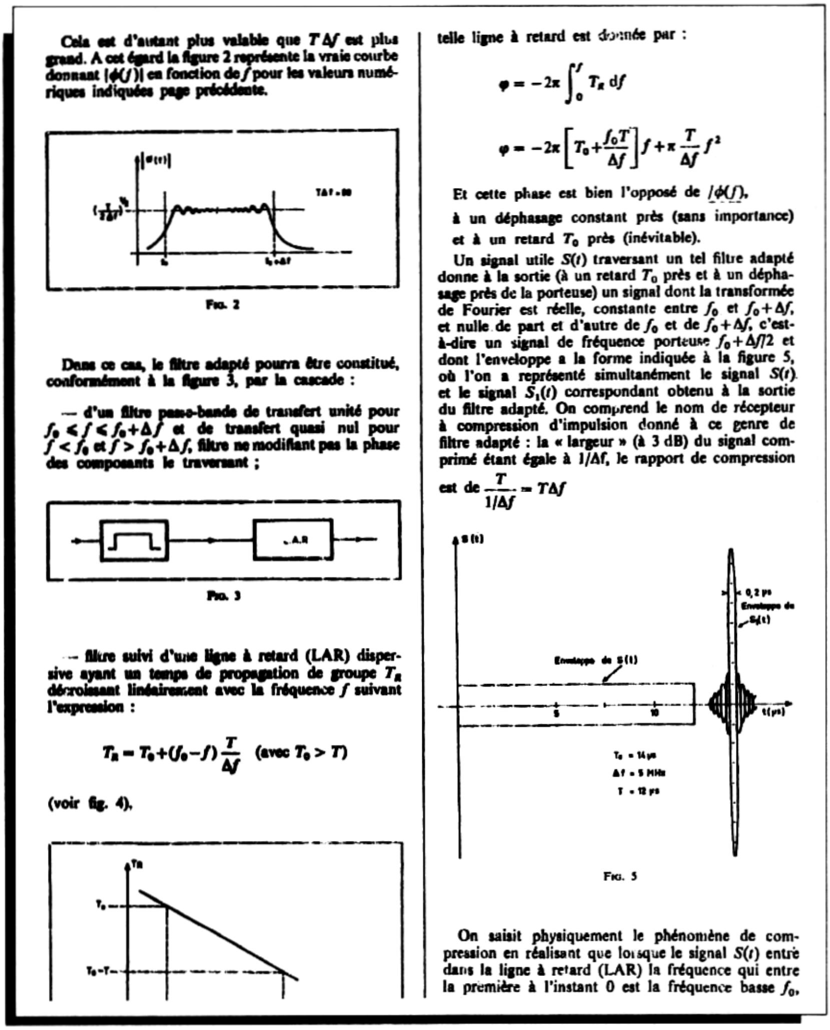
II - L'IMPLANTATION GEOGRAPHIQUE D'UN RESEAU INFORMATIQUE PERFORMANT

L'organisation de l'entreprise française des télécommunications repose sur l'existence de 20 régions. Des calculateurs ont été implantés dans le passé au moins dans toutes les plus importantes. On trouve ainsi des machines Bull Gamma 30 à Lyon et Marseille, des GE 425 à Lille, Bordeaux, Toulouse et Montpellier, un GE 437 à Massy, enfin quelques machines Bull 300 TI à programmes câblés étaient récemment ou sont encore en service dans les régions de Nancy, Nantes, Limoges, Poitiers et Rouen ; ce parc est essentiellement utilisé pour la comptabilité téléphonique.
A l'avenir, si la plupart des fichiers nécessaires aux applications décrites plus haut peuvent être gérés en temps différé, un certain nombre d'entre eux devront nécessairement être accessibles, voire mis à jour en temps réel : parmi ces derniers le fichier commercial des abonnés, le fichier des renseignements, le fichier des circuits, le fichier technique des abonnés contiendront des quantités considérables d'informations.
Le volume total de caractères à gérer en phase finale sur un ordinateur ayant en charge quelques 500 000 abonnés a été estimé à un milliard de caractères au moins. Au moins le tiers des données seront concernées par des traitements en temps réel.
Aucun des calculateurs énumérés plus haut ne permettait d'envisager de tels traitements.
L'intégration progressive de toutes les applications suppose la création d'un support commun pour toutes les informations, une véritable "Banque de données", répartie sur des moyens de traitement nationaux et régionaux, et qui devra rester alimentée, mise à jour en permanence, à partir de la base de l'entreprise, c'est-à-dire les chantiers, les magasins, les guichets des services d'abonnement, les services de personnel etc.
L'étude des différents fichiers à constituer a donc permis de définir les principales caractéristiques du réseau d'ordinateurs nouveaux à mettre en place pour aborder la réalisation du système informatif. L'obligation de faire appel à des ordinateurs de troisième génération, très puissants et dotés de volumineuses mémoires de masse, a conduit à en réduire substantiellement le nombre.
L'implantation de sept centres de calcul interrégionaux constituera un compromis entre : d'une part le désir de réduire le coût économique de l'ensemble, de faciliter la coordination des équipes d'informaticiens; et d'autre part le refus de créer des centres trop importants difficiles à gérer et à diriger,et posant des problèmes délicats de sécurité. Le regroupement des traitements relatifs à plusieurs régions sur chacun de ces sept centres permettra de leur donner une taille relativement homogène. Chaque centre "gèrera" environ un mil-

(c)

Fig. 16(c)—Original CCITT document four (first 2000 lines).

and a former nonofficial version often used for facsimile compression comparisons. Table III gives the coding lengths for the CCITT documents for the official and nonofficial set of CCITT documents, respectively. They include the code length for the different codes necessary for the pattern matching coding. Table IV gives the compression ratio for the same CCITT documents and compares them with the



Cela est d'autant plus valable que $T\Delta f$ est plus grand. A cet égard la figure 2 représente la vraie courbe donnant $|\phi(f)|$ en fonction de f pour les valeurs numériques indiquées page précédente.

FIG. 2

Dans ce cas, le filtre adapté pourra être constitué, conformément à la figure 3, par la cascade :

— d'un filtre passe-bande de transfert unité pour $f_0 < f < f_0+\Delta f$ et de transfert quasi nul pour $f < f_0$ et $f > f_0+\Delta f$, filtre ne modifiant pas la phase des composants le traversant ;

FIG. 3

— filtre suivi d'une ligne à retard (LAR) dispersive ayant un temps de propagation de groupe T_R décroissant linéairement avec la fréquence f suivant l'expression :

$$T_R = T_0+(f_0-f)\frac{T}{\Delta f} \quad (\text{avec } T_0 > T)$$

(voir fig. 4).

telle ligne à retard est donnée par :

$$\varphi = -2\pi \int_0^f T_R \, df$$

$$\varphi = -2\pi \left[T_0+\frac{f_0 T}{\Delta f}\right] f + \pi \frac{T}{\Delta f} f^2$$

Et cette phase est bien l'opposé de $/\phi(f)$,

à un déphasage constant près (sans importance) et à un retard T_0 près (inévitable).

Un signal utile $S(t)$ traversant un tel filtre adapté donne à la sortie (à un retard T_0 près et à un déphasage près de la porteuse) un signal dont la transformée de Fourier est réelle, constante entre f_0 et $f_0+\Delta f$, et nulle de part et d'autre de f_0 et de $f_0+\Delta f$, c'est-à-dire un signal de fréquence porteuse $f_0+\Delta f/2$ et dont l'enveloppe a la forme indiquée à la figure 5, où l'on a représenté simultanément le signal $S(t)$ et le signal $S_1(t)$ correspondant obtenu à la sortie du filtre adapté. On comprend le nom de récepteur à compression d'impulsion donné à ce genre de filtre adapté : la « largeur » (à 3 dB) du signal comprimé étant égale à $1/\Delta f$, le rapport de compression est de $\frac{T}{1/\Delta f} = T\Delta f$

FIG. 5

On saisit physiquement le phénomène de compression en réalisant que lorsque le signal $S(t)$ entre dans la ligne à retard (LAR) la fréquence qui entre la première à l'instant 0 est la fréquence basse f_0,

(d)

Fig. 16(d)—Original CCITT document five (first 2000 lines).

symbol-matching technique of Pratt et al.[2] and the two-dimensional CCITT code. The results are without any synchronization or stuffing bits, which is natural since pattern matching coding would be intended for future facsimile networks such as group four facsimile machines with fewer overhead bits. Therefore, the compressions of the two-dimensional CCITT code and symbol matching have been corrected by deleting the synchronization and stuffing bits and are different from their values given in Refs. 2 and 5.

THE SLEREXE COMPANY LIMITED

SAPORS LANE - BOOLE - DORSET - BH 25 8 ER

TELEPHONE BOOLE (945 13) 51617 - TELEX 123456

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.

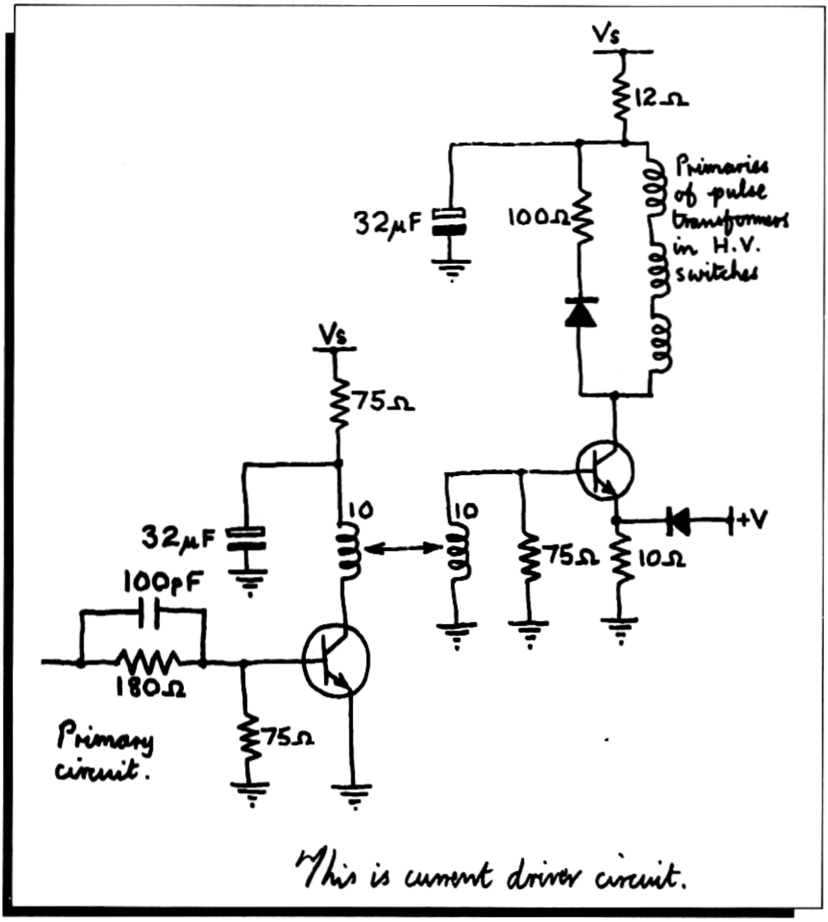
P.J. CROSS
Group Leader - Facsimile Research

(a)

Fig. 17(a)—Document on (first 2000 lines) after pattern matching (first 2000 lines).

Very high compressions are obtained—up to 80. The compression has often doubled compared to that of the two-dimensional CCITT code and is sometimes 4.8 times higher. The compression is, depending upon the documents, 20 to 80 percent higher than the compression derived from the symbol matching technique by Pratt et al.[2] More detailed comparisons and observations are useful when considering the performances of facsimile coding by pattern matching:

1. An astonishing fact is the difference in compression observed



(b)

Fig. 17(b)—Document two (first 2000 lines) after pattern matching.

between the old and the official version of the CCITT documents. For documents three and five the compressions are nearly twice as high for the old version than for the official version. Significant discrepancies are also observed for documents one and eight. This is in spite of the fact that old and official documents are the same except that they were scanned differently. It can also be noted that for the two-dimensional CCITT code, the difference in compression is smaller than five percent except for document eight, where the difference is about 20 percent. It must therefore be concluded that the performances of the pattern matching coding techniques are much more dependent upon the scanning and binary thresholding. Observing both versions of documents three and five, the main difference is that in the official

- 34 -

L'ordre de lancement et de réalisation des applications fait l'objet de décisions au plus haut niveau de la Direction Générale des Télécommunications. Il n'est certes pas question de construire ce système intégré "en bloc" mais bien au contraire de procéder par étapes, par paliers successifs. Certaines applications, dont la rentabilité ne pourra être assurée, ne seront pas entreprises. Actuellement, sur trente applications qui ont pu être globalement définies, six en sont au stade de l'exploitation, six autres se sont vu donner la priorité pour leur réalisation.
Chaque application est confiée à un "chef de projet", responsable successivement de sa conception, de son analyse-programmation et de sa mise en oeuvre dans une région-pilote. La généralisation ultérieure de l'application réalisée dans cette région-pilote dépend des résultats obtenus et fait l'objet d'une décision de la Direction Générale. Néanmoins, le chef de projet doit dès le départ considérer que son activité a une vocation nationale donc refuser tout particularisme régional. Il est aidé d'une équipe d'analystes-programmeurs et entouré d'un "groupe de conception" chargé de rédiger le document de "définition des objectifs globaux" puis le "cahier des charges" de l'application, qui sont adressés pour avis à tous les services utilisateurs potentiels et aux chefs de projet des autres applications. Le groupe de conception comprend 6 à 10 personnes représentant les services les plus divers concernés par le projet, et comporte obligatoirement un bon analyste attaché à l'application.

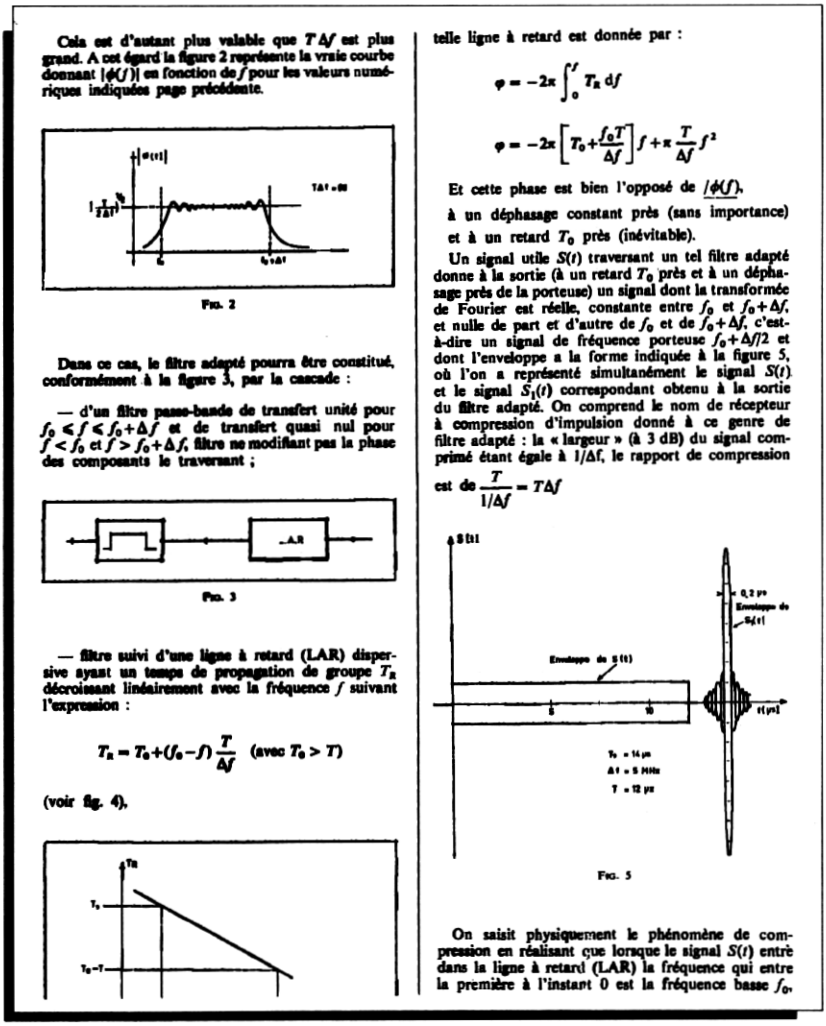
II - L'IMPLANTATION GEOGRAPHIQUE D'UN RESEAU INFORMATIQUE PERFORMANT

L'organisation de l'entreprise française des télécommunications repose sur l'existence de 20 régions. Des calculateurs ont été implantés dans le passé au moins dans toutes les plus importantes. On trouve ainsi des machines Bull Gamma 30 à Lyon et Marseille, des GE 425 à Lille, Bordeaux, Toulouse et Montpellier, un GE 437 à Massy, enfin quelques machines Bull 300 TI à programmes câblés étaient récemment ou sont encore en service dans les régions de Nancy, Nantes, Limoges, Poitiers et Rouen ; ce parc est essentiellement utilisé pour la comptabilité téléphonique.
A l'avenir, si la plupart des fichiers nécessaires aux applications décrites plus haut peuvent être gérés en temps différé, un certain nombre d'entre eux devront nécessairement être accessibles, voire mis à jour en temps réel : parmi ces derniers le fichier commercial des abonnés, le fichier des renseignements, le fichier des circuits, le fichier technique des abonnés contiendront des quantités considérables d'informations.
Le volume total de caractères à gérer en phase finale sur un ordinateur ayant en charge quelques 500 000 abonnés a été estimé à un milliard de caractères au moins. Au moins le tiers des données seront concernées par des traitements en temps réel.
Aucun des calculateurs énumérés plus haut ne permettait d'envisager de tels traitements.
L'intégration progressive de toutes les applications suppose la création d'un support commun pour toutes les informations, une véritable "Banque de données", répartie sur des moyens de traitement nationaux et régionaux, et qui devra rester alimentée, mise à jour en permanence, à partir de la base de l'entreprise, c'est-à-dire les chantiers, les magasins, les guichets des services d'abonnement, les services de personnel etc.
L'étude des différents fichiers à constituer a donc permis de définir les principales caractéristiques du réseau d'ordinateurs nouveaux à mettre en place pour aborder la réalisation du système informatif. L'obligation de faire appel à des ordinateurs de troisième génération, très puissants et dotés de volumineuses mémoires de masse, a conduit à en réduire substantiellement le nombre.
L'implantation de sept centres de calcul interrégionaux constituera un compromis entre : d'une part le désir de réduire le coût économique de l'ensemble, de faciliter la coordination des équipes d'informaticiens; et d'autre part le refus de créer des centres trop importants difficiles à gérer et à diriger, et posant des problèmes délicats de sécurité. Le regroupement des traitements relatifs à plusieurs régions sur chacun de ces sept centres permettra de leur donner une taille relativement homogène. Chaque centre "gèrera" environ un mil-

(c)

Fig. 17(c)—Document four (first 2000 lines) after pattern matching.

version, characters are often clustered together, which leads to incorrectly (or rather "nonconveniently") isolated characters (as shown in Fig. 18a), while for the old version, the characters are rarely clustered together. However, sometimes a character in the old version is isolated into several patterns because not all its pels are connected (as shown in Fig. 18b). The old version of documents three and five should have



Cela est d'autant plus valable que $T\Delta f$ est plus grand. A cet égard la figure 2 représente la vraie courbe donnant $|\phi(f)|$ en fonction de f pour les valeurs numériques indiquées page précédente.

FIG. 2

Dans ce cas, le filtre adapté pourra être constitué, conformément à la figure 3, par la cascade :

— d'un filtre passe-bande de transfert unité pour $f_0 \leq f \leq f_0+\Delta f$ et de transfert quasi nul pour $f < f_0$ et $f > f_0+\Delta f$, filtre ne modifiant pas la phase des composants le traversant ;

FIG. 3

— filtre suivi d'une ligne à retard (LAR) dispersive ayant un temps de propagation de groupe T_R décroissant linéairement avec la fréquence f suivant l'expression :

$$T_R = T_0+(f_0-f)\frac{T}{\Delta f} \quad (\text{avec } T_0 > T)$$

(voir fig. 4).

telle ligne à retard est donnée par :

$$\varphi = -2\pi\int_0^f T_R \, df$$

$$\varphi = -2\pi\left[T_0+\frac{f_0 T}{\Delta f}\right] f + \pi\frac{T}{\Delta f} f^2$$

Et cette phase est bien l'opposé de $/\phi(f)$, à un déphasage constant près (sans importance) et à un retard T_0 près (inévitable).

Un signal utile $S(t)$ traversant un tel filtre adapté donne à la sortie (à un retard T_0 près et à un déphasage près de la porteuse) un signal dont la transformée de Fourier est réelle, constante entre f_0 et $f_0+\Delta f$, et nulle de part et d'autre de f_0 et de $f_0+\Delta f$, c'est-à-dire un signal de fréquence porteuse $f_0+\Delta f/2$ et dont l'enveloppe a la forme indiquée à la figure 5, où l'on a représenté simultanément le signal $S(t)$ et le signal $S_1(t)$ correspondant obtenu à la sortie du filtre adapté. On comprend le nom de récepteur à compression d'impulsion donné à ce genre de filtre adapté : la « largeur » (à 3 dB) du signal comprimé étant égale à $1/\Delta f$, le rapport de compression est de $\frac{T}{1/\Delta f} = T\Delta f$

FIG. 5

On saisit physiquement le phénomène de compression en réalisant que lorsque le signal $S(t)$ entre dans la ligne à retard (LAR) la fréquence qui entre la première à l'instant 0 est la fréquence basse f_0,

(d)

Fig. 17(d)—Document five (first 2000 lines) after pattern matching.

more patterns than the official but fewer library patterns. In fact, old CCITT document three has 2199 patterns and 225 library patterns, while the official version has 1945 patterns and 551 library patterns. The coding length of a library pattern is much greater (by a factor of about 10) than that of a nonlibrary pattern, which explains the difference in compression ratios. It can be concluded that pattern matching is much more dependent on the scanning quality and the thresholding than two-dimensional facsimile codes.

Table III—Coding length in bits with pattern matching coding

(a) CCITT documents (2376 lines, 1728 pels/line)

	CCITT1	CCITT2	CCITT3	CCITT4	CCITT5	CCITT6	CCITT7	CCITT8
Mode bit	2376	2376	2376	2376	2376	2376	2376	2376
Horizontal position	11847	8096	21395	46464	23551	15356	36828	31240
No more pattern	888	1587	2193	1854	2235	1929	3165	3231
Vertical move of pattern	1114	411	1921	5724	2242	1347	3427	3642
Library identification code	4895	3749	9458	18584	10848	6426	19289	13085
Library pattern description	43983	71427	120749	41138	94923	71959	204113	115415
Total	65103	87646	158092	116140	136175	99393	269198	168989

(b) Older CCITT documents (same as documents used in Ref. 1) (2128 lines, 1728 pels/line)

	CCITT1	CCITT2	CCITT3	CCITT4	CCITT5	CCITT6	CCITT7	CCITT8
Mode bit	2128	2128	2128	2128	2128	2128	2128	2128
Horizontal position	12342	7733	24189	49610	27511	16049	38566	30635
No more pattern	846	1533	2103	2574	2091	2007	3942	3168
Vertical move of pattern	1155	463	2448	5726	3028	1411	3812	2441
Library identification code	4958	3599	11078	21385	11940	6927	21842	12773
Library pattern description	23254	62825	29368	26700	22441	55077	155681	122986
Total	44683	78281	71314	108123	69139	83599	225971	174131

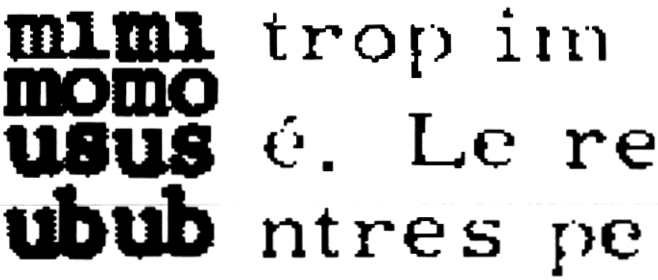
Table IV—Comparison of compression ratios

(a) Official CCITT documents (2376 × 1728 pels)

Picture	Pattern Matching	Two-Dimensional CCITT Code	Increase Versus Two-Dimensional CCITT Code
CCITT1	63.1	28.3	122%
CCITT2	46.8	47.5	−1%
CCITT3	26.0	17.9	46%
CCITT4	35.4	7.4	378%
CCITT5	30.2	15.9	90%
CCITT6	41.3	30.8	34%
CCITT7	15.3	7.4	106%
CCITT8	24.3	26.9	−8%

(b) Nonofficial CCITT document (2128 × 1728 pels)

Picture	Pattern Matching	Symbol Matching	Two-Dimensional CCITT Code	Increase Versus Symbol Matching	Increase Versus Two-Dimensional CCITT Code
CCITT1	82.3	63.1	28.6	30%	188%
CCITT2	47.0	38.1	45.4	23%	3%
CCITT3	51.6	32.4	18.5	59%	179%
CCITT4	34.0	25.6	7.5	33%	353%
CCITT5	53.2	33.6	16.5	58%	222%
CCITT6	44.0	29.5	30.2	49%	46%
CCITT7	16.3	9.0	7.1	81%	130%
CCITT8	21.1	17.8	22.1	19%	−5%



(a) (b)

Fig. 18—Characters isolated in an unwanted way. (a) Characters clustered together. (b) Text containing characters isolated into several symbols.

2. The increase in compression ratio compared to the two-dimensional run-length code is quite variable. For documents containing mostly handwritten drawings and text, such as documents two and eight, there is sometimes a slight decrease in the compression ratio. That is because there are few matching patterns. For example, for document two, there are 736 patterns, but 448 of them are library patterns. For documents containing mostly text, such as document four, the compression ratio increases by a factor of about 4.5. For documents containing a mixture of text and drawings, the increase varies between 35 and 220 percent, depending on the content and

thresholding. It should be noted that for document seven, which contains printed ideograms, the increase in compression ratio is smaller than for regular printed text because there are more ideograms than letters, but the compression ratio still doubled.

3. The increase in compression ratio by pattern matching is 20 to 80 percent compared to the symbol matching of Ref. 2. The increase has been obtained by a combination of several factors. The most important are (a) isolation of nonsymbols (lead to significant improvement for documents three, four, five, and six, but has a slight negative effect on documents two and eight), (b) better matching, leading to fewer library patterns, and (c) improved coding efficiency obtained by sorting the patterns and by other coding modifications.

By looking at the coding length necessary for the different kinds of code words, in Table III, it is clear that the predominant part of the code is used for the description of the library patterns, accounting generally for more than 60 percent of the total coding length. Therefore, improving it can bring the highest reward. The improvement can be obtained by reducing the number of library patterns or by coding the pattern description more efficiently. The next most bit-consuming part is the coding of the horizontal position; it uses about 20 percent of the total coding length.

6.3 Complexity

The pattern-matching coding has the disadvantage of being complex and time-consuming—the price to pay for an efficient coding. The most time-consuming parts are: the isolation, the template matching, and the matching decision. The isolation is both complex and time-consuming, and therefore the most difficult part, but by using fast logic, it is possible to isolate all the patterns in about one second. The template matching is a simple operation, but it takes a long time. It is therefore less of a challenge, since it is easily done in parallel and with simple hardware. The most time-consuming part of the matching decision uses local operators on, for example, 3×3 windows and can therefore also be realized without much complication. Most of the high-level operations are much slower and can be done by microprocessors. This system should not be more complicated than in Ref. 2.

An important factor is that the decoding is much easier and faster than the coding, since there is no isolation or matching. Such a technique is therefore particularly suited for transmission with one sender and several receivers.

An experimental pattern matcher has been built to show that the same kind of compression can be obtained when scanning real documents. By using a mixture of custom logic and programmed logic, transmission has reached speeds at rates up to 64 kb/s. A document is then usually sent in one to two seconds.

VII. EXTENSIONS

Several improvements or different applications of a pattern matcher can be considered. Some of them will be described here.

7.1 Multipage document and prestored libraries

When transmitting several pages of a document, the library from one page can be used for next page, thus reducing the number of library patterns for each page. In such cases, compressions up to eight times higher than with conventional coding techniques can be achieved. If a few fonts are prestored in the coder and decoder, the compression can be increased significantly.

7.2 Very high-quality transmission

It is possible to use a tighter matching algorithm when even slight distortions are not tolerated. Such a mode can easily be implemented. It reduces the compression by an average of 15 percent. In that case, most of the postprocessing can be deleted.

7.3 Standardization

The CCITT is looking into standardizing facsimile coding techniques for future facsimile machines communicating over digital links (Group four facsimile apparatus). The modified READ code (also called two-dimensional CCITT code) has been standardized. The pattern matching coding technique has been proposed by AT&T to the CCITT as an optional coding technique yielding much higher compression. The only difference in the proposal compared to this paper is that no cross decomposition is applied. The compression is therefore slightly lower.

7.4 High-resolution graphics

Future scanners and coders will probably include resolutions higher than 200 pels/in. They will probably use 300 and 400 pels/in. The pattern matching technique can easily be modified for such resolution. The maximum size of the patterns should be increased to keep the coding efficient. In addition, the codes for the positioning of patterns must be slightly changed. The matching algorithm would stay unchanged. Compared to conventional techniques, the improvement in the compression will be as high and often even higher at such resolutions.

VIII. CONCLUSION

A system for coding of facsimiles using pattern matching has been described. It allows an important increase in the compression ratio compared with a symbol matching system[2] and gives a compression ratio that is up to 4.8 times that of conventional facsimile coding techniques. The improvement is naturally greater for printed text than for handwritten text. It is felt that further significant inprovements are possible by better matching and coding. An important observation is that pattern matching coding is very dependent on the digitization and thresholding. Therefore, the combination of the thresholding and the isolation could lead to significant improvements in compressions. Another consequence is that if a bad quality scanner is used, the pattern matching will hardly lead to higher compressions than conventional facsimile codes. With modern electronics components, a pattern matcher can be realized by hardware and would lead to an important reduction in the transmission costs of high-volume facsimiles.

IX. ACKNOWLEDGMENT

The authors would like to thank A. N. Netravali for the numerous and fruitful discussions, E. F. Brown and E. S. Joseph for their contribution to the experimental unit, and S. M. Rubin for providing useful software.

REFERENCES

1. R. B. Arps, "Binary image compression," in *Image Transmission Techniques*, W. K. Pratt, ed., New York:Academic Press, 1979.
2. W. K. Pratt, et al., "Combined Symbol Matching Data Compression System," Proc. IEEE, *68*, No. 7 (July 1980), pp. 786–96.
3. R. N. Ascher and G. Nagy, "A means for achieving a high degree of compaction on scan-digitized text," IEEE Trans. Comput., *C-23* (November 1974) pp. 1174–9.
4. S. A. Dudani, "Region extraction using boundary following," in *Pattern Recognition and Artifical Intelligence*, C. H. Chen, ed., New York: Academic Press, pp. 216–32.
5. R. Hunger and A. H. Robinson, "International Digital Facsimile Coding Standards," Proc. IEEE, *68*, No. 7 (July 1980), pp. 854–67.

AUTHORS

Glenn L. Cash, B.S.E.E., 1981, Monmouth College; Bell Laboratories, 1981—. Mr. Cash has worked on the design and implementation of a real-time pattern matching facsimile coder. His research interests include image processing, artificial intelligence, and computer system design. Member, Eta Kappa Nu, Sigma Pi Sigma.

Ottar Johnsen, E.E., 1974, Ph.D., 1979, Ecole Polytechnique Federale de Lausanne, Switzerland; Ecole Polytechnique Federale de Lausanne, 1974–1979; Bell Laboratories, 1979—. Mr. Johnsen was assistant at the Signal Processing Laboratory of the Ecole Polytechnique Federale de Lausanne, working in the field of picture coding. At Bell Laboratories, as a member of the Visual Communications Research Department, he is involved in research in picture processing and coding, and in communication systems.

Jakub Segen, M.S.E.E., 1971, Technical University of Wroclaw, Poland; Ph.D., 1980, Carnegie-Mellon University; Bell Laboratories, 1980—. Mr. Segen worked as a computer engineer at ZETO, Woclaw, and New York University. Recently, he has been involved with signal and image analysis, and his main research interests are machine perception and learning. He is a member of Robotics Systems Research Department, working on aspects of computer vision.