SYMBOL MATCHING BASED FACSIMILE DATA COMPRESSION*

Wen-hsiung Chen, John L. Douglas, and Robert D. Widergren
Compression Labs, Inc.
Campbell, CA.

ABSTRACT

Presented here is an efficient facsimile data coding scheme (CSM) which combines an extended run-length coding technique with a symbol recognition technique. The CSM scheme first partitions the data into run-length regions and symbol regions. The run-length regions are then coded by a modified Interline Coding technique, while the data within the symbol region is further subpartitioned into regions defined as symbols. A prototype symbol library is maintained, and as each new symbol is encountered, it is compared with each element of the library. These comparisons produce a signature for the new symbol. A tolerance threshold is used to evaluate the "goodness" of the comparison. If the tolerance threshold indicates a matching symbol, then only the location and the library address need be transmitted. otherwise the new symbol is both transmitted and placed in the library. For finite sized libraries a scoring system determines which elements of the library are to be replaced by new prototypes. Simulation results are demonstrated for both CCITT and Xerox standard documents. In all cases the CSM compression ratio exceeds the best published run-length technique by a factor of at least two, for pages that are predominately test. For non-test predominate pages, the CSM scheme is at least as good as the best published run-length technique.

INTRODUCTION

During the past twenty years there has been considerable effort toward the development of efficient coding techniques for the transmission of binary valued facsimile imagery.[1,2] One large class of systems is based upon the run-length coding concept in which the length of black and white segments are coded along each scan line. This concept has been extended to the second dimension where the correlation between scan lines is exploited.[3-7] A second class of systems utilizes the skipping white concept based on the fact that the amount of black is usually a very small fraction of the image.[8] The principal advantage of these two classes is their adaptability to a wide variety of data sources such as alphanumeric characters, handwritten script, line drawings, etc. The limitation of these systems,

---

* This paper has been invited specially by the session Chairman. The essence of this paper was also presented at the SPIE symposium in August 1978.

however, is that only a limited amount of compression can be achieved. There is a third class of systems which involves character recognition techniques. Significantly larger compression can be achieved by this system. However, the application of such a system is very limited since the class of images it can deal with is restricted to typewritten and/or printed matter with a small number of fonts and formats which the transmitter and the receiver agree upon beforehand. A block character coding system which incorporates the advantages of run-length and character recognition has been proposed.[10] However this system has never been tested by simulation or other means. The drawback of this system is that the compression ratio is expected to be less than that of a run-length coder for those images where little or no matching can be conducted.

Combined Symbol Matching (CSM) utilizes the best features of an extended coding technique and a symbol recognition technique. The scheme switches between run-length mode and symbol matching mode by partitioning the data into run-length regions and symbol regions. The run-length regions are coded by a modified interline coding technique, while the symbol within the symbol regions is blocked and processed by a symbol matching technique. Simulation results of 96 lpi and 200 lpi images indicated that the compression ratios exceed the best published run length technique by a factor of at least two for predominately text pages. For non-text predominate pages, the CSM scheme is at least as good as the best published run-length coding technique. The CSM scheme has been implemented using Compression Labs, Inc. developed hardware for input images of 96 lpi resolution. In all instances the compression results agree with that of the computer simulation.

COMBINED SYMBOL MATCHING SYSTEM

The combined symbol matching system is a dual mode encoding system which features the advantages of extended run-length encoding and symbol recognition systems. Figure 1 illustrates the block diagram of such an encoding system. In operation, n scan lines of binary image data are stored in a rotary buffer (physically 1/2" of the scanned pages). The data then feeds into the text locater to determine if a text line can be defined based upon the structure of data. If the text locater determines that no test line can be defined, then a fixed number of lines of data in the buffer are coded by an extended run-length coding scheme—a modified Interline Coder. On the other hand if the text locater determine that a text line can be defined, a block symbol coding scheme is employed to block the symbols. A rectangular block is sequentially placed around each symbol of the text line. The first blocked symbol is automatically put into the prototype library, and as each new blocked symbol is encountered, it is compared with each element of the library. If the comparison indicates similarity then the library identification (ID) code along with the location coordinates are transmitted. If the comparison does not indicate a similarity, then the new symbol is both transmitted and placed in the library. A scoring system is employed to tract

the usefulness of each library element. This process continues until a specific library size is achieved. Thereafter, as each unmatched new prototype is placed in the library, the least useful library element is eliminated. At the receiver, the same size library and the same scoring system are utilized to maintain synchronization with the transmitter.

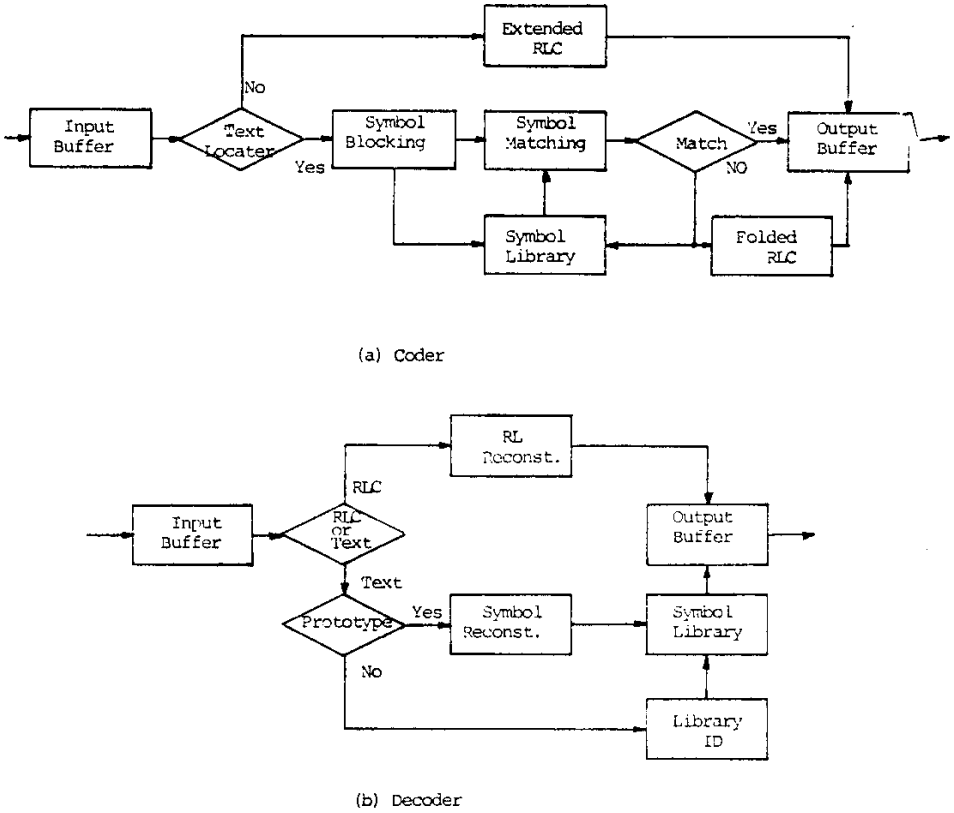


Figure 1. Combined Symbol Matching Coding/Decoding System

a) Determination of Text Line

In the implemented CSM system, the stored n scan-lines of binary data first undergo the text line determination process. The stored data is divided into k equivalent segments, and the projection of black pixels along the scan line of each segment is computed. Let h_1 (m), h_2 (m) --- h_k (m) be the k projections, where m is the line number in the rotary buffer. Then it can easily be seen that if the ith segment contains text

$$h_i(m_L) = o$$

$$\text{for } T_1 \leq m_H - m_L \leq T_2 \qquad (1)$$

$$h_i(m_H) = 0$$

$$h_i(m_j) = o \quad \text{for } j = L + 1, L + 2, \ldots H - 1$$

where T_1 = minimum number of scan lines that will span one text symbol

T_2 = maximum number of scan lines that one symbol will span

and non-text otherwise. Suppose that k', $k' \leq k$, is the number text segments determined from the above process; then a text line can be defined if

$$ck' - (1-c)(k-k') \geq o \qquad 2(a)$$

and non-text line if

$$ck' - (1-c)(k-k') < 0 \qquad 2(b)$$

where c is a constant ($o < c < 1$)

The purpose of dividing the stored data into k segments is to determine if symbol matching can be used for the regions where characters and noncharacters are mixed together (Figure 2). It also serves the purpose of determining the text line for those pages with a minor degree of misplacement. Figure 3 illustrates the skewing effect caused by a minor misplacement of a page. A test line is determined even if some of the segments are voted as non-text segments. Of course a few overhead bits are needed to to code the location of the top of the text line once the line is defined.

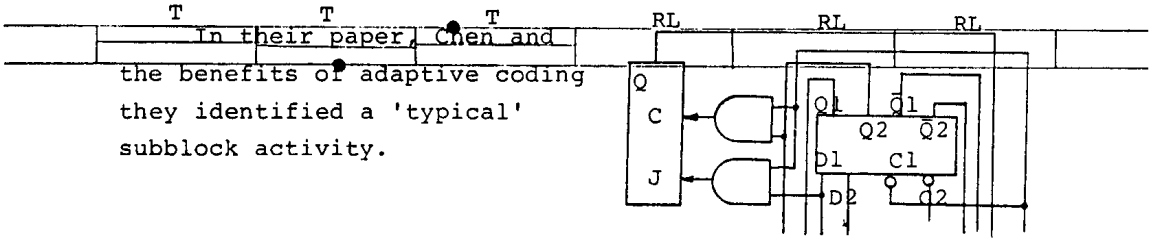


Figure 2. Determination of Text Line with Mixture of Text and Non-Text Regions

b) Symbol Blocking

The block technique used here emphasizes sequential blocking as opposed to the random blocking demonstrated in reference 10. This not only minimizes the code bits for horizontal addresses but also provides fast enough timing for real time implementation. The basic blocking process is a line casting process as illustrated in Figure 4. The data within the top and the bottom of the next line is scanned vertically from left to right until a black pixel is detected. This initially detected pixel defines the left boundary of the blocked symbol. The scanning process continues to the right until a white vertical line is

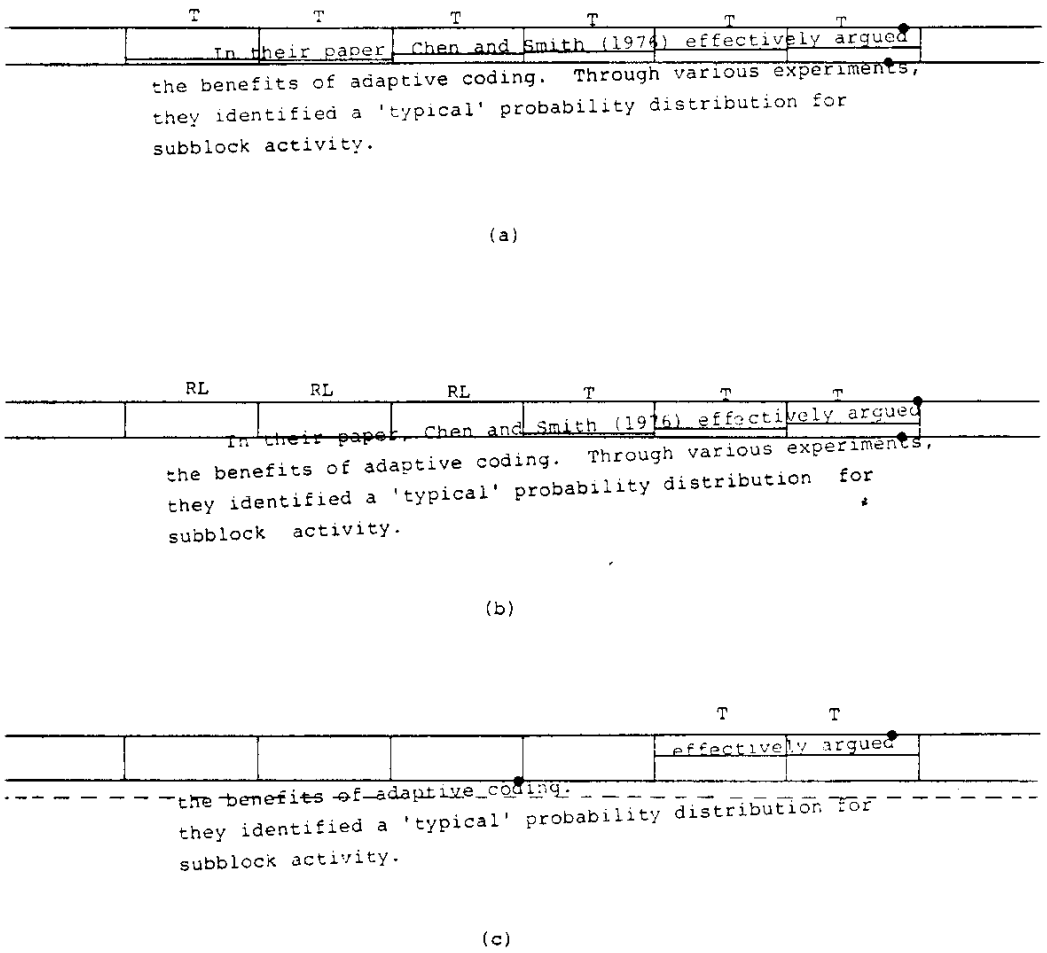


Figure 3. Determination of Text Line with Skew Effect

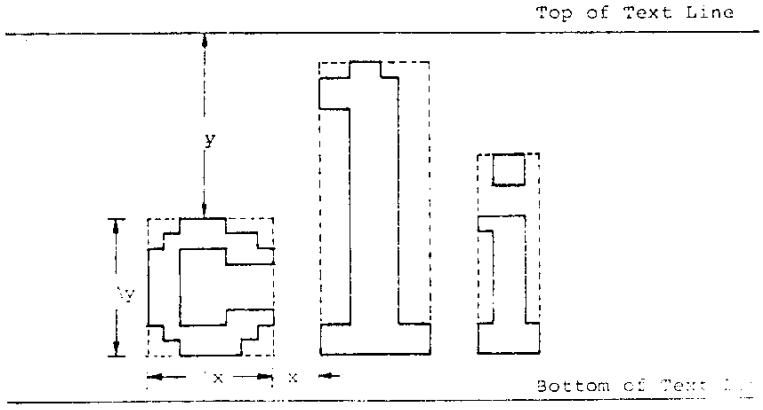


Figure 4. Symbol Blocking Example

detected which defines the right boundary of the blocked symbol. Next the data contained in the left and the right boundaries of the test line is scanned horizontally from top to bottom until a black pixel is detected. This defines the top boundary of the blocked symbol. The process continues from bottom up until a black pixel is detected which defines the bottom boundary of the blocked character.

The address of the blocked character can now be described horizontally by the pixel differences, x, between two adjacent blocks and vertically by the pixel differences, y, between the top of the text line and the blocked character. This address is coded using a Huffman code table. Next the blocked character can be described by its width (Δx) and height (Δy). These properties are later used in the symbol matching process and are coded only in the instance of a new prototype.

c) Symbol Coding

After a symbol has been blocked, the decision function is applied to each prototype element of the library. If a match is indicated only the matching library ID is coded. Otherwise the blocked symbol is coded with a folded run length coding scheme along with the symbol height (Δy) in addition to being placed in the library as a new prototype element. The symbol width (Δx) is not necessary since an end of data code is used in the run lengthing of the prototype. The use of the end of data code not only saves a portion of the bits over the symbol width code but also eliminates the last run length of the prototype. Figure 5 shows a typical example of bit allocations. Huffman codes are used for the run length coding as well as the Δy. The library ID is coded numerically with $\log_2 L$ bits where L is the number of elements in the library. While the usage of library elements is not a uniform process, the process is of such a structure and is dynamic to the point that employing Huffman codes provides little savings.

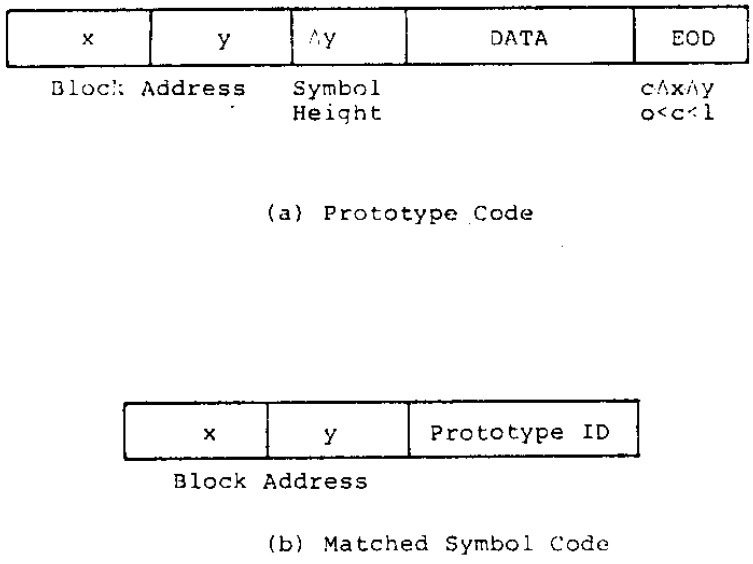


Figure 5. Transmission of Blocked Symbols

The run length coder used to code the data on the prototype blocks is simply a one dimensional folded run length coder with runs running along the vertical direction. The folded run length coder generally provides a small percentage improvement over the non-folded coder, while the run length along the vertical direction provides another degree of improvement over the horizontal direction. It is worthwhile to point out that little improvement can be achieved by using any extended run length coding techniques (e.g. RAC EDIC, ILC, etc.) to code the prototype blocks. This is due to the fact that more code bits are necessary to code the short interlines than are required with straight run length code even though there are correlations among interlines.

d) Prototype Library and Scoring System

The number of prototypes needed in the library is dependent upon the contents of the images. For a very dense page, such as the CCITT #4 document, a library of 256 elements or more is generally needed. For a typical business letter such as the CCITT #1, a library size of 128 is more than sufficient. The up grade of library size to a larger size generally improves the compression factor slightly since no library elements need be reinstated once they are bumped out by the scoring system. The reinstatement of library elements requires additional prototypes to be transmitted which generally takes more bits than those additional bits needed to code the up grade library ID. However, the expensive memory required for real time implementation practically limits one from using a large size library.

The scoring system provides a mean of bumping out the least used prototypes once the library is full. Many factors (e.g., size of blocked symbols, number of different fonts, size of prototype library etc.) are involved in determining a good scoring system which needs a thorough statistical analysis and mathematical modeling. Substantial effort has been devoted to derive a good scoring system. However, no significant result has been accomplished. In the current CSM system a simple minded scoring system is used, whereby the oldest, least used prototypes are eliminated. Every prototype element is pre-assigned an initial score. As a new blocked sysmbol is encountered the score of those non-matched elements is decreased by one and the matched element is increased 100. If no match occurs the library element with the lowest score is then replaced by the new prototype.

e) Symbol Matching

The idea of symbol matching is probably the ultimate solution to a high degree of image data compression. Since more than 80% of the FAX traffic today involves typewritten and printed matter, an efficient matching scheme for the blocked symbols is of great importance. It should be understood that symbol matching has a different goal than symbol recognition. The goal of a symbol matching scheme is to identify symbols that are alike

enough so that if one is used to replace another no error will result in the received copy. On the other hand, the goal of symbol recognition is to recognize symbols belonging to the same class, no matter how different they may be.

The natural way to achieve this "matching" goal is to perform a point-by-point comparison of two symbols, and to count the number of points which are different. If enough points are similar the symbols are the same for purposes of printed reproduction. If enough points are different, the symbols are different, from the reproduction point of view, even though they may belong to the same class.

Since the scanning and digitization process may result in identical symbols being differently oriented within a block, many unnecessary mismatches would occur if a simple point-by-point comparison were used. CLI has derived an adaptive matcher which takes care of this problem and allows for the orientation variations.

f) Extended Run Length Coding

The run length coding scheme used to code the non-text line data in the rotary buffer is a modified interline code. Several excellent coding schemes such as Predictive Coding[1], Relative Address Coding[4] ordering Techniques[5], Interline Coding[6], and Edge Differencing Coding[7] are available in the literature. Among them Interline Coding (ILC) generally provides the highest compression ratio. In the ILC scheme the odd numbered scan lines are coded by regular run length coding (RLC) where the black and white runs are treated separately due to the difference in statistics of black and white runs. The even numbered scan lines are separated into two regions depending on whether or not the color of the pixels immediately above and below are different. The pixels in the first regions are coded one after another by RLC. The positions of those pixels whose colors are different from pixels in the second regions are also encoded by RLC. The idea of reconstructing an even line by its two immediate odd lines is modified in the Combined Symbol Matcher to improve the coding efficiency. Since correlations among interlines often exist over more than three lines (even with 96 lpi images), the modification is made to estimate a variable number of lines by their two adjacent lines. The total number of lines involved ranges from 1 to n where n is dependent upon image resolution. Generally for the 96 1pi resolution images, the optimum number of n is 4 while for 200 1pi images the number of n is 7.

g) Huffman Code Generation

As described earlier separate Huffman Code tables are used for coding the various elements of the data. The codes are adoptively generated without overhead by the following technique. Both transmitter and receiver begin initially with a nominal set of Huffman Code tables and during the process of using these codes, both the transmitter and

receiver keep new statistical information. At the point in time when a sufficient amount of statistics has been gathered for any one table, both transmitter and receiver generate new codes and reinitialize the statistics for that table.

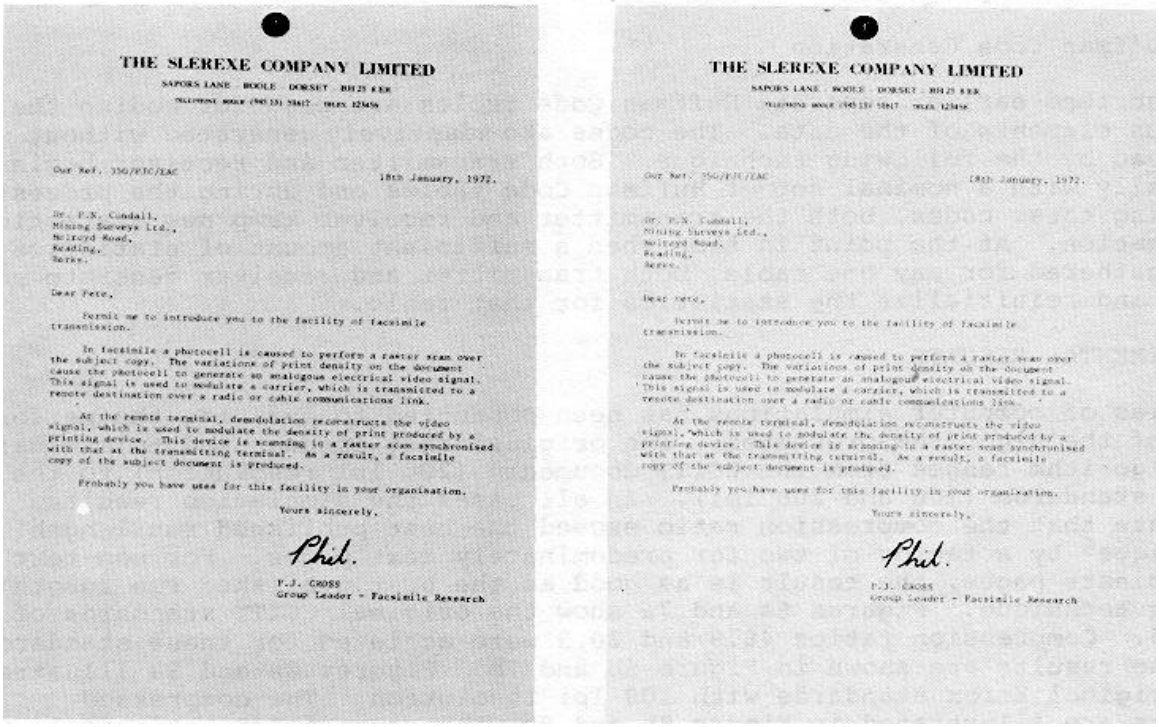
EXPERIMENTAL RESULTS

A series of computer simulations has been conducted to evaluate the performance of the algorithm presented. The original test images used to evaluate the algorithm ranges from the CCITT documents (200 lpi resolution) to the Xerox standards (100 and 200 lpi). In all cases the simulation results indicate that the compression ratio exceed the best published run length technique by a factor of two for predominately text pages. For non-text predominate pages, the result is as good as the best published run length coding technique. Figures 6a and 7a show the original CCITT standards of #1 and #4. Compression ratios 48.9 and 20.3 were achieved for these standards and the results are shown in Figure 6b and 7b. Figures 8a and 9a illustrate the original Xerox standards with 100 lpi resolution. The compressed results are illustrated in Figure 8b and 9b with compression ratio of 14.95 and 4.48 respectively. It is worthwhile to mention that the compression ratios for the Xerox documents could have been much better if the originals were clearer copies. The broken characters on the upper right hand area suggests that the original documents used for digitization must be several generation copies.

CONCLUSION

An efficient facsimile compression algorithm which combines an extended run length coding scheme and symbol recognition scheme has been presented. The algorithm uses a modified interline codes to code the non-test segment and uses a sequencing symbol blocking method to block the test data for matching. The blocking method presented is applicable only if a test line can be defined. This is equivalent to say the orientation of the page has to be in “portrait” mode. Furthermore the algorithm is not optimal since the processor selects either run length or symbol matching on the entire 8½" page direction basis depending on the outcome of the voting strategy. A more efficient algorithm which mixes the run length regions as well as symbol match regions along the 8½" page direction has been derived and test is currently underway at CLI to evaluate the algorithm. This algorithm can handle not only the “portrait” mode and “lanscape” mode as well.

It is worth to note that the algorithm processed is not an entropy preserving scheme. The symbol matching process itself allows for signature threshold difference in declaring a match as there is hardly ever a totally exact signature match even in obviously matchable symbols. This is due to various factors such as scan density and original page quality.



THE SLEREXE COMPANY LIMITED

SAPORS LANE · BOOLE · DORSET · BH25 8ER

Our Ref. 350/PJC/EAC 18th January, 1972.

Mr. 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.

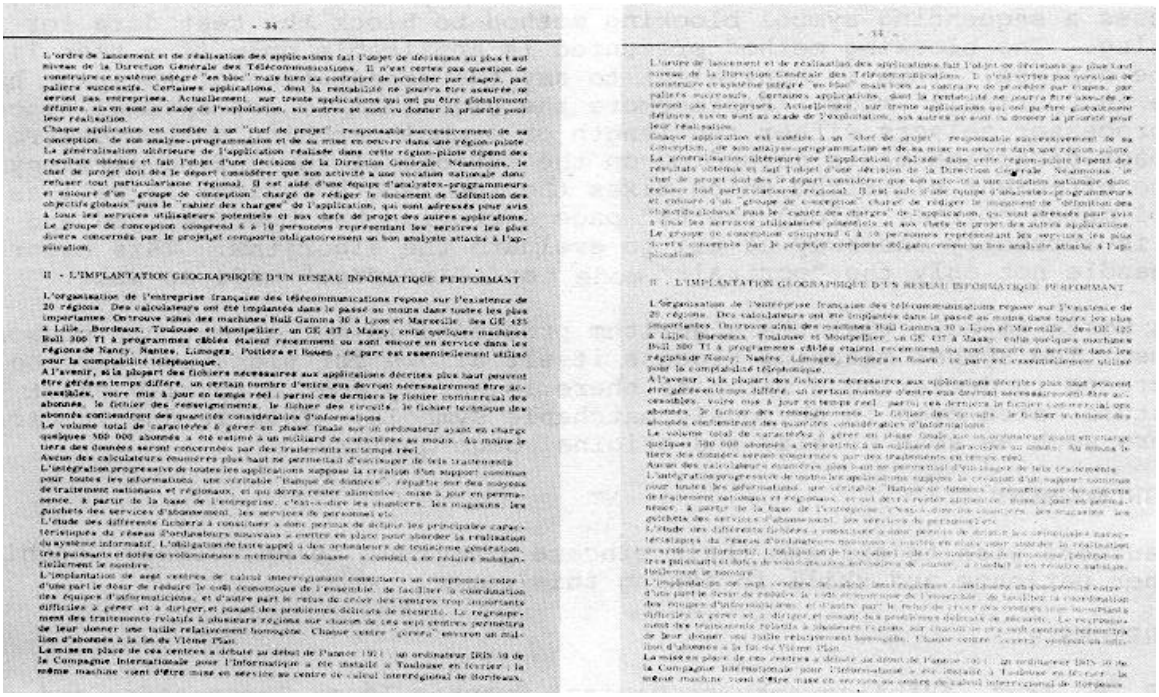
Yours sincerely,

Phil.

P.J. CROSS
Group Leader - Facsimile Research

(a) Original **(b) 48.9:1 Compression**

Figure 6. Combined Symbol Matching Experimental Result (200 ipi)



- 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 dans 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, le 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. A partir de la base de l'entreprise, c'est-à-dire des abonnés, les magasins, les guichets des services d'abonnement, les services de personnel etc.

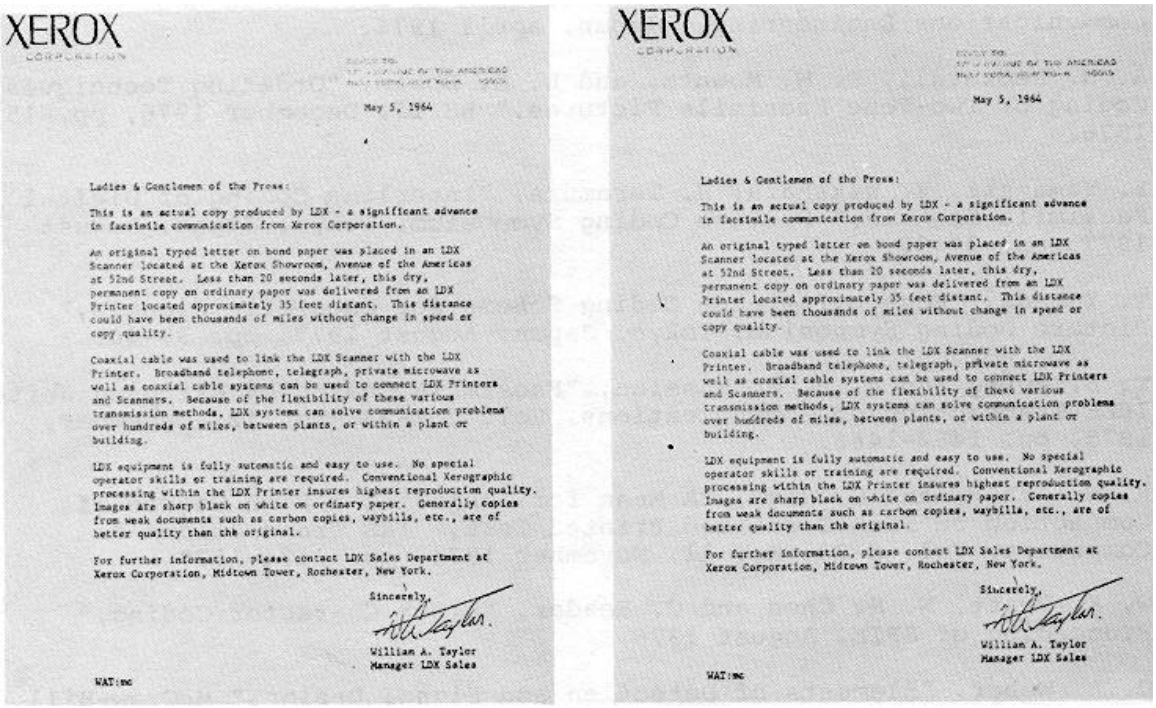
L'étude des différents fichiers à constituer a donc permis de définir les principales caractéristiques du réseau d'ordinateurs nouveaux à mettre en place pour aborder la réalisation du système informatif. L'obligation de faire appel à des ordinateurs de troisième génération, très puissants et dotés de 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 "gérera" environ un million d'abonnés à la fin du VIème Plan.

La mise en place de ces centres a débuté au début de l'année 1971 : un ordinateur IRIS 80 de la Compagnie Internationale pour l'Informatique a été installé à Toulouse en février ; la même machine vient d'être mise en service au centre de calcul interrégional de Bordeaux.

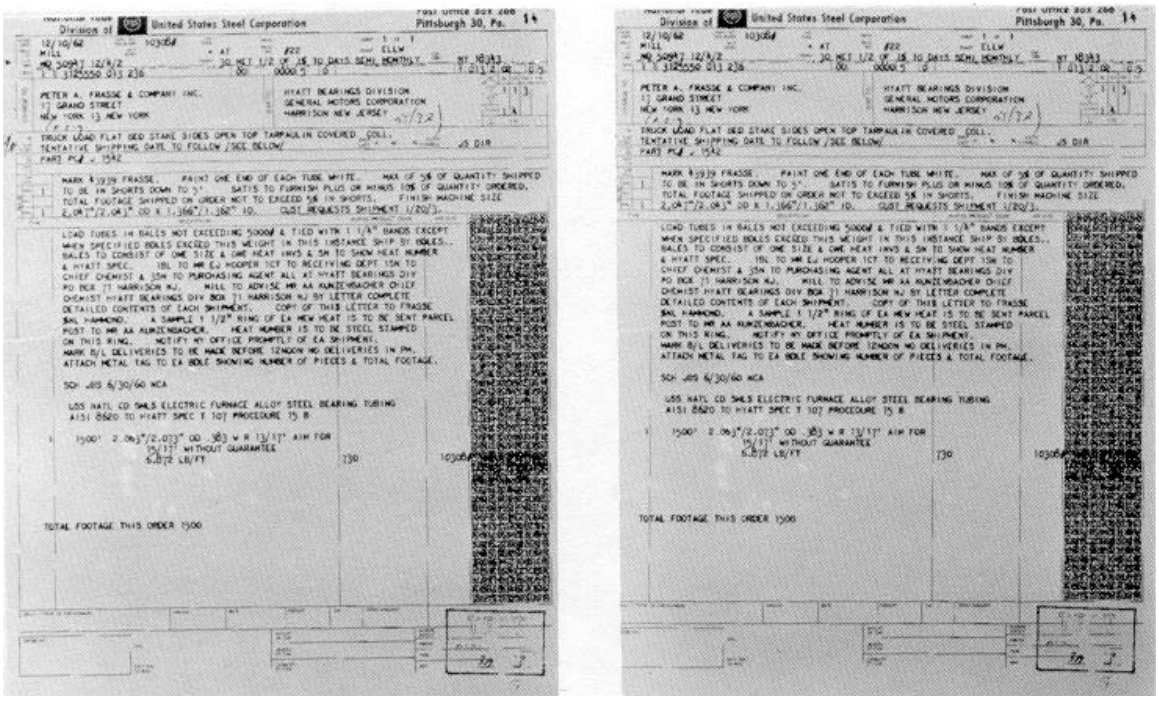
(a) Original **(b) 20.3:1 Compression**

Figure 7. Combined Symbol Marching Experimental Result (200 lpi)



(a) Original **(b) 14.95:1 Compression**

Figure 8. Combined Symbol Matching Experimental Result (100 lpi)



(a) Original **(b) 4.48:1 Compression**

Figure 9. Combined Symbol Matching Experimental Result (100 lpi)

ACKNOWLEDGMENT

The authors wish to express their sincere gratitude to Ms. Vickey Gonthier for her patience and skill in typing this paper.

REFERENCES

1. R. B. Arps, “Bibliography on Digital Graphic Image compression and Quality,” IEEE Transactions on Information Theory, Vol. IT-20, January 1974, pp. 120-122.

2. T. Huang, “Coding of Two-Tone Images,” IEEE Transactions on Communications, Vol. COM-25, No. 11, November 1977, pp. 1406-1424.

3. H. Kobayashi and L. R. Bahl, “Image Data Compression by Prediction Coding,” IBM TJ, March 1974, pp. 164-179.

4. Y. Yamazaki, Y. Wakahera, and H. Teramuda, “High Speed Digital Facsimile Equipment Quick-Fax,” IG 76-4. Institute of Electronic and Communications Engineering. Japan, April 1976.

5. A. N. Netravali, F. W. Mounts, and E. G. Bowen, “Ordering Techniques for Coding of Two-Tone Facsimile Pictures,” BS Ti, December 1976, pp. 1539-1976.

6. Y. Yamazaki, Y. Wakahara, H. Teramura, “Interline Coding of Digital Facsimile Signals,” Picture Coding Symposium, Tokyo, Japan, August 1977, pp. 37-38.

7. T. Yamada, “Edge Difference Coding Scheme for Facsimile Signals,” Picture Coding Symposium, Tokyo, Japan, August 1977, pp. 39-40.

8. T. S. Huang and A. B. S. Hussian, “Facsimile Coding by Skipping White,” IEEE Transaction on Communications, Vol., COM-23, No. 12, December, 1975, pp. 1452-1466.

9. R. N. Asher, and G. Nagy, “A Mean for Achieving a High Degree of Compaction on Scan-Digitized Printed Text,” IEE Transactions on Computers, Vol. C-23, No. 11, November 1974, pp. 1174-1179.

10. W. K. Pratt, W. H. Chen and C. Reader, “Block Character Coding,” Proceeding of SPIE, August 1976.

11. C. L. Weber, “Elements of Detection and Signal Design,” McGraw-Hill, Inc., 1968.