which has solution $\hat{d}_i^2(X_k)=1/N_i$ satisfying (53). By substituting $1/N_i$ and 0 for $\hat{d}_i^2(X_k)$ in (22) it can easily be shown that

$$g(N_i, 0) > g(N_i, 1/N_i) > 0. \quad (55)$$

This proves that $\hat{d}_i^2(X_k)=1/N_i$ is a minimum and that $g(\cdot)>0$ for $\hat{d}_i^2(X_k)$ satisfying (53).

References

[1] M. Hills, "Allocation rules and their error rates," *J. Stat. Royal Soc.*, series B, vol. 28, pp. 1–31, 1966.
[2] W. H. Highleyman, "The design and analysis of pattern recognition experiments," *Bell Syst. Tech. J.*, vol. 41, pp. 723–744, Mar. 1962.
[3] L. Kanal and B. Chandrasekaran, "On dimensionality and sample size in statistical pattern classification," in *Proc. Nat. Electronics Conf.*, vol. 24, 1968, pp. 2–7.
[4] P. A. Lachenbruch and R. M. Mickey, "Estimation of error rates in discriminant analysis," *Technometrics*, vol. 10, no. 1, pp. 1–11, 1968.
[5] T. Marill and D. M. Green, "On the effectiveness of receptors in recognition systems," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 11–17, Jan. 1963.
[6] E. Parzen, "An estimation of a probability density function and mode," *Ann. Inst. Statist. Math.*, vol. 33, pp. 1065–1076, 1962.
[7] R. P. Heydorn, "An upper bound estimate on classification error," *IEEE Trans. Inform. Theory* (Corresp.), vol. IT-14, pp. 783–784, Sept. 1968.
[8] T. Cacoullos, "Estimation of a multivariate density," *Ann. Inst. Statist. Math.*, vol. 18, pp. 179–189, 1966.
[9] K. Fukunaga and T. F. Krile, "Calculation of Bayes' recognition error for two multivariate Gaussian distributions," *IEEE Trans. Comput.*, vol. C-18, pp. 220–229, Mar. 1969.

An Interactive System for Reading Unformatted Printed Text

ROBERT N. ASCHER, GEORGE M. KOPPELMAN, MARTHA J. MILLER, GEORGE NAGY, MEMBER, IEEE, AND GLENMORE L. SHELTON, JR.

Abstract—A system intended to provide input of printed text to computers is applied to published patents, annotated law reports, and technical journals.

The principal improvement over previous methods is the elimination of training sets by relying on rejected characters to classify subsequent patterns, with intervention of the operator at the end of each run to attribute alphabetic identities to the classes. Another new feature is the application of a sequential search procedure based on accumulated symbol frequencies to speed classification of the approximately 400 different symbols encountered in a typical publication.

An interactive mode of operation for formatting, scan control, labeling, and post-editing is programmed along with the classification process on an experimental system comprising a small digital computer, an opaque page scanner and monitor, a data-entry tablet, a graphic display console, auxiliary storage units, and a fast digital correlator.

Results are reported on some 150 experimental runs on a total of about 30 000 characters from a dozen different source documents. Recognition rates of the order of 99.75 percent are achieved without resorting to post-editing.

Index Terms—Adaptive systems, data entry, graphic editing, graphic tablet, man–machine interaction, multifont recognition, optical character recognition, optical scanner, printed document reader, text reading.

Introduction

THE OBJECT of this study is to provide guidelines for automating the conversion of unformatted typeset text to computer code. The approach advocated is based on the belief that the extreme variability encountered in even a moderate range of publications precludes the design of a completely autonomous system in the foreseeable future and that the concept of keeping an operator in the system is both practical and economical.

This introduction contains a discussion of possible applications for print readers, an examination of the peculiarities of printed matter which affect the structure of the recognition system, some mention of previous work which we have found useful or relevant, and a rationale for the choices made in the course of the design process. There follows a description of both the experimental apparatus and the attendant program system in sufficient detail to give the reader some idea of the costs involved. The final sections present experimental results on a dozen or so documents, representing a wide variety of publications, in such a way as to render explicit the tradeoffs between the various parameters and also to give some indication of the performance to be expected in a production environment.

Intended Applications

Examples of possible applications of automatic text reading are: the conversion of over 3 000 000 U. S. patents into a form suitable for full-text computer search; coding the various bodies of state, county, and municipal statutes for the convenience of legislators; translating legal case histories into computer-readable form to aid attorneys in quest of precedent; converting the files of large newspapers and magazines into a form allowing more extensive cross-indexing and quicker consultation of articles of interest; automatic translation, should it ever become widespread; preparation of the contents of over 2000 technical journals not presently available in computer-readable form for automatic indexing,

Manuscript received November 25, 1970; revised May 7, 1971.
The authors are with the IBM Thomas J. Watson Research Center, Yorktown Heights, N. Y. 10598.

abstracting, extracting, and retrieval; committing to bulk storage over 4 000 000 existing abstracts in the general field of chemistry; incorporating in present information retrieval systems and terminal utility programs printed bibliographic indices and tabulated material in the engineering sciences.

The conversion of nontechnical material is also of interest. Reading aids for the blind should be able to cope with at least the paperback without illustrations. Publishers are interested in automating the resetting of new editions of out-of-print books. There are also several pilot projects underway to render accessible the contents of a whole library from simple display terminals. While it is, of course, possible to store printed material digitally in a strictly video format, economies of the order of one hundred to one may be expected from storing only the character-coded version of the information.

Most of the tasks listed in the preceding paragraphs are not large enough individually to warrant a development program aimed at producing a machine for just that single application. What is needed is a flexible system capable of being adjusted to a wide variety of possible input formats and type fonts. Let us therefore examine just what characteristics of printed material are likely to give trouble. The discussion will be confined to technical as opposed to entertainment reading, since this is where the bulk of the applications lie and because a machine capable of reading U. S. patents or the *Communications of the Association for Computing Machinery* would in any case have little trouble with *Fanny Hill.*

Style and Format Variability

Foremost among the problems faced by a general-purpose text reader is the profusion of type styles. It is estimated that there are no less than 300 000 symbols in common use in the U. S. printing industry. While some of this variability is accounted for in terms of the type sizes, even within the same face an increase in point size is usually accompanied by a change in the proportions of the character. Among the major font families, such as roman and gothic, boldface and lightface, italic and cursive, condensed and extended, the differences are quite striking, even without considering the esoteric symbol sets of the various scientific disciplines and foreign alphabets.

It is clearly out of the question, if for no other reason than because of the difficulty of collecting a suitable design set of representative sample characters, for a reading machine to have a standing repertory of reference parameters sufficiently extensive to allow it to interpret a significant fraction of all font styles. The alternative is to provide a means of adjusting the machine for each application as efficiently as possible.

Within a single technical publication a reference collection of about 400 symbols should allow the machine to recognize successfully the bulk of the main body of text, including uppercase, lowercase, numerals, punctuation, ligatures, headings, subheadings, boldface, italics, footnotes, subscripts, superscripts, and figure captions and labels.

It is questionable whether detailed analysis of equations and formulas is needed in most applications. For accurate interpretation of such material a thorough understanding of the subject matter is indispensable, and current information-retrieval programs have a long way to go to reach this degree of sophistication. A reasonable alternative is to store such formulas in digital video form.

Photographs and other grey scale material must also be discarded, or saved in facsimile form. The bandwidth reduction techniques developed in television broadcasting, and particularly in satellite photography and picturephone applications, may result in worthwhile savings in storage requirements. The required fidelity in reproduction will of course vary from application to application. Greater savings may be achieved on line drawings where line following and curve fitting methods have been used to advantage.

The principal difficulty in the overall format control of a completely automatic text reader resides not in finding all of the alphanumeric information on a given page, but in scanning, recognizing, and storing it in an acceptable (but generally not unique) sequence. This difficulty may be exemplified by considering a two-column page with a full-width illustration inserted at about the center of the page. After reading the upper portion of the left column, the machine must determine whether the text continues in the same column directly below the illustration, or whether it should transfer to the right-hand column and read the top portion before continuing with the left-hand column. Similar difficulties plague even the human reader, as evidenced for instance by our frequent lack of immediate cognizance of a skipped page.

Literature Review

A sampling of recent work in this area is available in [1]. In addition to a discussion of possible alternatives to optical character recognition for data input, a list of several surveys and anthologies is contained in [2].

There exist several preliminary studies [3], [4], as well as some partial simulations [5]–[8], of the format problem in print recognition, but the only complete system we know of which resembles ours, particularly with respect to the emphasis on dealing with the whole problem, from document to final computer coded text, is the machine developed by Weiss *et al.* (see [3]) at CompuScan, Inc.

The CompuScan scanner has built-in line-finding and normalization, as well as the ability to rotate the scan coordinates to rectify skewed characters or lines. The main recognition algorithm is based on correlation, with the addition of an unspecified feature-detection stage to differentiate difficult confusion pairs. Training takes place by having the machine display every unrecognizable character. Up to 800 character classes may be processed simultaneously. We have been unable to find any account of the actual performance of this system.

At least two other laboratories [10]–[12] have been active in text reading for a number of years, but because their principal objective is the construction of a reading aid for

the blind, their philosophy and methods differ in many respects from ours. In general, they do not seem too concerned about the speed obtainable in a production environment (since the output rate is in any case limited by human comprehension), or the error rate, or rapid extendability to large symbol sets, but devote greater emphasis to the output media. Both of these laboratories report very low error rates on highly restricted small data sets.

Here at IBM, much of the early work was directed at the multifont typewritten character recognition problem, and resulted in construction of the IBM 1975 reader for the Social Security Administration earnings reports [13]. Shelton's work on correlation [14], Raviv's experiments with letter context [15], and several different approaches to adaptive logic [16], [17] have also proved useful.

Preliminary work on format control was reported in [18], and some early results with the present system in [19]. Additional detail on the current series of experiments is available in an IBM report [20].

Method

The overall scheme is as follows. At the outset, the operator specifies, by means of a stylus-pointer, which areas of the page are to be processed in the text-reading mode, omitted completely, or processed in a facsimile mode—not discussed in this paper—suitable for diagrams, photographs, and complicated mathematical expressions. Processing is continued in the text-reading mode, where the details of character acquisition, such as line-finding, centering, normalization, registration, and segmentation are under direct computer control.

The next step is a clustering process based on cross correlation. The machine, in effect, saves an example (the "prototype") of each new pattern class it encounters; this includes hitherto unseen styles or fonts, esoteric symbols, blotched or otherwise mutilated characters, and missegmented fragments or groups of letters. At the end of a batch of suitable size, the video bit pattern of each prototype is displayed on a screen, and the operator keys in the appropriate label (alphabetic or symbolic identity, including, if desired, a coded font identification). This label is then applied by the machine to each member of the corresponding class.

The final step consists of proofreading and post-editing displayed versions of entire pages, with special attention to critical passages and phrases, such as names, titles, and references. In the display, characters recognized by the system with an insufficient margin of certainty are intensified on the screen to attract the operator's attention.

In this attempt to define a complete system for reading text, relatively little weight is given to the recognition of isolated characters. In typeset text, one may generally depend on high print quality; in fact, most of the variation between different samples of the same symbol tends to be introduced by the character acquisition process. The main concern is the efficient utilization of the machine. In view of the basic cost of the optical scanner, it is essential to operate at a high rate of throughput; the scanner cannot be allowed to pause while the operator keys in parameters to aid in the recognition process. On the other hand, it is unreasonable to expect the operator to hunt through a whole publication in quest of examples of rare characters, such as boldface *q*, for training the machine.

Data Acquisition

The computer facility used is described in detail in [21]. The nucleus of the system is an IBM 1800 digital computer. Built-in analog-to-digital and digital-to-analog conversion allows access to custom-built units such as the optical scanners, while digital input and output registers facilitate communication with a graphic input tablet, an IBM 2250 vector display, and a binary correlator which permits bit-by-bit comparison of two 768-bit patterns in 40 different shift positions. Auxiliary storage in the form of disks, magnetic tapes, and a high-speed core array [22] supplement the 1800's 32K 16-bit words of memory.

The optical input is provided by two flying-spot scanners, one for $8\frac{1}{2}$-in by 11-in pages, and one for 35-mm transparencies. The spot from a cathode-ray tube is focused upon the document, and the amount of light transmitted or reflected is measured by a photomultiplier tube light-collection system. A threshold circuit is used to determine whether the pattern was black or white at the coordinate location corresponding to the deflection voltages set under program control. Several refinements intended to ensure repeatability, linearity, positional stability, adequate resolution, and a consistent photometric response at relatively high scan rates are described in [21]. A cathode-ray display tube slaved to the scanners allows direct monitoring of the scanning process.

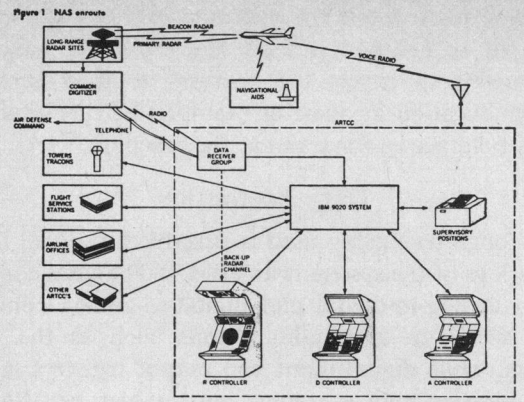
Field Definition

The principal tool used by the operator to specify which areas of the page are to be submitted to the line-finding and character-scanning programs is a graphic input tablet with a stylus.

In principle it is sufficient to place a copy of the page on the tablet, and use the stylus to outline the areas to be scanned. It is, however, convenient to show on the monitor tube both a coarse rendition of the page, as seen by the scanner, and the precise position of the stylus.

As the stylus is brought close to the tablet, a bright spot on the screen is superimposed over the picture of the document in a position corresponding to that of the stylus on the tablet. The operator brings the stylus to the corner of the field to be processed, then draws a diagonal stroke to the opposite corner, meanwhile pressing down on the stylus to activate it. The maximal x and y excursions attained by the stylus before it is raised are calculated from the coordinate information provided by the tablet, and the corresponding rectangle is overlaid on the display for the operator's inspection. Fig. 1 shows the trace of the pen as well as the final rectangle.

The field selection method implemented is relatively slow and requires an expensive graphic tablet and display device.



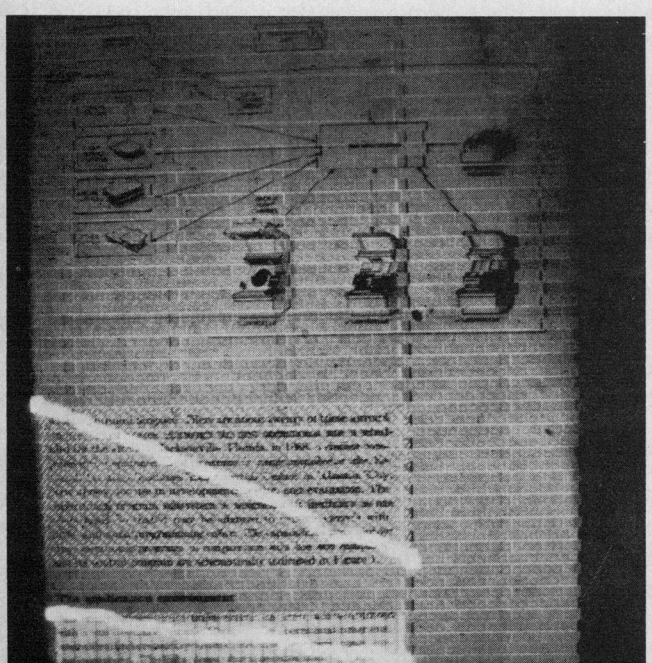
within controlled airspace. There are about twenty of these ARTCC's in the United States. Although the first operational site is scheduled for the ARTCC in Jacksonville, Florida, in 1968, a similar complement of equipment and programs is being installed at the National Aviation Facilities Experimental Center in Atlantic City, New Jersey, for use in development, testing, and evaluation. The operational program subsystem is designed with flexibility as one of its goals, so that it may be adapted to various ARTCC's with little additional programming effort. The capabilities provided by the operational programs in conjunction with the 9020 equipment and its control program are schematically indicated in Figure 1.

The application environment

As a center for enroute air-traffic control, an ARTCC will be equipped with a 9020 Central Computer Complex, the operational programs, controller displays and keyboards, and other peripheral input/output equipment. The flight plans, which are submitted by the pilots

APPLICATION PROGRAMS 125

(a)



(b)

Fig. 1. Field definition. (a) The page to be processed. (b) The trace of the stylus on the display. In this time exposure one may also see the rectangular overlay which helps the operator to confirm that the boundaries have been correctly demarcated. In this example, two texts fields have been selected.

Its main advantage is that the system is practically foolproof and required only a modest amount of programming support. Marking unwanted portions of the page on the document itself (in ink to which the scanner is highly responsive), and letting the line-finding logic locate and process these marks, may be an acceptable alternative solution in many applications. The main drawback is the possibility of missing marks and the difficulty of conveying precise information (such as resolution or number of grey levels) to the system for the processing of nontextual fields.

Line Finding and Character Acquisition

The line-finding routine is designed to locate characters (of any size) within any specified area of a page (Fig. 2). The program causes the horizontal scan to progress downward to find lines of print, computes the raster size based on the height of each line, and initiates the character scan to isolate the characters.

Even if the characters scanned are of almost uniform density, the light output of the scanner varies with time as well as with position on the document. Therefore, the threshold, which the machine uses to separate black from white points when producing binary video, must be periodically recomputed throughout the field being scanned.

The actual acquisition of characters is performed by a binary vertical scan of 32 bits moving across the line of print from the left margin of the field to right (Fig. 3). Each character is converted into a binary matrix 32 bits high by a variable width and registered within the matrix in a standard position for subsequent recognition.

The separation of individual characters remains a difficult problem of character recognition, particularly on variable-pitch material. Elaborate algorithms exist for fixed-pitch gothic and roman-type styles [23], [24], but in our case the limiting factor was the spot size obtainable with our scanners. The segmentation procedure used is shown in Fig. 4; its net effect is to eliminate small fragments and to facilitate segmentation towards the end of a long connected region. An example of successfully segmented italics is shown in Fig. 5.

Character Classification

The output of the scanning program is a tape containing one record per character. At this point, the tape would be ready for the clustering phase if the system were set up for actual use. Since it was designed for experimental purposes, including determination of error rates and confusion matrices on various types of input, the identity of each character is keyed in by the operator upon visual inspection of the video bit pattern and inserted in the header accompanying each character on the tape.

The video-display and keyboard routine implemented on the IBM 2250 is used with little change in different phases of the overall process. The essential feature of this routine is the display of the video image of a string of scanned characters, followed by the association of appropriate "identification" or "label" tags, entered by the operator, with each character (we consistently use the terms "identity" or "identification" for alphanumeric tags inserted only for experimental purposes, and "label" for tags assigned by either the operator or the machine in the course of the actual operation). To decrease the frequency of operator errors, the character code keyed in by the operator is also

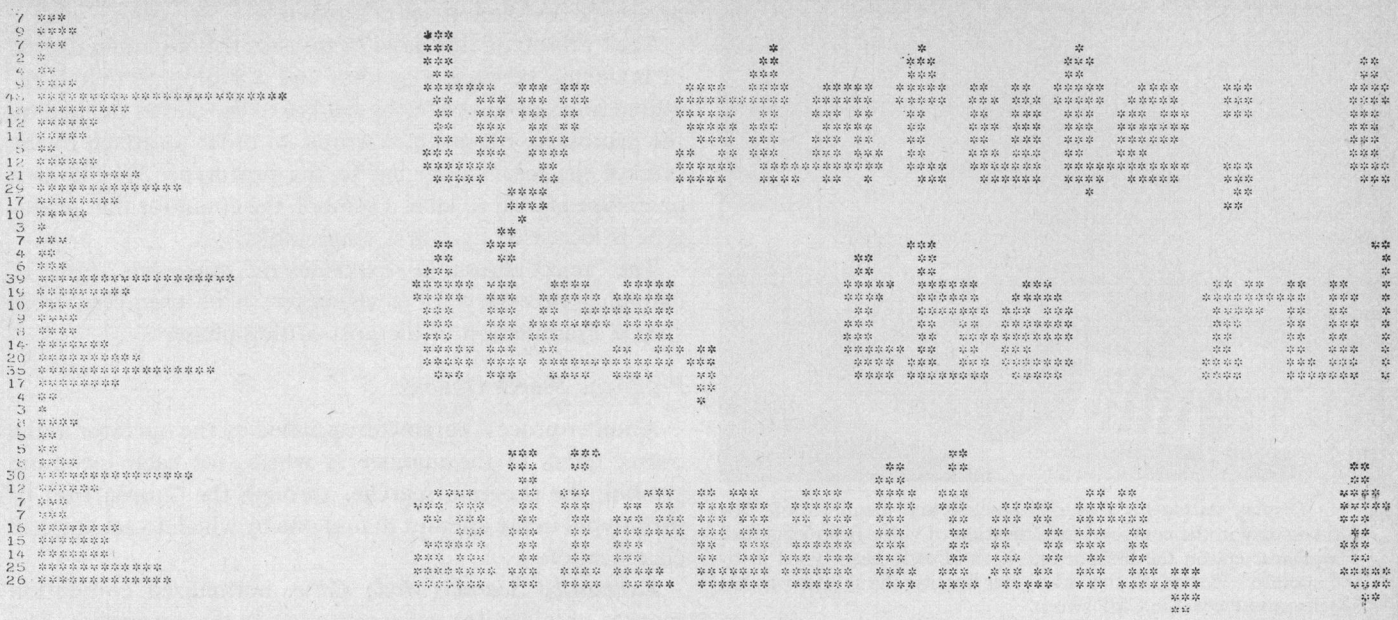


Fig. 2. Line finding. To the left is the histogram showing the number of bits in the EXCLUSIVE OR of successive pairs of horizontal scans. The occurrence of a value below a preset threshold indicates to the line-finding program that it has found the top or bottom of a line; the height of the line is calculated, however, from the maxima which are more stable. To the right is the video at the resolution seen by the line-finding scan.

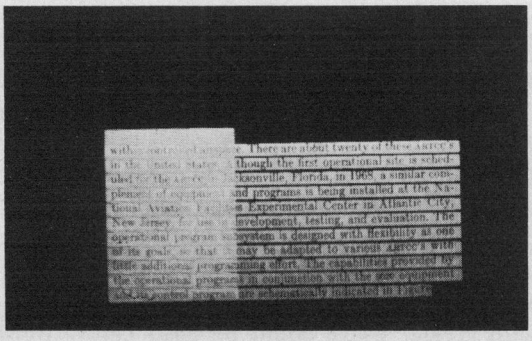


Fig. 3. Character acquisition. The brighter region on the left is created by a time exposure of the horizontal line-finding scan. The character acquisition scan, consisting of short vertical strokes, is initiated as soon as a line is found. The shutter was closed just before the process reached the end of the upper text field shown in Fig. 1.

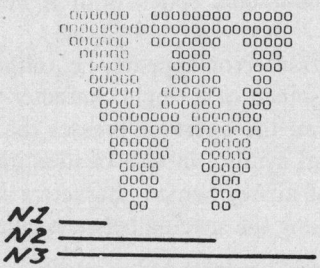


Fig. 4. Regions of segmentation. Any pattern which does not extend beyond the first region is discarded. In the center, a preset number of blank scans are required to end the character. To the right, a condition is imposed only on the number of black ("1") bits in the AND of successive scans. Beyond the third region the pattern is arbitrarily truncated.

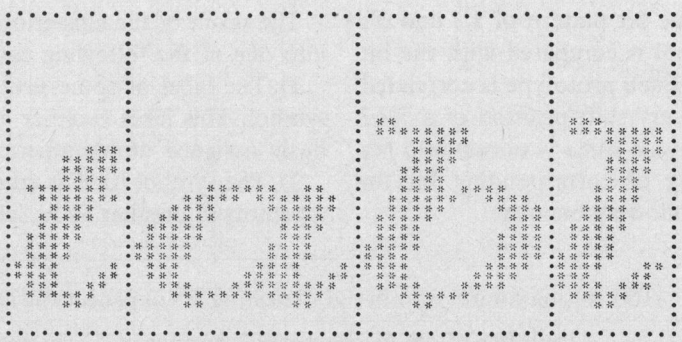


Fig. 5. Well-segmented italics. The segmentation scheme described in the paper is not as powerful as the more time-consuming technique of looking for white paths of certain acceptable orientations, but it performs suprisingly well on this modern italic typeface from the *IBM Syst. J.*

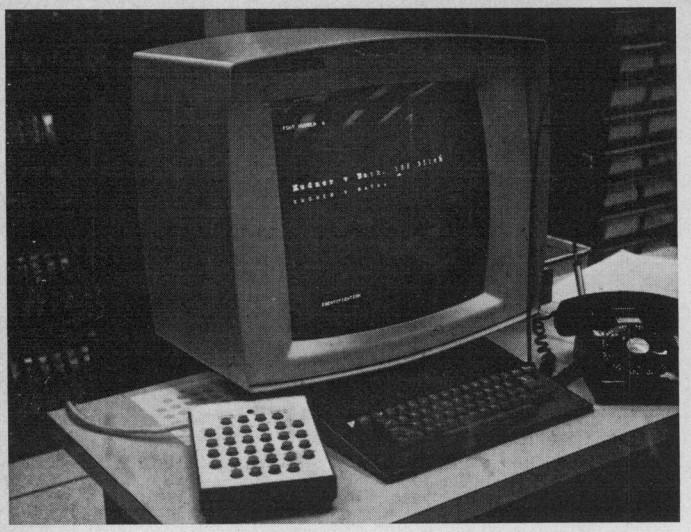


Fig. 6. Display station for character labeling and identification. The 2250 display in the course of identification of video from legal text. The characters on the bottom line have already been typed in by the operator. Part of the line is brighter because the shutter was not synchronized with the CRT sweep.

displayed, as a "stick character," below the video image (Fig. 6).

In addition to the individual character identities, the operator may also enter a font-identifying code. Once a font code has been entered, it is automatically attached to all subsequent character codes until a new font code is specified.

If a segmentation error appears, i.e., either two adjacent characters were joined or a single character was split by the scanning program, the operator presses the "#" key which enters this special symbol in lieu of identification or label. The detection of missegmented characters is facilitated by artifically increasing the spacing between adjacent characters on the screen.

Typographic ligatures (the commonest being "fi") are considered as single symbols rather than rejects, and are identified by special codes. Two-character codes may also be defined as needed for any symbol which has not been previously coded as a stick character.

Correlation Against Prototypes

In the clustering operation the bit pattern of 1's and 0's representing an incoming symbol is compared with the bit patterns of prototype symbols. Each prototype is correlated with the incoming symbol in every shift position of a 5×7 matrix, and the maximum of these sums is normalized (to take into account mismatching of corresponding bits in symbol and prototype) in the following manner:

$$\text{score (percent)} = \frac{100 \times (\text{maximum number of matching 1's in prototype and character})^2}{(\text{number of 1's in character})(\text{number of 1's in prototype})}.$$

Thresholds

Among the process parameters that the operator must supply to the system at the outset are two thresholds, against which the normalized scores resulting from symbol versus prototype correlation are compared.

The "prototype threshold" expresses the minimum degree of matching between a symbol and a prototype that is required in order to assign the symbol to the cluster defined by the prototype, or, in other words, in order to attach to the symbol the *label* belonging to the prototype. Whenever a prototype is used to label a symbol, the count for that prototype is increased by 1 in a usage table.

The "reject threshold" expresses the minimum degree of matching necessary for a character to be exempted from special examination in the post-editing phase.

Prototype Search Options

Another process parameter specified by the operator at the outset refers to the manner in which, for each incoming symbol, the program searches through the (current) set of prototypes in the attempt to find one by which to accomplish classification.

Exhaustive Search (Mode 1): A normalized correlation score is obtained for every prototype in the current set. The highest of these scores is then tested against the prototype threshold.

Sequential Search in Order of Occurrence (Mode 2): Prototype number 1 is tried first. If its normalized correlation score does not exceed the prototype threshold, prototype number 2 is tried, and so on. The process stops with the first prototype that satisfies the threshold, or when the list of prototypes is exhausted. The purpose of this procedure is to reduce the average number of prototypes that will be tried per symbol.

Sequential Search Ordered by Usage (Mode 3): The search is ordered by the usage table starting with the largest entry. It extends, however, into the zero-usage prototypes at most to a depth prespecified by the operator. The usage table is ordered in such a manner that not only will prototypes with greater usage count be tried before those with lesser, but within a set of prototypes all having the same usage count, they will be tried in inverse order to that in which they attained their current count. Each prototype is thus given a chance to establish itself, but is eventually dropped from consideration if it does not continue to participate in the classification process.

The result of the clustering phase is that each symbol falls into one of the following categories.

1) The label of some prototype has been applied to the symbol. This label is either a prototype number or a previously assigned alphanumeric code.

2) The symbol itself is taken as a prototype and assigned a prototype number as its label.

Prototype Labeling and Decoding

Prototype labeling, an operation performed by means of visual inspection on the 2250, produces a file containing an

alphanumeric code for each prototype created in the clustering phase. This file is indexed by the prototype number.

The decoding program makes use of this file to modify the output of the clustering program. Specifically, it runs through the successive symbols and for each one that has a prototype number for a label, looks up the corresponding alphanumeric label in the file and substitutes this alphanumeric label for the numerical label previously possessed by the symbol.

Due to the availability of the true identities of the characters, the prototype labeling and decoding phases need not actually be performed in an experimental run unless one wishes to go on to post-editing.

Post-Editing

This program allows an operator to review and edit the output of the classification programs before final output. The operator makes two passes through the text. In the first pass, *reject verification*, he is presented with several lines of text, displayed on a 2250 screen as they were identified by the classification program (Fig. 7). Reject characters are displayed more brightly than the rest: the operator must check each rejected character, confirming or correcting it. The program keeps a record of how many times the operator disagrees with the prestored "true" identity.

When there are no rejects remaining, the *editing* pass begins. Now the operator may insert or delete characters, words, or whole lines at the position of the pointer. The pointer is repositioned whenever necessary with the light pen on the 2250. Segments of text may be inserted by typing on the alphanumeric keyboard, or deleted by means of special keys.

When the operator indicates that the currently displayed text is acceptable, it is printed out in its final form. Fig. 8 shows the corrected version of the segment shown in Fig. 7, as displayed on the screen prior to printing. Another group of lines is then read, and the procedure described above is repeated. The line arrangement of the original document is preserved on the display for ease of proofreading.

The original text and the final printout are shown in Fig. 9, together with some information on the speed and accuracy of the editor.

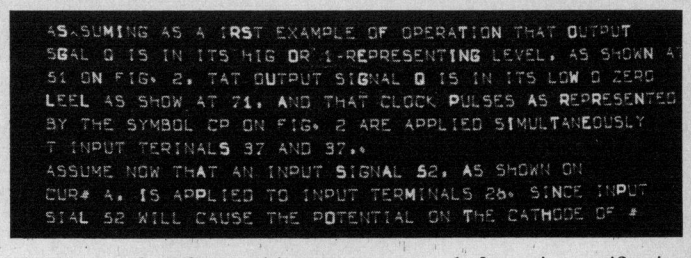


Fig. 7. Display of recognition system output before reject verification and editing. Rejected characters appear brighter than the rest to attract the operator's attention. There are also several other errors, mainly due to missegmentation. The caret points to the first reject.

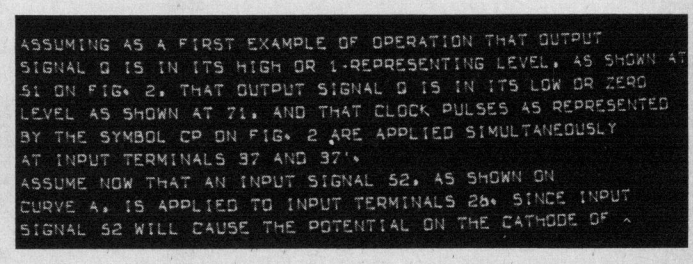


Fig. 8. Display after reject verification and editing. The operator is now ready to press the button to bring in the next section of text. This section is now in the form of the final output.

Assuming as a first example of operation that output signal Q is in its high or 1-representing level, as shown at 51 on Fig. 2, that output signal $\bar{Q}$ is in its low or zero level as shown at 71, and that clock pulses as represented by the symbol CP on Fig. 2 are applied simultaneously to input terminals 37 and 37'.

Assume now that an input signal 52, as shown on curve A, is applied to input terminals 28. Since input signal 52 will cause the potential on the cathode of

```
ASSUMING AS A FIRST EXAMPLE OF OPERATION THAT OUTPUT
SIGNAL Q IS IN ITS HIGH OR 1-REPRESENTING LEVEL, AS SHOWN AT
51 ON FIG. 2, THAT OUTPUT SIGNAL Q IS IN ITS LOW OR ZERO
LEVEL AS SHOWN AT 71, AND THAT CLOCK PULSES AS REPRESENTED
BY THE SYMBOL CP ON FIG. 2 ARE APPLIED SIMULTANEOUSLY
AT INPUT TERMINALS 37 AND 37'.
ASSUME NOW THAT AN INPUT SIGNAL 52, AS SHOWN ON
CURVE A, IS APPLIED TO INPUT TERMINALS 28. SINCE INPUT
SIGNAL 52 WILL CAUSE THE POTENTIAL ON THE CATHODE OF


OF    250 REJECTS,
      247 WERE ENTERED CORRECTLY
        3 INCORRECTLY

NUMBER OF CHARACTERS ENTERED
      104 AT CHARACTER LEVEL
        0 ON WORD LEVEL
        0 ON LINE LEVEL

NUMBER OF CHARACTERS DELETED
       34 ON CHARACTER LEVEL
        0 ON WORD LEVEL
        0 ON LINE LEVEL

     1990 CHARACTERS PROCESSED

TIMING
    2.04 MINUTES FOR PROGRAM TO SET UP DISPLAY
    2.28 MINUTES CORRECTING REJECTS (    0.009 PER REJECT)
    5.58 MINUTES EDITING
   10.92 MINUTES TOTAL FOR THIS RUN
```

Fig. 9. Printout of edited segment of text and log. The original printed text is on top, the version produced by the system below. The log traces the work of the post-editor and records how much time the operator and the system expend waiting for each other. The times listed refer to processing two columns of print rather than just the segment shown above.

Performance

The experiments reported were performed on the various data sets illustrated in the Appendix. Data collection, i.e., scanning the documents, took place over a span of about twelve months, during which both scanners underwent considerable change and some improvement. We have attempted to evaluate the effect of various parameters only on experimental runs performed on identical data sets, and the reader should also be wary of drawing conclusions from comparable runs on similar, but not identical, character sets.

Altogether some 30 000 characters were scanned, identified, and processed in one or several recognition modes. As the quality of the data improved as a result of changes in the hardware, we shifted our attention to the latest and best character sets, even at the cost of losing some comparability. Since some 150 separate recognition runs, amounting to about 200 h of computer time, were required, all of the experiments could not be repeated on a more restricted set of standardized data.

The important design characteristics of any character recognition system are *throughput, error/reject rate, equipment cost, and manpower* required for operation. While none of these characteristics may be directly estimated from an experimental system, each is reflected in some observable

Tape Number	Document	Skip	Total	Initial Prototypes	Mode	Depth of Search	Prototype Threshold	Are #'s Processed	Number of #'s	Error Rate (%)	Mean Length of Search	Fraction Saved as Prototypes
4	ALR		1500		3	25	80	Y	28	0.06	40.0	18.1
4		1500	1400	124	3	100	75	N	59	1.12	23.2	4.4
			1500		3	25	75	Y	27	0.93	22.2	10.5
			1500		1	25	75	N	28	0.66	93.3	9.0
			1500			25	50	N	28	45.21	22.3	2.1
			1500		3	100	80	N	28	0.0	47.8	16.5
			2900		3	25	75	N	88	0.76	23.1	7.1
			1500		3	25	75	N	28	1.06	22.1	8.9
			1500		3	100	75	Y	27	1.13	24.7	10.4
			1500		3	100	80	Y	27	51.0	12.1	12.1
			1500		3	25	77	Y	27	0.73	26.0	12.8
			1500		3	100	77	Y	27	0.86	28.3	11.9
			1500		3	100	78	Y	27	0.80	34.1	13.5
			1500		3	10	77	Y	27	0.73	22.8	17.5
			1500		3	5	77	Y	27	0.73	20.0	19.1
			942		1	2	77	Y	27	0.53	14.0	42.3
			1000		3	100	75	Y	16	1.00	24.4	13.1
		1000	1000	73	3	100	75	Y	28	1.50	23.5	7.7
		1000	1000	73	1	100	10	Y	28	7.50	37.0	
		1000	1000	73	1	100	10	Y	28	9.40	73.0	
		1000	1000		3	100	75	Y	28	0.50	19.5	12.3
			2374		3	100	80	Y	58	0.17	58.6	16.8
282	ALR		800		1		80	N	25	0.12	69.2	14.0
			800		3	400	80	N	25	.62	19.8	14.1
			800		2		80	N	25	.75	25.6	14.1
			800		2		80	N	25	.75		14.1
			600		2		75	N	13	7.00		8.0
			132	107	2		80	N	2	7.00	121.5	31.8
			800		3	25	80	Y	25	0.50	17.5	15.1
			800		1		50	N	25	70.50	12.8	
					1		65	N	25	12.75	27.0	
					2		70	N	25	21.12	12.0	5.37
					2		75	N	25	7.00	14.1	7.50
					3	25	80	N	25	0.50	17.5	15.12
					3	25	75	N	25	6.62	9.1	7.50
					3	25	70	N	25	23.25	7.1	5.37
556	ALR		500		2		80	N		0.0	33.6	21.4
			500		2		80	N		0.0	33.6	21.4
			500		3	400	80	N		0.0	27.5	21.4
			500		2		75	N		0.8		12.6
			500		2		70	N		3.0	12.0	9.0
			500		1		70	N		0.6	31.1	9.0
			500		2		70	N		3.0		9.0
			500		2		65	N		12.6		6.4
62	PATENT		1000	100	1		10	Y	25	4.3	100.0	
			2975		3	100	75	Y	172	1.53	48.9	13.4
		3000	2800	3	100	75	Y	97	1.56	46.3	14.0	
		5500	717		3	100	75	Y	21	1.25	27.9	19.2

Fig. 10. Correlation runs. Each tape number refers to a different data set. The relations among the various parameters are explored in more detail in subsequent diagrams.

quantity in the experiments, and one can at least arrive at tentative conclusions regarding the tradeoffs among them.

Throughput, in this context, is inversely proportional to the average number of prototype characters examined before reaching a decision. It matters little that the comparison process is about one thousand times slower than it would be in an operational system; if throughput can be improved by a factor of 2 by an ordered search, a proportional saving would result in a real system.

The *error rate* is simply the number of characters to which the system attributes an identity different from that given by the typist who tagged each character after the scanning process. This includes characters which the typist could not identify (because they were mutilated) but to which the machine nevertheless assigned an alphabetic label.

The *reject rate* is the number of characters tentatively recognized by the machine, but which stand in need of confirmation by the post-editor. A good reject criterion would reject almost all of the erroneously labeled characters and only a small fraction of the correctly labeled characters. We have, however, not succeeded in finding such a scheme for the ordered search decision method.

Finally, *manpower* is the sum total of the amount of time required for field selection, for the labeling of prototypes, and for post-editing. These "human factors" figures are given in [20].

A good cross section of typical recognition runs is tabulated in Fig. 10, which lists the results obtained on different data sets for a variety of parameter settings. All of the curves chosen to illustrate particular tradeoffs contain material from this table. Note that some of the runs include all "operator rejects," which are characters tagged as segmentation errors by the operator, while other runs exclude these with the purpose of focusing on the classification errors only. Again, some runs were made with an exhaustive search procedure, in others the prototypes were searched in the order they were generated, while still others used the sequential search according to usage. The "length of search"

Tape Number	Document	Skip	Total	Initial Prototypes	Mode	Depth of Search	Prototype Threshold	Are #'s Processed	Number of #'s	Error Rate (%)	Mean Length of Search	Fraction Saved as Prototypes
57	PATENT		2800		3	100	75	Y	78	0.66	33.6	9.0
			2000		3	100	76	Y	64	0.25	48.7	14.5
			2000		3	100	77	Y	64	0.40	41.6	13.2
33	PATENT		1169		3	100	75	N	40	0.51	51.9	20.1
		4200	1692		3	25	75	Y	121	0.70	36.7	23.6
			4200		3	100	75	N	223	1.35	42.4	9.3
			2500		3	25	75	N	108	1.00	34.8	17.2
			2100		3	400	75	N	104	0.95	48.6	14.1
			4200		2		75	N	223	1.59	64.8	9.1
		4200	1659	100	3	100	75	Y	122	1.12	53.6	17.6
		1500	1861	96	3	96	75	Y	90	1.05	53.3	16.3
		2700	1500	123	3	123	75	Y	81	1.66	36.9	9.6
		1500	1200	96	3	96	75	Y	61	1.25	56.6	20.0
			1500		3	25	75	Y	55	0.86	26.7	14.2
347	PATENT		1000		2		80	N	40	0.66	51.8	20.1
5	PATENT		1100		1		75	N	24	0.45	76.7	10.3
			2700		3	400	75	N	90	1.37	43.3	10.4
			2181		2		75	N	23	2.33	48.8	9.8
			1925		2		75	N	45	1.50	38.4	10.5
			1800		3	25	75	Y	44	1.33	21.6	13.4
352	IBM J.		517		2		80	N	243	37.9	30.3	20.7
557	IBM J.		900		3	400	80	N	70	0.80	48.6	22.0
			900		1		60	N	70	7.60	33.2	5.3
			900		2		60	N	70	22.33	10.8	5.3
			900		2		70	N	70	3.44	20.2	8.8
			900		2		65	N	70	12.00		6.5
			900		3		80	Y				
563	DOLPHIN		750		2		80	N	62	0.66	38.6	18.9
			750		3		80	N	62	0.66	30.2	18.9
			750		3		80	N				
552	HURAKAN		1400		3	400	70	N	34	1.15	48.7	16.2
			1400		3	400	70	Y				

Fig. 10 (continued).

is simply the total number of correlations performed divided by the total number of unknown characters. The "fraction saved as prototypes" refers to the number of prototypes available at the end of the run. In most runs the log was routinely printed out every 100 characters (Fig. 11) in order to provide a trace of performance throughout the run, and to prevent catastrophic loss of performance figures through hardware failure.

Thresholds and the Error-Throughput Tradeoff

The actual setting of the prototype and reject thresholds is of no direct concern to a potential user of the system, yet for the experimenter these are important control parameters. Fig. 12 shows the dependence of the error rate on the prototype threshold, with forced decision and usage-ordered search. Fig. 13 shows the dependence of the throughput on the prototype threshold for the same data. The information in these two figures is then combined in Fig. 14, which shows explicitly the variation in error rate with throughput.

In a similar manner one may combine the information on Figs. 15 and 16, depicting the variation in the substitution error rate and reject rate with the reject threshold, and obtain the unprepossessing error/reject curve in Fig. 17. As mentioned before, this inadmissible tradeoff curve represents, in our opinion, the weakest point of our experimental results.

Search Procedures

The exhaustive (parallel) search procedure consists simply of comparing each unknown character against every available prototype, and assigning it to the class of the prototype with the highest correlation score above threshold. In the event none of the scores is above threshold, the character is saved as a prototype. When the same number of references are available to both parallel and sequential procedures, the parallel method always yields a lower error rate than sequential search, albeit at the expense of much lower throughput.

Fig. 18 shows the change in error rate with the average number of prototypes examined. Forced decision was used in order to avoid having to weigh the relative importance of rejects and outright substitution errors. Clearly, *for the same*

I N T E R I M S T A T U S R E P O R T

RUN IDENTIFICATION PR004 - LEGAL - 4K CHS. - 11-5-69
RECOGNITION MODE 3
MAXIMUM NO. OF ZERO-USAGE REFERENCES TO BE SEARCHED 25
SWITCH SET FOR HARDWARE CORRELATOR
REFERENCE THRESHOLD THET1= 75
IDENTIFICATION THRESHOLD THET2= 77
INITIAL CHARACTER SKIP 0
NO. OF CHARACTERS PUT THROUGH RECOGNITION 1300
NO. OF OPERATOR REJECTS 24
NO. OF REFERENCES--PRIOR 0
--SUSPENDED 132
--TOTAL 132
AVERAGE NO. OF REFERENCES CONSULTED 22.79
OVERALL ELAPSED TIME FOR RECOGNITION 18.24 MINUTES

RECOGNITION STATISTICS

	(ACTUAL COUNT)				(PERCENTAGE)			
	C O R R E C T		S U B S T I T U T I O N		C O R R E C T		S U B S T I T .	
	PENDING	COMPLETE	PENDING	COMPLETE	PENDING	COMPLETE	PENDING	COMPLETE
REJECT	132	XXXXXX	XXXXXX	XXXXXX	10.15	XXXXXX	XXXXXX	XXXXXX
UNCERTAIN	262	0	7	0	20.15	0.00	0.53	0.00
CERTAIN	895	0	4	0	68.84	0.00	0.30	0.00
) BEFORE FORCED) RELABELLING	1289	0	11	0	99.15	0.00	0.84	0.00
DECISION) AFTER) RELABELLING	1289		11		99.15		0.84	

> S CH. SQ. NO. 1347 '1 .1 8= 78

Fig. 11. Interim recognition results from a typical run. Most of the items are self-explanatory. Characters and prototypes involved in substitution errors are listed below the dotted line. Here character no. 1347, an "'" in the main-text type (font-code 1), was mistaken, with a score of 78, for a "," of the same typeface, prototype no. 8.

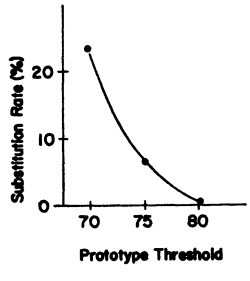


Fig. 12. Variation in substitution rate with prototype threshold. These results were obtained on legal text with usage-ordered search and forced decision. As the prototype threshold is increased, unknown characters are more likely to initiate a class of their own than to be committed (perhaps erroneously) to an existing cluster.

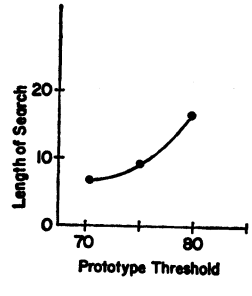


Fig. 13. Variation in length of search with prototype threshold. Same data as in Fig. 12. The length of search is the average number of prototypes examined in reaching a decision for each character. The average includes the cases where all of the prototypes (up to the preset limit) had to be tested and the character set aside as a new prototype.

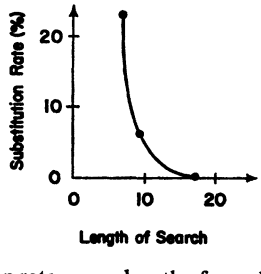


Fig. 14. Substitution rate versus length of search. Same data as Fig. 12. If the prototype threshold is set too high, almost all of the characters will be turned into prototypes, increasing the number of characters to be labeled at the end as well as the length of search.

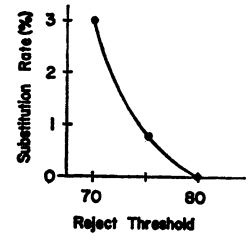


Fig. 15. Dependence of substitution rate on reject threshold. Legal text, fixed-order search. If the reject threshold is high, the number of incorrectly clustered characters with scores above the reject threshold will be small.

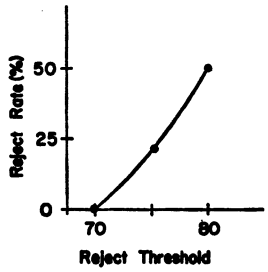


Fig. 16. Dependence of reject rate on reject threshold. Same data as in Fig. 15. As the reject threshold is increased, many of the correctly clustered characters will also be rejected.

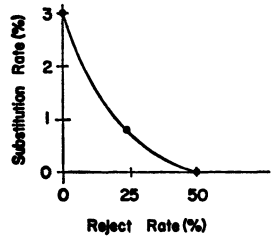


Fig. 17. Error/reject curve. Same data as in Fig. 15. The substitution rate decreases much too slowly with the fraction rejected. Unfortunately, the usual criterion based on the difference in scores between the top candidates is almost meaningless in a sequential scheme.

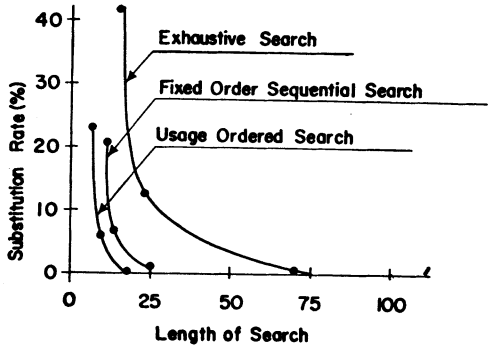


Fig. 18. Variation in substitution rate with the average length of search. Legal text, forced decision. Each point on the graph represents a run on 800 characters. Clearly the usage-ordered search is most effective in yielding the lowest error rate for a given throughput. The different points on each curve were obtained by varying the prototype threshold.

length of search, the sequential procedures make far fewer errors, and conversely require fewer prototypes on the average at the same error rate.

In comparing the fixed-order variable length search with the variable-order search according to usage, roughly the same error rate is achieved with about a 20-percent saving in time in favor of the variable-order search. Of course, the most frequently occurring characters tend to be reached first even in the fixed-order search simply because they were encountered early in the recognition run.

Depth of Search

There are two ways of controlling the average number of references searched, and thus the throughput. Either one limits the depth of search by examining only the more likely candidates, or one decreases the total number of prototypes created by decreasing the prototype threshold.

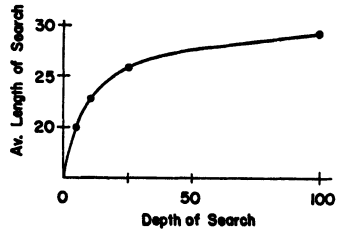


Fig. 19. Length of search and depth of search. Legal text, 1500 characters, forced decision. The number of zero-usage prototypes which may be examined must be curtailed quite sharply in order to have much effect on the average length of search. This shows that acceptable matches are found in short order for most characters.

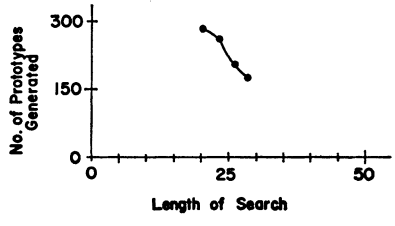


Fig. 20. Prototype generation and depth of search. Same data as in Fig. 19. This curve shows that although the depth of search has little effect on the average length of search, it does influence strongly the number of prototypes generated.

Clearly the direct limitation on the depth of search results in fewer errors for the same throughput; this gain, however, may be offset by the additional number of prototypes which must be eventually labeled by the operator. The relation between the number of prototypes generated, the average length of search, and the depth of search at constant prototype threshold is depicted in Figs. 19 and 20; here the variation in the average number of prototypes was obtained by direct control on the maximum number of zero-count prototypes searched.

The above data on the depth of search were obtained in Mode 3 operation, i.e., order-of-usage search. The exact location of the best operating point would depend on the number of segmentation failures which inevitably show up as zero-usage prototypes, as well as on the relative costs assigned to the correlation and the labeling processes and the cost of errors.

Prototype Generation and Usage

In the unsupervised recognition modes, almost every character at the beginning of a new segment of text is converted into a character prototype. The buildup never ceases completely, but tends to reach some asymptotic rate proportional to the number of mutilated or missegmented characters introduced in the scanning process.

Fig. 21 shows the generation process for both the patent and the law report materials. The cusps in the curves correspond to the transfer from the bottom of one column to the top of the next one, and indicate a lack of uniformity in the scan field. Although there are only about 100 distinct symbols in each of these passages, several examples of the most frequent letters are saved due to wide variations in the video among characters of the same class.

The cumulative distribution function, showing for what

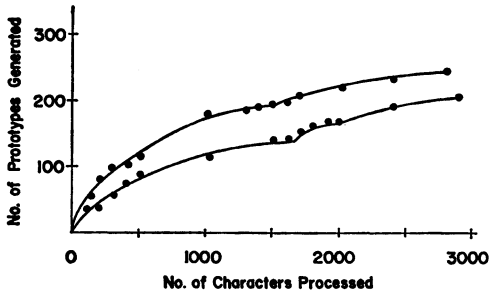


Fig. 21. Prototype generation as a function of the elapsed number of characters. Usage-ordered search and forced decision. Patent text on the upper curve and legal text on the lower curve. The breakpoints correspond to the ends of the columns; the scanner characteristics were markedly different at the top and bottom extremities of the field.

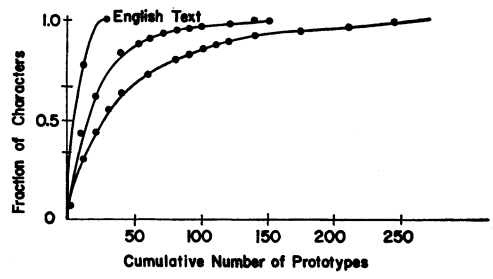


Fig. 22. Distribution of character assignments. As expected, the cumulative distribution function of the fraction of characters for which the most useful prototypes are responsible, shown for two values of the prototype threshold, increases less rapidly than the cumulative singlet probability function for English.

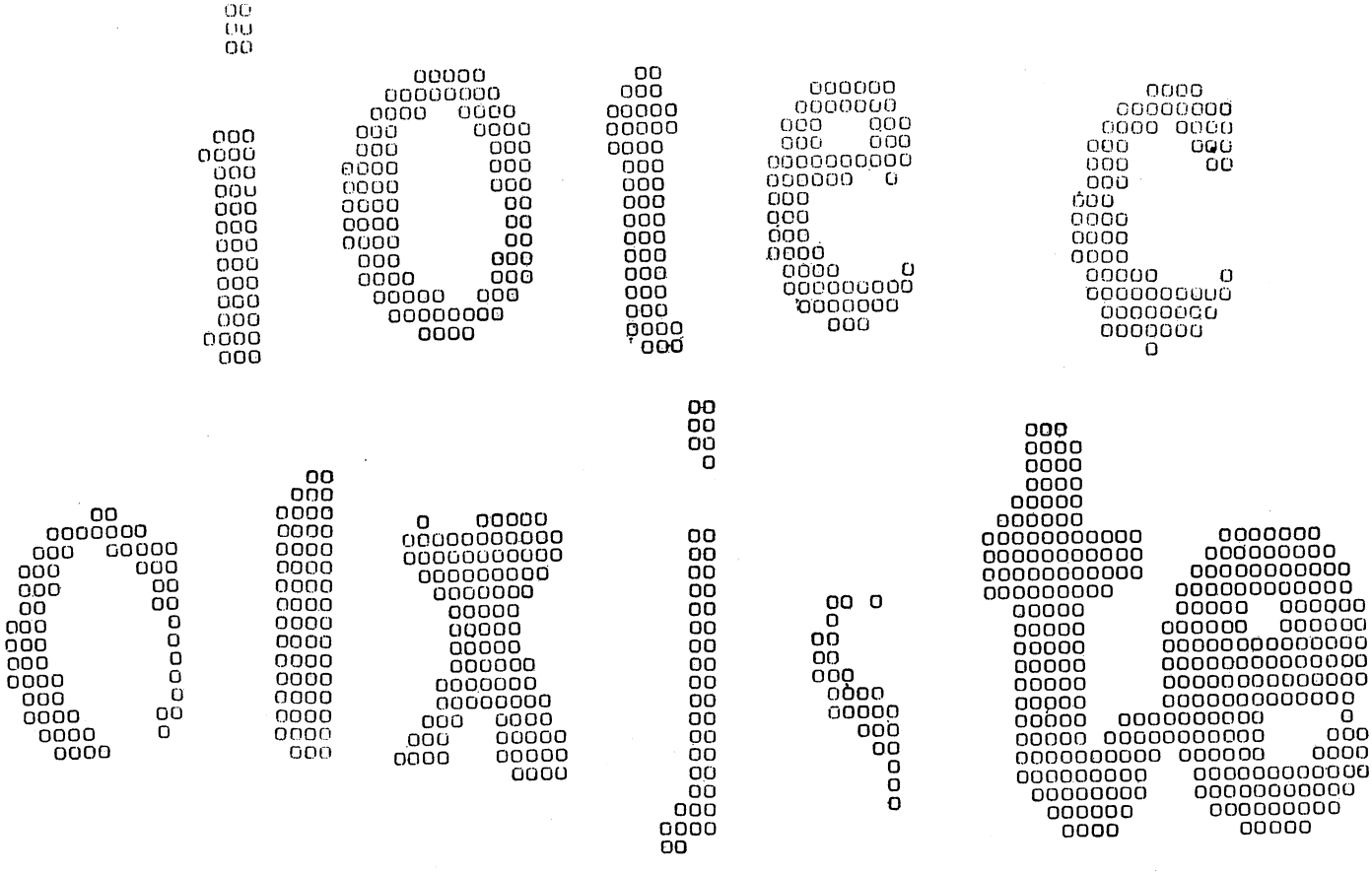


Fig. 23. Examples of prototypes. High-usage prototypes (top row) are clean and well formed. Low-usage prototypes (bottom row) include rare, frowzy, and missegmented characters.

portion of the recognition a given fraction of the prototypes accounts, is plotted in Fig. 22. This curve rises more slowly than the cumulative letter-probability distribution function for normal English (leftmost curve), due partly to the fact that more than 26 symbols are used in text and principally to the noisy nature of the patterns.

The long linear tail of the curve corresponds to the zero-usage prototypes generated from mutilated characters. Curtailing the depth of search effectively eliminates these prototypes from consideration. Examples of high-usage and zero-usage prototypes are shown in Fig. 23.

Other experiments, not reported here in detail but shown in Fig. 10, show that the error rate in the supervised case, using prelabeled prototypes only, is quite comparable to the asymptotic rate of generating new prototypes in the unsupervised case, i.e., to the hard-core residue of unrecognizable characters. Thus the use of preset prototypes results only in a negligible saving in the keying operation.

Substitution Errors

Classification by means of correlation results in very few unexpected errors. Most of the substitution errors in the different types of material occur among only a few confusion pairs, such as (h, b), (1, i), (1, l), (0, O), or among symbols where the positional information should be retained for easy discrimination, as in (,') and (-.). Broken characters result in substitutions only when the fragments are sufficiently small to be confused with punctuation marks, though

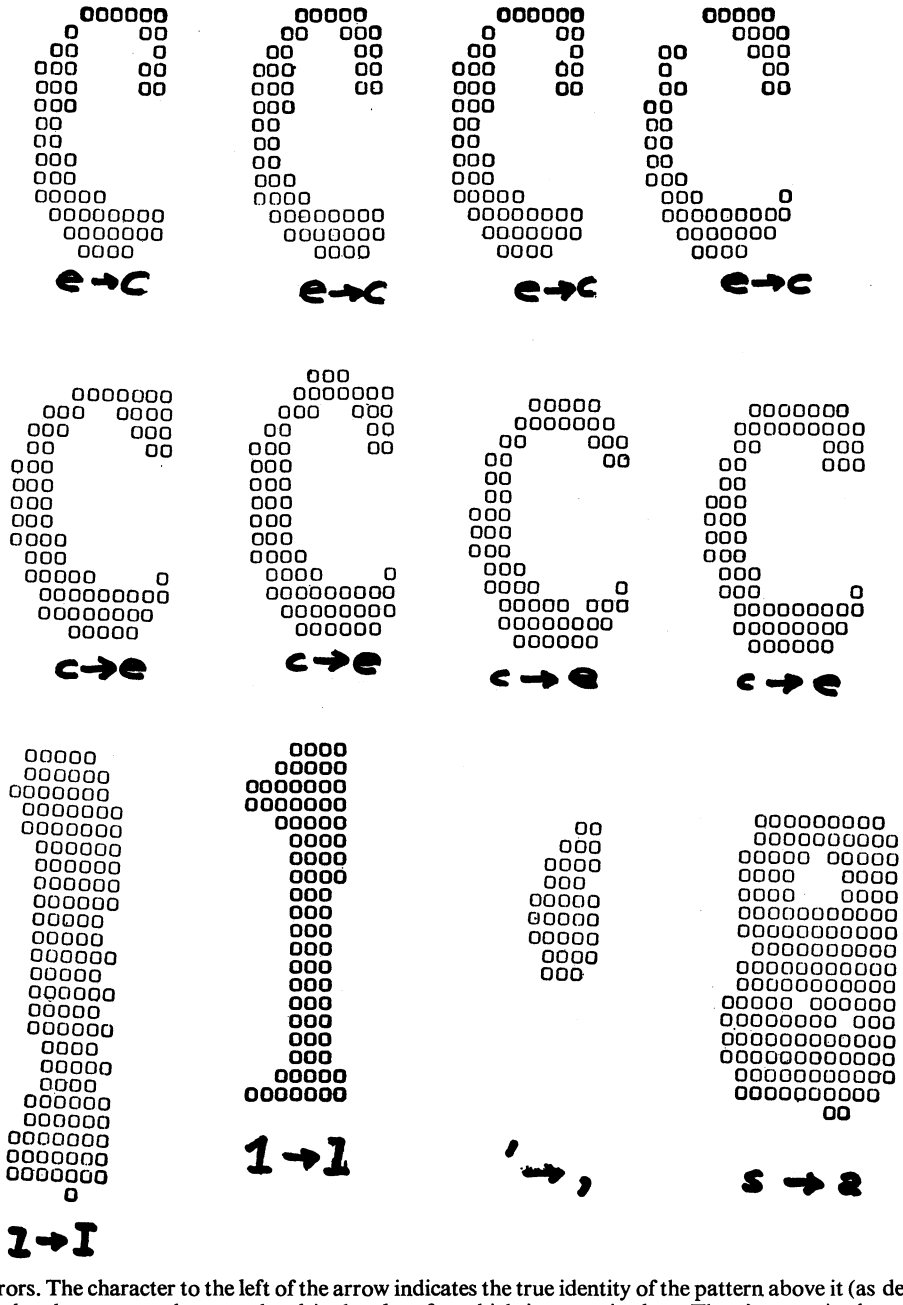


Fig. 24. Substitution errors. The character to the left of the arrow indicates the true identity of the pattern above it (as determined by the operator using context), while the character at the arrowhead is the class for which it was mistaken. The c's are mistaken for e's because one of the "e" prototypes looks like a "c."

sometimes half an "m" is mistaken for an "n." Fig. 24 shows typical errors.

The effect of omitting missegmented characters from the run is shown in Fig. 25 for several types of data. Of course, if the resolution of the scanner were sufficient to avoid missegmented characters, probably some of the other mistakes would be also eliminated, for instance those due to disappearance of the thin horizontal stroke in "e." Typical segmentation errors are shown in Fig. 26.

Representative results are tabulated in Fig. 27 to give an idea of the performance to be expected in typical operation.

Conclusions

The use of the stylus for field selection seems practicable on pages with a relatively simple format resulting in a small number of fields, such as a journal with photographs and other illustrations but few equations and formulas. For very simple formats, as of the U. S. Patents, for instance, a completely automatic scheme would suffice for the vast majority of the pages. For a mathematical journal, however, either a better display, or adequate means of defining the position of the document in the scanner in relation to the data tablet, is required. In experimentation, the principal advantage of the on-line selection proved to be the ease of directing the scanner to any area of interest.

The character-acquisition routines differ from previous methods in the elimination of multiple scans of the same character. This is achieved through line-by-line normalization and centering. Line finding is quite reliable even on crowded pages, though occasionally a line is still missed.

The use of unsupervised learning, or clustering, to avoid having to prepare labeled training samples, proved work-

Data	Substitution Rate (%)		Average Length of Search		Fraction Saved as Prototypes (%)		Fraction Missegmented	
	with #	w/o #	with #	w/o #	with #	w/o #	with #	w/o #
1500 ch ARL	0.06	0.00	51.0	47.8	0.18	0.16	1.80	1.86
1500 ch patents	0.86	1.06	26.7	26.2	0.14	0.14	3.65	3.65
1400 ch Hurakan	1.22	1.15	49.0	48.7	0.18	0.16	3.45	3.45
900 ch IBM J.	0.90	0.80	53.5	48.6	0.27	0.22	7.44	7.80
750 ch Dolphins	1.06	0.66	36.1	30.3	0.24	0.19	8.00	8.30

Fig. 25. Effect of segmentation errors. The principal improvement in performance obtained when segmentation errors (#'s) are omitted is a reduction in the number of characters saved as prototypes, with a corresponding decrease in the average length of search. The substitution error rate does not decrease by the percentage of missegmented characters since most of these are normally saved as prototypes and do not constitute substitutions (exceptions are shown in Fig. 24). The runs appearing in this table show an unusually high number of segmentation errors; on most material these effects are negligible.

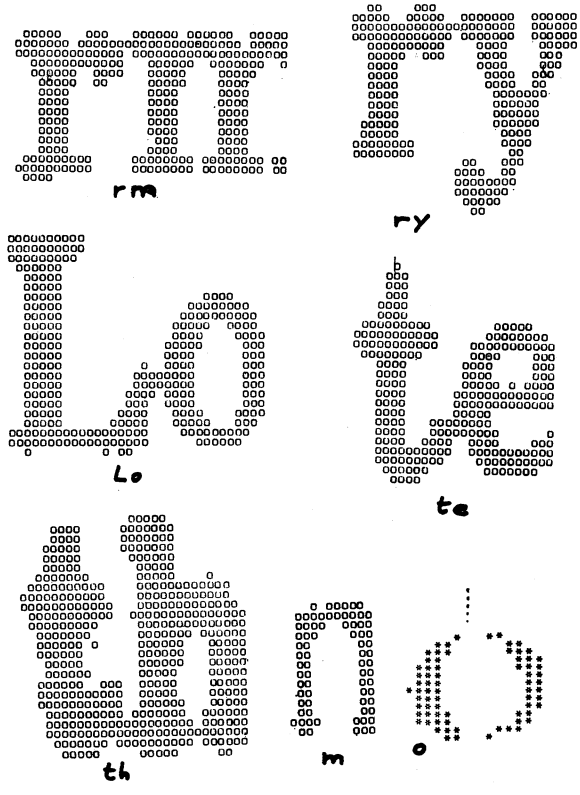


Fig. 26. Substitutions due to missegmentation. Each of these doublets was recognized by the system as the first number of the pair. Since the operator identified them as segmentation errors (#), they were included in the substitution count. This type of substitution error can be eliminated by keeping track of the relative width of the character and the prototype.

Material	Number of Characters	Average Length of Search	Fraction saved as prototypes	Substitution Errors (%)
Spectrum '70 Feb. p. 114	2819	54.3	0.14	0.31
ALR Vol. 91 p. 43	2374	58.6	0.17	0.17
Patent 2802067	2000	48.7	0.14	0.25

Fig. 27. Typical classification results with parameters set for normal operation. These conservatively selected figures are representative of the performance to be expected with the existing system without pre- or post-editing. Only material scanned with the scanner in proper adjustment, and for which a fairly large sample was available, is shown.

able. Of course, other systems also set aside unrecognizable characters for subsequent identification, but we do not know of any system other than ours which makes use of these "rejects" to classify other characters.

The usage-ordered search procedure is effective in reducing the number of comparisons to that currently performed by OCR machines designed for a much smaller symbol vocabulary.

The post-editing process is too slow in its present state to allow proofreading and correction of the complete file. The object was simply to demonstrate that any required degree of accuracy may be attained, without much complication, in critical portions of the data. In some applications, the 99.75 percent accuracy we have repeatedly obtained without post-editing may be acceptable for the bulk of the material.

Implementation

The amount of computation performed in all phases of the text reading operation is roughly comparable to the work expended in format control and recognition in conventional OCR machines. Hence there is no reason to believe that a hard-wired version of this system could not attain the 1000–2000 character/s rates current in OCR (albeit at a much higher error rate). We estimate that 50 to 300 characters/s, depending on the CPU, could be obtained with the logic performed in software, taking advantage of special instructions in READ-ONLY memories. At the top of this range it would be more economical to have some special-purpose hardware just for the bit comparisons than to increase the size of the CPU.

The most problematic component is the scanner itself. A satisfactory device should be able to cover an 8- by 10-in area with linearity within about 20 mil and sufficient amplitude uniformity to allow an effective spot size under 4 mil in diameter. To avoid having to process blank areas of the page the scanner must be completely under program control yet must be fast enough to keep up with the chosen processing rate. At the moment such a device is not available commercially (nor, for that matter, experimentally).

For field selection, improved versions of the data tablet are available at relatively low cost. For the display, a static high resolution device such as the storage tube, should be considered. This would also serve for prototype labeling.

Post-editing requires only a character display. It is not yet clear how many characters should be displayed simultaneously for most efficient proofreading and correcting.

The number of operators kept busy on field selection, prototype labeling, and post-editing is a function both of the overall processing rate and of the difficulty of the material. Only the field selection is linked closely to the actual movement of the documents; the other functions may be performed off-line or on time-shared terminals.

Problems Outstanding

In formatting the page, it would be desirable to incorporate in the system existing techniques for coding line drawings and for scanning and storing continuous-tone images. Methods used in automatic typesetting and photocomposition for scanning and interpreting formulas and equations should also be studied in this connection. Finally, simpler aspects of the overall format problem, such as locating page numbers, footnotes, and short headings, could be gradually programmed, thus lessening the burden on the operator.

In character acquisition, segmentation remains the principal problem. Increasing the resolution of the scanner will help considerably, but more sophisticated algorithms will nevertheless have to be developed to deal with characters which actually touch. Broken characters do not, as is the case in typescript, pose much of a problem.

In the classification phase, there are two obvious approaches available for decreasing the required number of comparisons. The first approach is to use a few simple features of the character, such as height, width, and symmetry, to limit the search to certain subgroups. The second approach is to use context in the form of n-gram probabilities to order the prototypes for the search.

We also expect to use context to detect, and perhaps correct, errors in the "recognized" text. A dictionary lookup procedure would complement the n-gram approach which, to some extent, will already have been "used up" earlier, but this is complicated by the possibility of letters missing due to missegmentation.

The classification may be more directly improved by the use of some averaging in the computation of prototypes, or by the use of a decision method of greater power than simple correlation. The latter may turn out to be actually easier since averaging must take into account the imperfect registration of the individual characters and the nature of the shifting process used in the bit comparisons. In either case a separate routine must be written for punctuation where translation-invariant methods are bound to fail.

In post-editing there are some obvious improvements to introduce, such as the use of both upper and lowercase characters in the display and better means of tagging and correcting characters. For instance, an all light pen or all keyboard operation may be faster than the particular combination utilized, and frequent resort to the function keys should also be abandoned. Furthermore, better use should be made of the computational facility available. Erroneous words could perhaps be found more quickly by the computer on the basis of several keyed-in characters or a superimposed coordinate grid than by pointing to them or moving a cursor through keyboard codes. Detailed studies of these matters must unfortunately be based on the particular cost criteria of individual applications, and consequently a versatile system for reading text would have to include a number of options for editing and error correction.

Finally, it will be necessary to investigate how to live with the residual errors. Since most of these consist of confusion pairs such as I-l, 0-O, or l-1, each pair could probably be represented by a single computer code in many text processing applications. Certainly the use of ambiguous display characters for such pairs would cause little hardship for human readers.

APPENDIX

SOURCE DOCUMENTS

[1] United States Patent Office, U. S. Patent 2 791 703, May 7, 1957.
[2] R. H. Jeppesen and H. L. Caswell, "Prenucleation of lead films with copper, gold, and silver," *IBM J. Res. Develop.*, p. 297, Oct. 1963.
[3] *American Law Reports.* Rochester, N. Y.: Lawyers Cooperative Publishing Company, 1953, p. 248.
[4] P. Lowber, *The Friendly Dolphins.* New York: Random House, 1963, p. 69.

United States Patent Office 2,791,703 Patented May 7, 1957

1

2,791,703

SEMI-CONDUCTOR DEVICES HAVING FOCUSING ELECTRODES

Jacques I. Pankoveheikoff, now by change of name Jacques Isaac Pankove, Princeton, N. J., assignor to Radio Corporation of America, a corporation of Delaware

Application May 1, 1951, Serial No. 223,905
7 Claims. (Cl. 307—88.5)

This invention relates generally to semi-conductor devices, and particularly relates to transistors having improved electrode configurations.

A transistor is a semi-conductor device having a semi-conducting body such as a crystal of silicon or germanium provided with three electrodes. One of the electrodes is in low-resistance contact with the crystal and is called the base electrode. The other two electrodes are in rectifying contact with the crystal and are called the emitter and the collector electrodes. A potential in the forward direction is impressed between emitter and base and a potential in the reverse direction is applied between collector and base. Such a transistor may be used, as is well known, in amplifier, modulator, oscillator or the like circuits.

During operation of a transistor, both the emitter and the collector electrodes inject charge carriers of opposite polarity into the crystal. If the crystal consists of an N type material, the emitter will be biased positively and the collector negatively with respect to the base. Under these conditions the emitter injects holes into the crystal which have a positive charge, while the collector injects electrons into the crystal. The holes behave somewhat like electrons having a positive charge.

The operation of a transistor is dependent to a large extent on the transit time of the holes and on their recombination. The transit time should be as short as possible to permit high frequency operation. The recombination of the holes with electrons should be as small as possible. It will, of course, be obvious that the recombination is proportional to the path length of the holes. It has further been found that recombination takes place faster at the surface of the crystal than in the interior. Furthermore, for efficient operation the barrier layer which is formed within the crystal or at its surface should be as thin as possible. It has been found that the thickness of the barrier layer decreases as the potential gradients within the crystal increase.

In accordance with the present invention, the electrodes of a transistor are disposed in accordance with the principles of electron optics to obtain optimum conditions for its performance. In this manner the gain as well as the efficiency of the transistor may be increased and the output impedance reduced. The conventional transistor has an input impedance of the order of 200 ohms and an output impedance of the order of 10,000 ohms. In view of the large difference between the input and output impedances of the transistor, matching of successive amplifier stages connected in cascade presents appreciable problems to the circuit designer. Therefore, a smaller ratio of output impedance to input impedance would be desirable.

It is therefore an object of the present invention to provide a transistor having electrodes of such a configuration that the gain is increased and that the required emitter current as well as the output impedance are reduced.

2

Another object of the invention is to provide an improved transistor having a ratio of output impedance to input impedance which is considerably reduced, thereby to facilitate matching of successive transistor amplifier stages.

A further object of the invention is to provide a transistor wherein the electrodes are so disposed that the charge carriers injected by the emitter are substantially focused onto the collector while the charge carriers injected by the collector are substantially collected by the base.

In accordance with the present invention the transistor comprises a semi-conducting body or crystal having a first substantially plane surface and a second surface which may also be plane and which is disposed opposite the first surface. The base electrode is in low-resistance contact with substantially the entire area of the first plane surface with the exception of a small surface portion within which the collector is disposed. The emitter electrode is in contact with the second surface substantially opposite the collector. When a voltage in the forward direction is applied between emitter and base and a voltage in the reverse direction between collector and base, the charge carriers injected into the crystal by the emitter are substantially focused upon the collector. On the other hand, the charge carriers injected into the crystal by the collector are defocused and, accordingly, the majority of these charge carriers is collected by the base. A transistor in accordance with the invention will have an improved gain and a lower emitter current, that is, an improved efficiency. This is, of course, due to the fact that substantially all charge carriers injected by the emitter are collected by the collector. Furthermore, the collector impedance or the ratio of the collector impedance to the emitter impedance is reduced.

The novel features that are considered characteristic of this invention are set forth with particularity in the appended claims. The invention itself, however, both as to its organization and method of operation, as well as additional objects and advantages thereof, will best be understood from the following description when read in connection with the accompanying drawing, in which:

Figure 1 is a sectional view illustrating schematically a transistor embodying the present invention; and

Figure 2 is a sectional view of a preferred embodiment of the transistor of the invention.

Referring now to the drawing wherein like components are designated by the same reference characters and particularly to Figure 1 there is illustrated a transistor having a semi-conducting body 10. Body 10 may consist of a crystal of silicon or preferably of germanium. The crystal may be of the P type or of the N type, but for the following discussion it will be assumed that it consists of the N type. The body 10 is provided with a first substantially plane surface 11 and with a second surface 12 disposed opposite the first surface 11. Surface 12 may also be substantially plane as illustrated. However, it is to be understood that surface 12 may also have concave shape and may, for example, consist of a spherical depression. Transistors having two faces with ground spherical depressions are well known. They are called coaxial transistors and have been described, for example, by J. N. Shive in "Physical Review," vol 75, 1949, p. 689, and by W. E. Kock and R. L. Wallace, Jr., in "Electrical Engineering," March 1949, pp. 222–223 (see also Patent 2,522,521 to Kock).

A metallic sheet or foil 14 is soldered as indicated at 15 to the first plane surface 11 of the crystal 10. The sheet or foil 14 is provided with a central aperture indicated at 16 so that a predetermined small surface portion of the crystal is not covered by the foil 14. A fine

[1]

R. H. Jeppesen*
H. L. Caswell

Prenucleation of Lead Films with Copper, Gold, and Silver

Abstract: Lead films evaporated onto thin nucleating layers of Cu, Ag, and Au were studied by the techniques of electron diffraction and electron microscopy. Electron micrographs indicated that films nucleated with Au become continuous much sooner than films nucleated with Ag or Cu. Examination of the diffraction pattern showed Au to be the only metal of the three to form an intermetallic compound with lead, indicating that compound formation aids in nucleation.

Introduction

The presence of minute quantities of selected surface impurities can dramatically affect the nucleation and growth characteristics of a thin film. For example, by using a nucleating layer of silver several monolayers in thickness, continuous zinc films have been deposited where the probability of the zinc sticking to the substrate would normally be zero.[1] Also a thin (~10 A) nucleating layer of copper has been successfully used in depositing fine-grained tin films on 90°C substrates. Normally these 5000 A films would consist of agglomerated islands of well oriented (200) crystallites. Other metals, e.g., silver and gold, can be used to produce a similar effect with tin.

At least two possible mechanisms may be responsible for the observations cited above. First, the presence of discrete silver and copper nuclei provides stronger nucleating sites for the arriving film atoms much in the same manner as a cleavage step on an NaCl surface will promote nucleation.[2] The increased binding energy in this case arises from electrical interactions of the nucleating material and arriving metal atoms. Since the density of these nuclei in very thin films of silver and gold is quite high, e.g., 5×10^{11} cm^2 for a 10 A Au film on NaCl, very fine grain films can be deposited. The second mechanism involves intermetallic compound formation, which further increases the binding energy of the film material to the nucleating metal by the heat of formation of the compound. Because the binding energy of a film atom to the substrate enters nucleation theory exponentially,[3] one might expect intermetallic compound formation to have a dramatic effect on the density of critical nuclei present during film growth. The objective of the investigation reported here was to determine the importance of intermetallic compound formation in the selection of prenucleating materials.

Experimental technique

As stated previously, silver, copper, and gold are all effective nucleating agents for tin. However, in bulk material they all form intermetallic compounds with tin.[4] In addition, the presence of Cu_6Sn_5 has been reported in layered structures formed by alternately evaporating thin films of copper and tin.[5] Therefore lead was chosen for the film material because the Pb-Ag and Pb-Cu systems reportedly have no appreciable solubility at either end of the phase diagram (eliminating any contribution to the binding energy from the heat of solution of alloying) and form no intermetallic compounds.[6] In the Pb-Au system there is again no appreciable solubility at room temperature, but the bulk phase diagram indicates the existence of Au_2Pb and $AuPb_2$. By preparing films of these materials and using electron diffraction and microscopy, it was possible to evaluate the importance of intermetallic compound formation in the prenucleation process.

Films were prepared by evaporating high purity metals

* Physics Department, Montana State University, Missoula, Montana.

[2]

[§ 46]

provements, its section men placed cinders on it, and when the rails were elevated placed cinders on the roadway to bring it up to the new level of the railroad, the court in Harriman v New York, C. & St. L. R. Co. (1930) 253 NY 398, 171 NE 686, affirmed judgments for an infant plaintiff's injuries and his father's expenses, where it appeared that the plaintiff had accompanied his uncle to the farm and after they departed and were en route to the highway, immediately after the truck passed over the railroad tracks, the left rear wheel cut into the soft cinders, the bank cut off, and the truck went over the embankment, as it was shown that other vehicles had done on several occasions. Declaring that the question of the defendant's negligence and of the plaintiff's freedom from negligence was properly submitted to the jury, and that there was sufficient evidence upon which the jury could find that the railroad constructed and maintained the roadway in an unsafe and negligent manner, the court rejected the defendant's argument that the plaintiff was a bare licensee upon the roadway and that defendant owed him only the duty not to wilfully injure him. Assuming that the defendant was under some obligation to construct the crossing, and pointing out that the defendant interfered with and changed natural conditions by constructing the embankment upon the land of its grantor and knew that the only way its grantor and other persons lawfully going and coming to and from the highway to the farm could proceed would be over the roadway which defendant undertook to construct, the court said that when the defendant undertook to carry out its obligation to construct and maintain a roadway, the law placed upon defendant the duty of performing the obligation with reasonable care and the measure of its duty was to construct and maintain the roadway so that it should be reasonably safe for travel in the ordinary way by those who were lawfully using the roadway. Declaring that an actor's obligation to exercise due care in the performance of an obligation owed to another may inure to a third person under certain circumstances, the court concluded that the jury was justified in finding that if the defendant had exercised the care of an ordinarily prudent person, it would have apprehended that the probable result of its negligent construction and maintenance of the roadway would be injury to persons lawfully using it.

In Felton v Midland Continental R. Co. (1915) 32 ND 223, 155 NW 23, plaintiff was injured and his automobile was damaged when an earthen approach to a crossing gave way beneath the automobile. A statute required the railroad company to construct and maintain a "good and sufficient" crossing, with the grade of the approach being "of such slope as shall be necessary for the safety and convenience" of travelers and no less than 20 feet wide, and the complaint charged the railroad with negligence in failing to comply with the statute in that the approach was not 20 feet wide, the slope on either side in the approach was not gradual and convenient but steep and difficult of and the crossing was so constructed of firm and solid materials that would bear throughout its passage a vehicle in ordinary use, for purposes of travel, and the crossing was so constructed and grade so steep that persons approaching on one side were not sufficiently elevated when entering upon the approach to see over the embankment and know whether there were other vehicles approaching on the other side. The question of defendant's negligence was an issue under the pleadings, but the case was tried upon the theory that the amount of plaintiff's damages, if any, and plaintiff's contributory negligence, constituted the only issues. The court affirmed a judgment for plaintiff and decided that defendant was not entitled to a directed verdict on the sole ground that plaintiff's damages were caused by his own carelessness and negligence. The court noted that the evidence showed that the approaches to the track rose about 10 feet and were not 20 feet wide but were only 17 feet wide up near the track and the

[3]

The longest study was made by Winthrop N. Kellogg.

Dr. Kellogg knew about bats and sonar. He wondered if dolphins could also sense echoes. He decided to find out. He chose to study bottle-nosed dolphins. Bottle-noses live chiefly along coasts. There the water is less clear than in the middle of the ocean. And there are more objects a dolphin must avoid. So the bottle-nose would need the best sonar—if dolphins had sonar.

Dr. Kellogg started with the rapid clicking noises that dolphins make. Were they like sonar signals? Could they be used to locate objects? The answer turned out to be yes.

Dr. Kellogg set up a laboratory in an outdoor pool. In the pool he placed two dolphins from Marineland. They were called Albert and Betty. (Albert turned out to be the star performer.) Then Dr. Kellogg began experiments that went on for six years.

There is not space to tell of all Dr. Kellogg's work. But here are six of his discoveries.

(1) He found that the dolphins kept close watch on their pool. Every 15 to 20 seconds,

[4]

ACKNOWLEDGMENT

The idea of using clustering to circumvent the difficulty of preparing an adequate training set was suggested to us by Dr. R. Casey. For advice and help in running the experiments we are grateful to J. Duffy, B. Farwell, Dr. W. Fitzgerald, Dr. N. Herbst, C. Marr, and to Mrs. V. Zeph, our "operator."

REFERENCES

[1] M. E. Stevens, Ed., *Pattern Recognition (Special Issue on Optical Character Recognition)*, vol. 2, no. 3, Sept. 1970.

[2] G. Nagy, "Current problems in character recognition," in *Proc. Purdue Symp. Inform. Processing*, pp. 369–378, Apr. 1969.

[3] A. Ferrari, "A versatile system for the automatic reading of text typographically printed," in *Mechanized Information Storage, Retrieval and Dissemination.* Amsterdam: North Holland, 1968.

[4] J. W. Schwartz, R. D. Turner, and P. Vlahos, "Application of print readers to the needs of the intelligence agencies—A preliminary survey of present machine capability," Study S-147 I.D.A., Research and Engineering Support, Mar. 1964.

[5] J. A. Fitzmaurice, "Reading Russian scientific literature," in *Optical Character Recognition.* Washington, D. C.: Spartan, 1962.

[6] R. B. Greenly *et al.*, "Universal print reader techniques," Link and General Precision Corp., Rep. AD 623613, 1964.

[7] M. Blitz *et al.*, "Implementation of document format recognition," Sylvania Corp., Rep. AD 803633, Oct. 1966.

[8] R. B. Thomas and M. Kassler, "Advanced recognition techniques study," Defense Documentation Cen., RCA Corp., Rep. 417814, Dec. 1963.

[9] M. Weiss, "Design considerations for a multifont OCR machine scanning microfilm images," in *Proc. Electro-Optical Syst. Design Conf.* (New York, N. Y.), Sept. 1969.

[10] F. F. Lee, S. J. Mason, and D. E. Troxel, "An experimental reading machine for the blind," in *Proc. Int. Conf. Pattern Recognition* (L.E.T.I., Grenoble), pp. 269–285, Sept. 1968.

[11] J. W. Biglow, E. S. McVey, and J. W. Moore, "Pattern recognition in a learning automatic control system," in *IFAC Symp. Rec. Pt. I*, Sec. 1.5, pp. 1–7, June 1967.

[12] R. M. Bowman and E. S. McVey, "A method for the optimal design of a class of pattern recognition system," *Pattern Recognition*, vol. 2, pp. 187–198, Sept. 1970.

[13] R. B. Hennis *et al.*, "The IBM 1975 optical page reader," *IBM J. Res. Develop.*, vol. 12, pp. 345–371, Sept. 1968.

[14] L. P. Horowitz and G. L. Shelton, Jr., "Pattern recognition using autocorrelation," *Proc. IRE*, vol. 49, pp. 175–185, Jan. 1961.

[15] J. Raviv, "Decision making in Markov chains applied to the problem of pattern recognition," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 536–551, Oct. 1967.

[16] C. N. Liu and G. L. Shelton, Jr., "An experimental investigation of a mixed-font print recognition system," *IEEE Trans. Electron. Comput.*, vol. EC-15, pp. 916–925, Dec. 1966.

[17] R. G. Casey and G. Nagy, "An autonomous reading machine," *IEEE Trans. Comput.*, vol. C-17, pp. 492–503, May 1968.

[18] G. Nagy, "Preliminary investigation of techniques for automated reading of unformatted text," *Commun. Ass. Comput. Mach.*, vol. 11, pp. 480–487, July 1968.

[19] R. N. Ascher and G. Nagy, "An integrated man-machine system for reading printed text," *Proc. NEC*, vol. 28, pp. 486–489, Dec. 1970.

[20] R. N. Ascher, G. M. Koppelman, M. J. Miller, G. Nagy, and G. L. Shelton, Jr., "A production oriented system for reading text," IBM Rep. RC 3023, Yorktown Heights, N. Y., 1970.

[21] N. M. Herbst and P. M. Will, "Design of an experimental laboratory for pattern recognition and signal processing," in *Proc. Brunel Symp. Comput. Graphics*, Apr. 1970; also, IBM Rep. RC 2619, Yorktown Heights, N. Y., Sept. 1969.

[22] N. H. Kreitzer, "Buffer store for an 1800 computer and experimental magnetic recording system," IBM Rep. RC 2362, Yorktown Heights, N. Y., Feb. 1969.

[23] D. O. Clayden, N. B. Clowes, and S. R. Parks, "Letter recognition and segmentation of running text," *Inform. Contr.*, vol. 9, pp. 246–264, June 1966.

[24] R. L. Hoffman and J. W. McCullough, "Segmentation methods for recognition of machine printed characters," *IBM J. Res. Develop.*, vol. 15, pp. 153–165, Mar. 1971.