which shows the compiled program of Fig. 3 together with the new version which requires only four (instead of 14) words for intermediate results.

The principles of the minimization algorithm are as follows:

1) A correspondence table is set up between "old" starred integers as they appear in the instructions of the first stage and the successive integers 1*, 2*, 3*, etc;

2) Starting with the second stage instructions, each stored operand symbol in the leftmost two address fields is looked up in the correspondence table and replaced by the value found there. The "old" value label is then dropped from the table entry, thus placing the entry on the "available" list;

3) The result location is taken as the first entry on the "available" list; it is labeled with the starred integer from the instruction "result" field (rightmost address field) of the instruction under consideration.

At the conclusion of compilation, all storage for intermediate results will appear as "available" with the exception of one word, which is assigned to the final result of the computation.

References

[1] H. Hellerman, "System organization for detection and execution of parallel operations in algebraic statements," IBM Research Note NC-164, October 31, 1962.
[2] D. L. Slotnick, W. C. Borek, R. C. McReynolds, "The SOLOMON computer," in *1962 Proc. Fall J.C.C.*, vol. 22. Baltimore, Md.: Spartan Books, 1963, pp. 97–107.
[3] H. Hellerman, "Experimental personalized array translator system," *Comm. ACM*, vol. 7, pp. 433–438, July 1964.
[4] P. Dreyfus, "Programming design features of the GAMMA 60 computer," in *1958 Proc. E.J.C.C.* New York: AIEE, 1959, pp. 174–181.
[5] C. L. Hamblin, "Translation to and from Polish notation," *Computer Journal*, vol. 5, pp. 210–213, October 1962.
[6] K. E. Iverson, *A Programming Language*. New York: Wiley, 1962.
[7] M. E. Conway, "A multiprocessor system design," in *1963 Proc. Fall J.C.C.*, vol. 24. Baltimore, Md.: Spartan Books, 1963, pp. 139–146.
[8] R. N. Thompson and J. A. Wilkinson, "The D-825 automatic operating and scheduling program," in *1963 Proc. Spring J.C.C.*, vol. 23. Baltimore, Md.: Spartan Books, 1963, pp. 41–49.
[9] D. J. Howarth, P. D. Jones, and M. T. Wyld, "Experience with the ATLAS scheduling system," in *1963 Proc. Spring J.C.C.*, vol. 23. Baltimore, Md.: Spartan Books, 1963, pp. 59–67.
[10] J. S. Squire and S. M. Palais, "Programming and design considerations of a highly parallel computer," in *1963 Proc. Spring J.C.C.*, vol. 23. Baltimore, Md.: Spartan Books, 1963, pp. 359–400.
[11] A. J. Critchlow, "Generalized multiprocessing and multiprogramming systems," in *1963 Proc. Fall J.C.C.*, vol. 24. Baltimore, Md.: Spartan Books, 1963, pp. 107–126.
[12] R. R. Seeber and A. B. Lindquist, "Associative logic for highly parallel systems," in *1963 Proc. Fall J.C.C.*, vol. 24, Baltimore, Md.: Spartan Books, 1963, pp. 489–493.
[13] R. W. Allard, K. A. Wolf, and R. A. Zemlin, "Some effects of the 6600 computer on language structures," *Comm. ACM*, vol. 7, pp. 112–119, February 1964.

Recognition of Printed Chinese Characters

R. CASEY AND G. NAGY

Abstract—**The problem of recognizing a large alphabet (1000 different characters) is approached using a two stage process.**

In the first stage of design, the data is partitioned into groups of similar characters by means of heuristic and iterative algorithms. In the second stage, peephole templates are generated for each character in such a way as to guarantee discrimination against other characters in the same similarity class.

Recognition is preceded by establishing an order of search through the groups with a relatively small number of "group masks." The character is then identified by means of the "individual masks." through a threshold criterion.

The effects on the error and reject rates of varying the several parameters in the design and test procedure are described on the basis of computer simulation experiments on a 20 000 character data set. An error rate of 1 percent with 7 percent rejects, is obtained on new data.

Manuscript received August 13, 1965; revised October 24, 1965.
The authors are with the IBM Watson Research Center, Yorktown Heights, N. Y.

I. Introduction

A. Need for Chinese Character Recognition

THE RAPID EMERGENCE of China as one of the leading producers of publications has fairly swamped United States translators monitoring Chinese activity. In 1962 the level of Chinese to English translation was estimated at 3.5 million words per year, in contrast to the estimated need of 34.4 million words per year by the intelligence community alone [7]. This requirement is expected to grow at the rate of about 25 percent per year.

Spurred by such statistics, numerous serious machine translation groups have devoted considerable attention to Chinese syntax and lexicography. The restrictions imposed by the peculiar structure of Chinese do not appear to be critical; in fact, much of the tech-

nique and hardware developed for other languages, particularly Russian, is applicable [6]. The translation effort has been severely handicapped, however, by the difficulty of coding large amounts of Chinese material in a form suitable for computer processing. The best coding device to date, the manually operated sinotype machine [6], processes only about 12 characters per minute, a rather low rate compared to keypunching operations in the Western languages. Thus an automated system for encoding Chinese text may be even more appealing—and urgent—than its counterpart in English or Russian.

The very real need for machine recognition of printed Chinese has formed, however, only part of the motivation for the study reported here; the experience derived from working with a large pattern vocabulary was thought to be worthwhile in itself. In addition, it was hoped that useful techniques would be accumulated for certain closely related applications, such as searching a large file of Chinese documents for particular significant groups of ideographs (references to rockets, for example) and recognition of Japanese symbols.

B. Chinese Ideographs

Chinese ideographs usually correspond to complete words in Western languages, although in many instances a whole group of ideographs may be translated by a single word in English. An example of Chinese print is shown in Fig. 1.

七 成 選

Fig. 1. Chinese ideographs. This font style is commonly used in Taiwan, Hong Kong, and the U.S.A. Simplified forms have been gaining acceptance in mainland China.

Most common ideographs are derived from a pictorial representation of objects. Complex ideas are described by groupings of patterns, usually within an imaginary rectangular frame.

Subpatterns common to many ideographs are called radicals. Radicals may convey either a meaning or a sound. Scholars recognize only 214 distinct radicals, but many of these have so-called "variant" forms which bear little resemblance to the "main" forms [9]. The size and position of a radical varies according to the complexity and structure of the remainder of the ideograph.

Characters and radicals may be further analyzed in terms of "strokes." The term serves as a reminder that for millenniums all Chinese was brushed, but aside from this it defies precise definition.

Radicals contain from one to seventeen strokes, while even common characters may run up to 27 strokes. This renders learning and writing Chinese script a formidable task, and imposes severe problems even in printing. Simplified characters containing fewer strokes were introduced as early as 2000 years ago. The process has been greatly accelerated in the last ten years; in mainland China several hundred simplified characters have already replaced their complex counterparts [5]. There is also a sustained drive towards latinization, but the introduction of a phonetic alphabet is hampered by the fact that the same symbols are often pronounced in completely different fashion in the various regions.

The size of the "alphabet," and the difficulties encountered in assigning identities to individual characters, preclude the widespread use of typewriters and coding devices (such as the flexowriter) in Chinese. Perhaps this is why character frequency counts, and especially digram and trigram probabilities, are virtually nonexistent. There is very little agreement as to how many characters would constitute a useful complement for a reading machine. Estimates range from 2000 to 5000 characters [4], [6]. About all that can be said for certain is that the more, the better.

Figure 2 shows a character frequency count prepared for a survey of teaching methods in elementary schools [3]. The study from which this count was taken is quite a thorough one, including even frequency counts for radicals; a similar count on scientific matter would be most helpful.

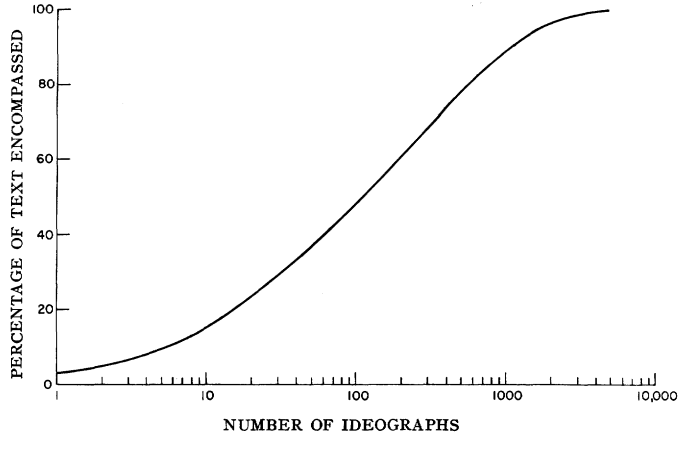


Fig. 2. Cumulative frequency of Chinese characters. Curve based on character count comprising 902 678 occurrences by H. C. Chen.

The one feature which raises hopes of constructing a workable Chinese reading machine is the uniformity of printed ideographs. Standardization of font styles has progressed furthest in mainland China, but few printers anywhere would willingly store many styles of a 7000 to 8000 character type case.

Printed lines run horizontally in mainland China, and vertically almost everywhere else. The size commonly used in scientific journals runs eight characters to the inch. The type slugs are perfectly square; the spacing is thus fixed, while the intercharacter separation depends only on the complexity of the adjacent characters. Minimum separation with clean type is of the order of one fifteenth of a character width.

Detail is richer in the vertical sense, with a maximum of nine horizontal black lines above one another in a

character. Line width is greatly variable, in imitation of the old brush strokes. On the average about half of the character is composed of black areas.

C. Data Set Used in Simulation Experiments

For large scale experiments an ordered (or labeled) set of test characters is essential. Accordingly, a Chinese printer in New York was asked to prepare a 20 000 character data set in the format shown in Fig. 3. All the characters on a given line are identical, and each of the 1000 lines represents a different ideograph. Ten samples of each ideograph were used to "train" the recognition system; the remainder was reserved for testing.

The ordering of the lines is arbitrary; the printer was merely asked to include one thousand of the most commonly used ideographs. Both new and old slugs were used in hand set frames with standard spacing. Letterpress impressions were produced on glossy Kodak paper.

This material was photographed on 35 mm film, ten lines to a frame, and scanned with the cathode ray tube transparency scanner described by Potter [8]. During scanning each line was automatically labeled with its sequence number; this "line number" thus became the identity of all twenty characters in the line.

The resolution of the scanner was set to yield about 23 bits for both the length and width of a character. The character separation circuits were adjusted to require four blank vertical scans for separation before the twentieth scan, one blank scan between the twentieth and the twenty-fifth, and to truncate the character unconditionally at the twenty-fifth scan. Small blobs of black outside the larger configuration were also excluded. These precautions are sufficient to keep character mutilation at about the 0.2 percent level.

The scanned characters are stored on magnetic tape with each 25 word record of video tagged by an identification word.

Fig. 3. Part of the data design set for recognition experiments. The left-hand side of the page was used for design, the right-hand side for testing.

D. The Strategy

Two considerations guided the overall approach to the problem. First, the recognition process to be selected must be capable of extension to single font alphabets larger than the 1000 symbol set used in the initial study. Second, an efficient form of implementation must be available within the present state of the art.

Perhaps the simplest method compatible with these requirements is that of peephole templates, or masks. There is one mask associated with each character; the decision procedure consists only of determining whether any of the masks afford a sufficiently good fit for an unknown sample. This scheme is readily adaptable, by a change of threshold, to the identification of key ideographs in extensive files of Chinese print, where the penalty for false negatives is much greater than for false positives.

More complicated methods of pattern recognition, including character tracing and the identification of topological features, have been successfully used on printed characters [10]. Because of the single font nature of this application, a recognition system based on templates might reasonably be expected to compete with more sophisticated techniques.

If the character is to be described by a 25×25 binary matrix, a mask might typically consist of 40 judiciously chosen points. Figures 4 and 11 show the binary representation and the masks for some of the characters illustrated earlier. Registration invariance is achieved by moving the mask about on the pattern.

Peephole templates are easy to generate automatically, even for large alphabets, since only the character for which the mask is generated, plus a few similar characters, need be considered. The relative sparseness of points in the masks contributes greatly to both ease of implementation and simulation; therefore, the gener-

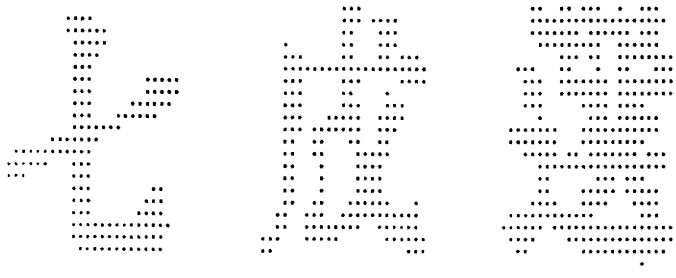


Fig. 4. Scanned characters. These patterns are quantized versions of the ideographs shown in Fig. 1.

ation procedure attempts to minimize the number of points as well as to insure maximum discrimination against similar characters. The precise manner, thought to be novel, in which the set of 40 or so points are selected among the many $({}^{625}C_{40} \cdot 2^{40})$ possibilities will be described in Section III.

It is easy to see that if the masks are tested in random order on each new sample, then for an n-symbol alphabet $n/2$ tests will be required on the average. To reduce the number of tests, a two-level search was instituted.

The 1000 individual masks which have already been described are partitioned into 58 groups. Each group, containing masks for a number of similar characters, is represented by a single group mask. In the first stage, an unknown character is compared to all of the group masks, and a preferred order of search through the groups is defined by the mismatch scores. In the second stage, the unknown is compared in this order to the individual masks until a sufficiently good match is found.

The two levels of operation are essentially the same as far as hardware and simulation are concerned, although the group masks are derived differently from the individual masks. Membership in the groups is established once and for all during design; the clustering procedures necessary to effect this classification are described in Section II.

In addition to increasing machine throughput by decreasing the average number of mask matching operations, clustering also facilitates the design of individual masks by focusing attention on likely confusion sets. This information is used to advantage by the mask generator.

It is estimated that character rates of the order of 3400 characters per minute may be attained with the proposed scheme. One possible form of implementation, based on an existing scanner and making use of standard digital techniques and a modest amount of analog hardware, will be described in Section IV.

In Section V the relationships between processing time, error, and reject rates are explored by varying a number of parameters in both the design and test programs. The causes of errors and rejects are analyzed; this breakdown itself suggests the improvements necessary to increase the rate of correct identification.

II. Preliminary Classification

A. Clustering

The order of search established by the first stage is obviously a critical factor in quickly locating the matching template for an unknown sample. What is desired is a set of first-stage masks, each designed to detect the members of a small subset of the 1000 character alphabet. The task of forming appropriate subsets, called "clustering," is a well-known operation in statistics, and a number of methods are available.

The four different clustering routines described below were evaluated by a means of a special test program. Several other more sophisticated schemes could not be tried because, in the present application, they required excessive computer storage and processing time. A "similarity matrix," for example, used in graph theoretic approaches to clustering [1], would have had a half-million entries.

B. Methods

1) *Uniform Distribution Hypothesis*

The wide variations present in the 1000 Chinese characters may lead one to hypothesize that in effect the characters consist of random assemblages of black areas. An efficient set of mask points in this case should also be uniformly distributed over the character field.

To test this line of reasoning 64 templates were constructed by randomly choosing 50 black points for each.

In order to assure that meaningful points were selected, they were constrained to be in an 18 by 18 raster area. Groups were formed by assigning each character to the mask giving the best fit for the majority of the samples of that character.

2) *Pairwise Correlation Chain (Typical Characters)*

Bonner [2] has used clustering techniques in the course of generating Boolean logic for character recognition. His program for partitioning a set of binary samples, based on forming a list of "typical" characters, is as follows.[1] An arbitrary integer τ is specified. The first sample is taken as a prototype for a group. The second sample is then compared with the first. If they differ in no more than τ bit positions, then the second sample is assigned to Group One. If the difference exceeds τ, then the second sample is chosen as the prototype of Group Two. The third sample is compared with both prototypes and either starts a new group or is assigned to one of the existing groups. The program continues in this manner until the list of samples is exhausted. The number of groups formed is a decreasing function of parameter τ.

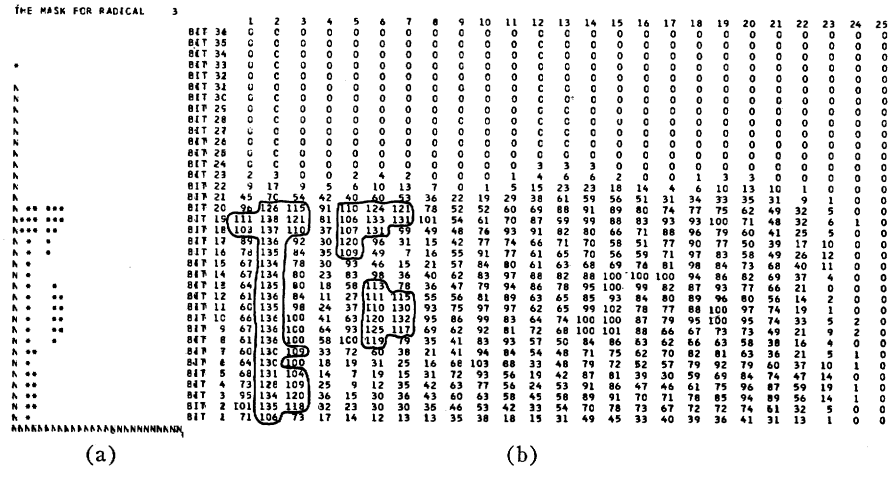
To apply this algorithm to Chinese characters, a sample of each character was registered tangent to the

[1] Dr. Bonner kindly gave us a copy of his clustering program for use on Chinese characters.

降 陷 限 院 阿

隱 隨 陳 隔 陣

Fig. 5. Characters having the same radical. The radical appears on the left-hand side of the character.



(a) (b)

Fig. 6. (a) Example of a radical mask. (b) Count matrix for radical mask. The fifty points chosen for the mask are outlined.

left and bottom margin of the raster field. These samples were fed to the program in arbitrary order. Parameter τ was varied until 64 groups were generated.

3) *Radicals*

The characters possessing a common radical form a subgroup defined in a natural way. A mask designed to detect the radical would seem to be a reliable indicator of the presence of a member of the subgroup.

To obtain such a template, a list of characters containing the same variant of a radical was formed. Ten registered samples of each character contributed to a bit count for each of the 625 raster bit positions. With 30 characters in the list, for example, this bit count could range from zero to 300. The radical mask was defined to consist of 50 black points corresponding to the 50 bit positions having the highest counts. Figure 5 shows a number of sample characters of the same radical class. Figure 6 illustrates the sample matrix of bit counts for that radical and the resulting mask. In this manner 64 radical masks were generated, associated with groups of from 5 to 60 characters. Only 70 percent of the alphabet could be assigned to the lists on the basis of the approximate geometrical congruence of given variant forms. The remaining 300 characters were not amenable to grouping in this manner.

4) *Minimization of the Average Distance*

As a fourth approach to clustering, a distance measure was defined to represent the goodness of matching between a mask and a character. The masks were then varied until the average within-group distance was minimized.

To minimize storage requirements in programming the method, inputs were derived as follows. A number of samples of each character were averaged, and this average was quantized into three levels. The resulting "skeleton" character, described more fully in Section III, is a representation of the white, black, and uncertain regions of the character.

In the algorithm used to generate the masks,[2] the bit positions of the character field are numbered sequentially from 1 to 625. Suppose there are N sample characters (or skeleton characters) which are to be resolved into k clusters. Let the state of bit position r in the ith character be represented by a number s_{ir}, which is 0, $\frac{1}{2}$, or 1 if that bit position is white, uncertain, or black, respectively. In the same manner let the rth bit position in the jth mask be represented by a number x_{jr}. A measure of the similarity between the ith character and the jth mask is the function

$$d_{ij} = \left\{ \sum_{r=1}^{625} (s_{ir} - x_{jr})^2 \right\}^{1/2}.$$

A cluster is defined here as the set of characters which are closest to a given mask. Let I_j be the set of indices of characters which are closer to mask j than to any other mask. A measure of the clustering performance of an arbitrary set of masks is the mean square distance

$$D^2 = \frac{1}{N} \sum_{j=1}^{k} \sum_{i \in I_j} d_{ij}^2 = \frac{1}{N} \sum_{r=1}^{625} \sum_{j=1}^{k} \sum_{i \in I_j} (s_{ir} - x_{jr})^2.$$

[2] Dr. R. Bakis of IBM Research Division suggested the original algorithm, which was modified for the ternary case described here.

Starting with a given set of masks the mean square distance can be reduced by choosing each x_{jr} to minimize the sum

$$\sum_{j=1}^{k} \sum_{i \in I_j} (s_{ir} - x_{jr})^2.$$

Forming new lists $I_1, I_2, \cdots, I_k$, based on the altered masks, gives an even lower value of D^2. The masks are then changed again to minimize D^2 for the new cluster assignments. Alternate recomputing of masks and determining of cluster assignments is carried out until a stationary set of masks is attained. The distance is decreased in both steps of the cycle; hence, a local minimum at least will be achieved.

A 7094 computer program implementing the ternary algorithm was supplied with skeletons of the 300 or so characters which have no usable radicals. The first 25 skeletons in the list were chosen as initial masks. After twelve iterations the masks were changing very slowly and the program was stopped.

Because the radical masks were deemed adequate for grouping most of the characters, the very considerable expenditure of computer time required to try out the minimum distance algorithm on all 1000 characters was not felt to be justifiable.

C. Evaluation of the Methods

Sets of group masks for Chinese characters were obtained with each of the preceding procedures. The clustering methods were then ranked according to the average number of incorrect groups searched. To obtain this quantity experimentally, 64 groups were generated by each technique, and the first stage decision procedure was simulated for an input of 600 sample characters. Since only 25 masks were supplied by the distance minimizing algorithm, a number of the radical masks were included to make up the fourth set of 64.

Table I lists the results of this experiment. By a considerable margin, the combination of radical and minimum distance masks gives the search procedure which most quickly arrives at the correct group. This set of group masks was therefore carried over into the recognition experiments.

TABLE I
CLUSTERING PERFORMANCES

Method	Av. No. of Incorrect Groups
Random masks	3.28
Typical characters	2.07
Radical masks	0.92
Radical and minimum Distance masks	0.77

III. INDIVIDUAL MASKS

A. Objectives

In designing a mask for an individual character, three objectives are paramount:

1) The mask must fit the design character;
2) A certain minimum mismatch level must be maintained against all other characters;
3) The number of points in the mask should be kept as low as possible.

The design program described below is a logical approach to achieving these conditions by "building up" the mask. The scheme has been found to generate good masks very quickly. Obviously, when 1000 masks are to be made the amount of design time allowed each mask is small. Several heuristic devices contribute to efficient operation. Masks are constructed from compressed versions of sample characters called "skeletons." Initial masks are specified simply, points being added by the design program only if necessary. Finally, masks are designed to discriminate only against the other characters in their own similarity group. This step will not incur additional recognition loss if the first stage is effective in determining the correct group; it is assumed that characters easily confused will fall in the same group.

To counter registration errors during recognition, a mask is tried out in several locally shifted positions on the scanned character. The best match is accepted as the score for that mask. These shifts must also be taken into account during design; sufficient mismatch of the mask against alien characters or "impostors" must be secured in every shift position. For the Chinese recognition problem nine shift positions, corresponding to the vertical and/or horizontal translations of the character by one bit position, are used.

B. Skeleton Characters

When the binary version of a printed character is registered tangent to the bottom and left-hand margins of a rectangular field, some points are found to be black for most characters in the same class, while other points are usually white. There is also a region of uncertainty, distributed mainly along the borders of the strokes. Since the mask designing procedure requires knowledge of the stable points, i.e., those which are reliably black or reliably white for a given character, so called "skeleton" characters are derived through a computer program[3] as follows.

Ten samples of each character are read from tape and registered in the field. The number of black points in each bit position is counted. If this count is nine or ten, the point is designated a "stable black point," and if it is zero or one, the point is called "stable white." A point which fails to satisfy either criterion is labeled "unstable."

Examples of the resulting skeletons are shown in Fig. 7. In general there are enough stable points to identify a character from its skeleton. The distribution of the number of stable points in a character is shown in Fig. 8. The skeletons average about 200 stable points each.

[3] The program for deriving skeletons was originated by George Hu of IBM Research.

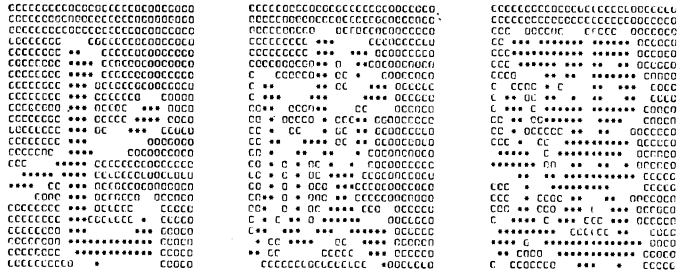


Fig. 7. Skeleton characters. Asterisks show stable black points, zeroes show stable white points. Same characters as in Fig. 4.

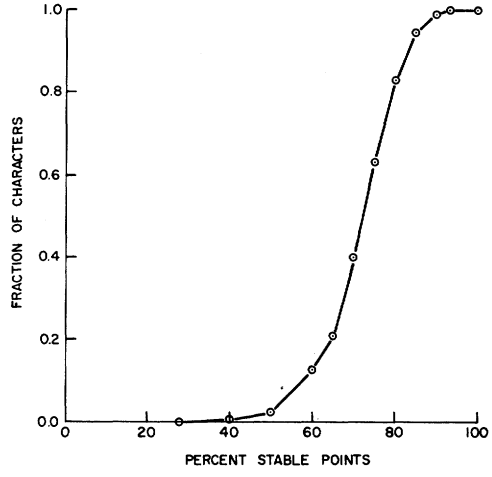


Fig. 8. Cumulative distribution of the number of stable points in a skeleton. The point count is obtained in a rectangle defined by the outermost stable black points in a skeleton in order to remove the effect of character size variation.

C. Design of Masks

Samples and skeletons of characters belonging to a single cluster constitute the input to the mask designing program. A mask for a given character is composed of stable points selected from the skeleton of that character. Initially thirty stable points (15 black and 15 white) are chosen at random from this skeleton, e.g., Fig. 9. The initial mask is then tried out on samples of the other members of the cluster. If sufficient mismatch is achieved against every sample, the mask is accepted as it stands. Otherwise, the characters which fail to give the required mismatch are listed as impostors, and a second design loop begins. The minimum mismatch is a program parameter.

In this second loop, additional skeleton points are transferred to the mask; however, the selection is no longer done randomly. Instead it is based on a mismatch table of the form shown in Fig. 10. The headings on the left are the various shift positions of each impostor while at the top are the 625 (25 by 25) bit positions. The top row of the table contains the skeleton from which the mask is to be selected. In the rest of the table an entry is "1" if there is a mismatch between the skeleton of the impostor and the top row entry. If either skeleton has an "unstable" point at this bit position, or if there is a match, the entry is "0."

Mask points are picked one at a time corresponding

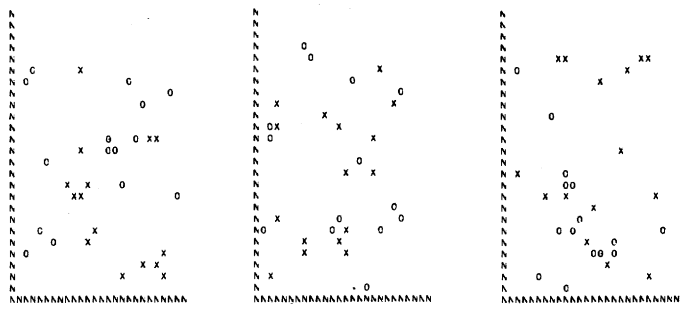


Fig. 9. Initial masks. These masks consist of thirty points picked at random from the skeleton. X's represent black points, 0's white points.

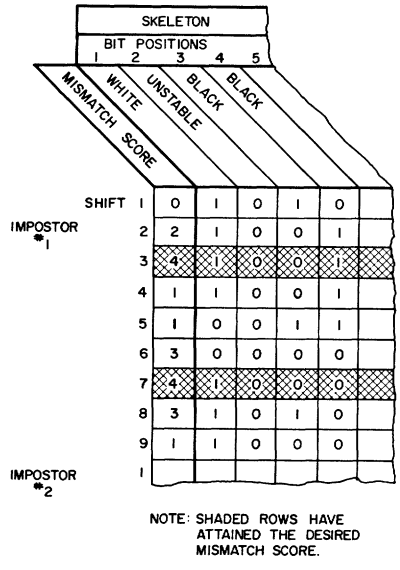


Fig. 10. Mismatch table for design of masks.

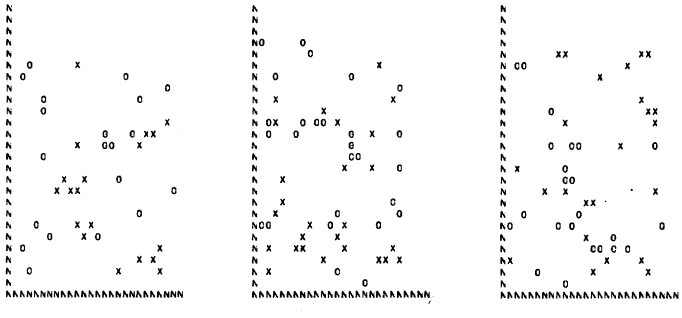


Fig. 11. Individual masks. X's are black mask points, 0's are white masks points. Size of masks ranges from 30 to 60 points. Same characters as before.

to the column in the table having the greatest number of "1's." As the mask builds up, a count is kept of the number of mismatches for each impostor shift position (the left-hand column of Fig. 10), and the corresponding row is dropped from the table when this count reaches the prescribed threshold. The design ends when no more rows are left or when the mismatch counts for the remaining rows can no longer be increased by adding mask points.

The program now goes into a brief testing phase, applying the mask just derived to three samples of each

of the characters in the group. If a character not on the previous impostor list fails to give the required number of mismatches with the augmented mask, it is added to a new list of impostors, and the design cycle begins again. Two cycles are usually sufficient to complete the mask. Examples of completed masks are shown in Fig. 11.

The mask obtained by this procedure has the following properties:

1) It has a high probability of matching the design character in at least one shift position, since only stable points are chosen for the mask.
2) It has achieved the desired mismatch level against as many impostors as possible.
3) Points in the unstable region of the remaining impostors are likely to contribute to the mismatch score when the mask is tried on sample characters.

Without some kind of packing scheme the 1000 individual masks could not be stored in a 32 000 word 7094 memory, and lower speed tape storage would have to be employed. The scheme used in this experiment was to store indexes denoting the number of bits (in scanning order) between successive points of the mask. The first word of a mask record contains the identification of the character as well as the number of black points and the number of white points in the template. Then follow a list of indexes giving the distances between successive black points, followed by a similar list for the white points. Using this procedure 20 000 36-bit memory locations were sufficient to store the 1000 masks. Unpacking from core storage the masks needed for each character is an order of magnitude faster than reading unpacked masks from tape.

IV. Implementation

In order to demonstrate the plausibility of building a fairly efficient Chinese character reading machine along the lines suggested, a possible mode of machine organization will now be described.

For concreteness, it is assumed that 100 groups will be sufficient for the partitioning of a 4000 character alphabet. Thus, the order of search through the 4000 individual masks is established by comparing the input character to each of the 100 group masks.

A schematic diagram of the proposed system is shown in Fig. 12. Shift register A, which contains the video bits, takes care of vertical registration, while shift registers B and C, containing the (ternary) masks, allow horizontal registration. Analog summing circuitry allows instantaneous computation of the mismatch scores.

Since only nine shift register cycles are sufficient to try a mask on a character in a 3 by 3 region, it is evident that the throughput of the system is limited mainly by the time required to transfer the masks into shift registers B and C. With a basic clock rate of one megacycle, and an average of 400 individual masks (plus 100 group masks) tested for each character, a rate of 3400 characters per minute appears attainable. It is assumed that all the masks are contained in a one megabit core store, and that a length-of-zero-run decoder unpacks each mask in 25 microseconds. With only buffered disk storage available for the individual masks, the throughput would drop to 520 characters per minute.

If the character reader were used in conjunction with a translation project, all the required bulk storage would be already available as part of the system. Nor

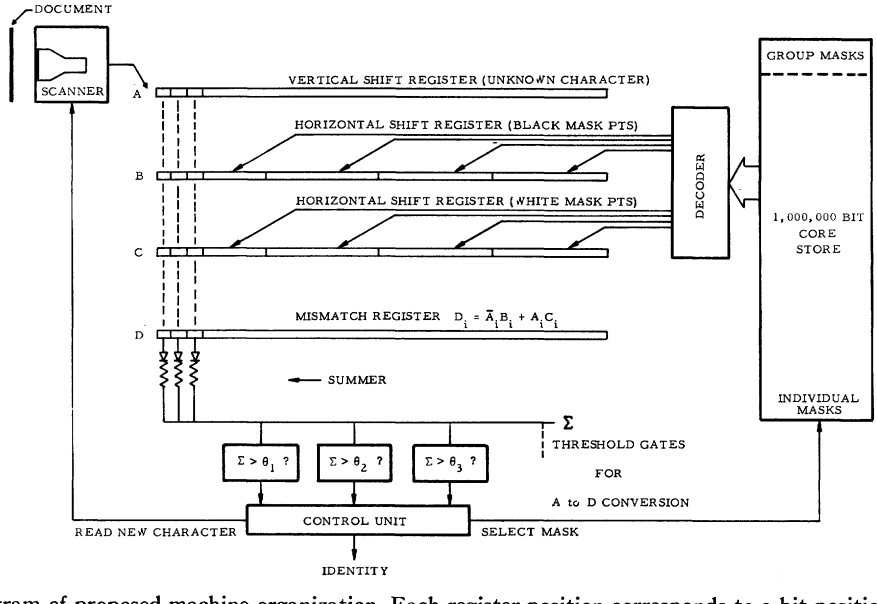


Fig. 12. Schematic diagram of proposed machine organization. Each register position corresponds to a bit position in the character; therefore, about 625 bits are processed simultaneously. While a single threshold circuit is sufficient to test for a match among the individual masks, several threshold circuits are needed to obtain the actual score on the group masks.

would the long "compare" registers, and parallel transfer capacity, be wasted.

The present simulation rate is of the order of 40 characters per minute on an IBM 7090/94 Mark II computer. The corresponding rate on an IBM System 360 Model 62 would be 800 characters per minute.

V. Results

Figures 13 and 14 summarize the various experiments on the test data. Note that the first set of tests were run on only 2000 characters, while the second made use of all 10 000 characters available. The different points on the error-reject curves were obtained by varying the mismatch threshold in the test program.

Figure 13 is a comparison of four types of masks, containing on the average 30, 37.5, 43.5, and 200 points, respectively. It is seen that while there is significant improvement in going from 37.5 to 43.5 points (from a mismatch threshold of five to seven in the mask generating program), a further increase in the size of mask does not appreciably ameliorate performance.

Figure 14 shows the effect on the error and reject rate (43.5 point masks) of changing the cutoff parameter. The cutoff parameter specifies after how many groups should the search be abandoned if a satisfactory match is not obtained. The dotted line shows the corresponding

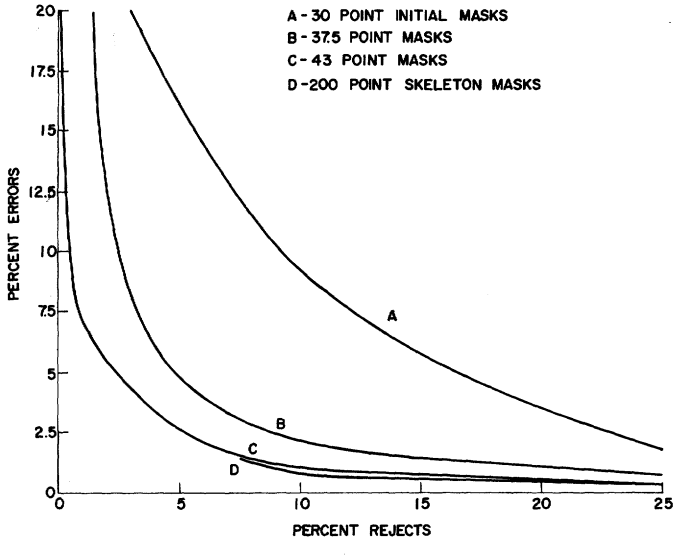


Fig. 13. Performance of four different types of masks. Curve A shows the relationships of errors to rejects for the random initial masks, curves B and C are for masks designed by the program described in Section III, and curve D describes the performance of skeleton characters used as masks.

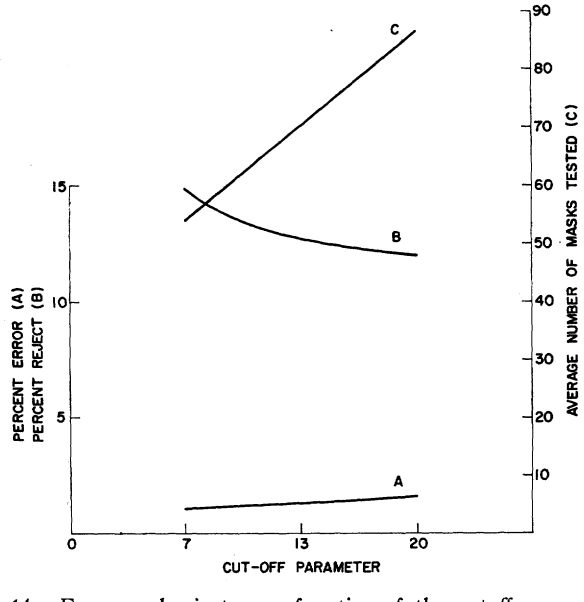


Fig. 14. Errors and rejects as a function of the cutoff parameter. Curve A is the percent error, curve B, the percent reject, and curve C, the average number of masks examined for a given value of the maximum number of groups to be examined. The masks are the same as on curve C of the previous figure, but this experiment was run on a larger number (10 000) of characters.

Fig. 15. Examples of easily confused character pairs. Because the C.R.T. beam sometimes misses very thin, horizontal lines, there may not be sufficient information in the quantized version of the character to differentiate such pairs.

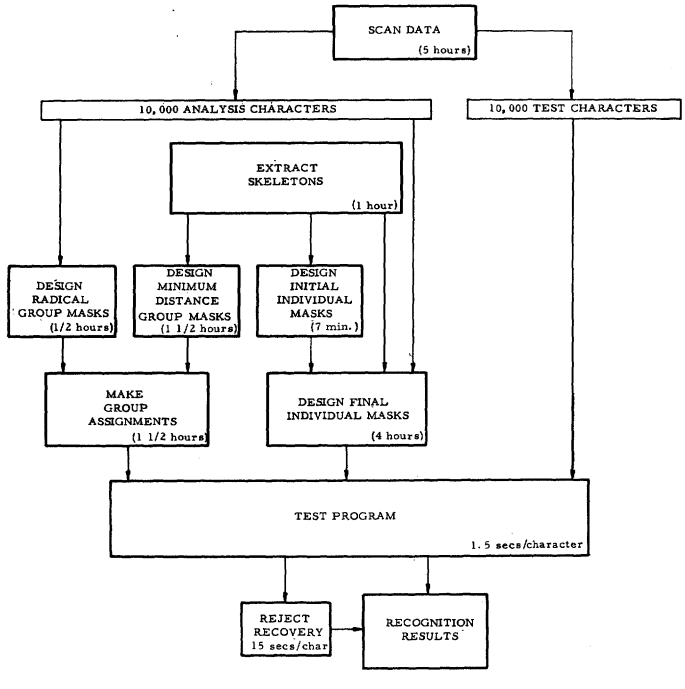


Fig. 16. Flow diagram of major computer programs. The running times relate to an IBM 7090/94 computer, except for scanning, which is controlled by an IBM 1401 computer.

TABLE II

Pass	Description	Percent of Test Data
1	Excusable errors (see text)	0.6
1	Broken and unresolved characters	1.0
2	Errors introduced by reject recovery procedures	0.2
1 and 2	Total inexcusable errors	1.2
1	Rejects before reject recovery	12.0
2	Rejects after reject recovery	7.0
1 and 2	Total rejects	7.0

savings in the number of comparison tests. A satisfactory operating point would be established by balancing the cost of processing rejects against the gain in increased throughput.

Substitution errors tend to fall into three classes. In the first class, similar characters are confused because the detail transmitted by the scanner is insufficient to differentiate them. Examples of such confusion pairs are shown in Fig. 15. An obvious remedy is to increase the resolution of the scanner, with corresponding increases in the capacity of the rest of the system. It appears, however, that most of the errors are due to missed horizontal lines, which are, on the average, four times thinner than their vertical counterparts. This might be corrected by raising the frequency response of the video threshold circuits, and perhaps asymmetrically defocusing the cathode beam to increase vertical sensitivity.

The second class of errors comprises broken and chipped test characters. The prevalence of this class of errors will likely decrease as a result of the more sophisticated thresholding and centering techniques developed since the data was scanned at the end of 1963.

The third class of errors (0.6 percent of the test data) are the "excusable" errors, due to scanning failures on the analysis set. If, for example, a broken character (counted as two characters) is included among ten analysis characters used for generating a skeleton, relatively few of the points in the central area will pass the nine-out-of-ten threshold requirement. The corresponding skeleton will be "sparse," and not enough points will be available to the mask generator to effectively discriminate against other characters. A "sparse" mask, in turn, will yield very few mismatches, and will commonly cause misidentifications. An all "don't care" mask would, of course, satisfy every video sample. Poor samples in the analysis set could be weeded out either manually, or by bit counts on the skeletons. Figure 8 shows that good skeletons always contain above 50 percent "care" points.

Presumably, the number of rejects would also be reduced by more careful editing of the analysis data, and better scanning techniques. Because of the number and length of the computer runs required for a complete evaluation cycle—Fig. 16 gives an idea of the steps involved—a full exploration of such improvements properly belongs to a larger scale development effort.

Even in the present system, however, some of the rejects can be recovered by a simple extension of the program. Rejected characters are correlated with the skeletons derived earlier—in other words, the skeletons themselves are used as masks. Here a larger shift (7 by 7) is used to decrease the effect of severe misregistration due to clipping. Table II shows the change in error and reject rate achieved by this procedure.

A decrease in the number of masks tested for each character may be obtained by ordering the masks within each group according to usage. On the basis of the data shown in Fig. 2, it is estimated that a 15 percent savings could thus be achieved on nontechnical material.

VI. Conclusions

Generalization based on a data set comprising only one thousand different characters must be subject to some reservations. With this caveat, the following conclusions are suggested.

1) The algorithm described in Section II for selecting high information points is effective for the discrimination of Chinese ideographs on the basis of the quantized video image. Parameters for both the design and the test programs may be derived from the curves in the preceding section. An error rate of 1.2 percent with a reject rate of 7 percent has been obtained on new material; whether this constitutes an acceptable performance level when extrapolated to a large data set is still a matter of conjecture.

It may be of interest to note that since its conception in connection with the Chinese program, the algorithm has also been applied to the recognition of Latin characters, both printed and Leroy lettered, and to the selection of n-tuplet type measurements. It is well suited for implementation on a large digital computer, and constitutes a systematic method of attack for many (>1000) class discrimination problems in high dimensionality (>500) binary hyperspace.

2) A two-level search procedure based on fixed group assignments offers substantial savings in the time required to identify an ideograph. The method of large masks is applicable at both levels so that the extension to two levels needs very little additional hardware.

3) Highly parallel logic with analog summation appears to be a "natural" form of implementation for the system described. Processing rates of the order of 3400 characters per minute may be attained with relatively modest core access requirements.

4) Among the clustering procedures examined, the radical extraction method yields the most stable classification whenever it is applicable. Its range is limited to about 70 percent of the Chinese alphabet.

The distance minimization algorithm is adequate for the grouping of the remaining 30 percent of the alphabet. This clustering method has also been applied to the classification of multifont typewritten characters and of handprinted numerals in the design of piece-wise linear hyperplane boundaries.

The hypotheses tested by the other two clustering routines, i.e., uniform distribution in video space and transitivity of the correlation chain, were not borne out by the experiments.

5) Further improvements in the recognition rate for Chinese characters are to be sought at the scanner end rather than in the decision logic. While the total number of video coordinate points used is almost sufficient, for

maximum efficiency the resolution in the vertical direction should be greater (perhaps by a factor of two) than in the horizontal direction. Some form of automatic threshold adjustment will be needed, especially for lower grade paper.

Finally, it seems to us that, while the results presented are necessarily incomplete, they offer sufficient promise to warrant further work in this direction.

ACKNOWLEDGMENT

The authors are indebted to many of their colleagues at the IBM Thomas J. Watson Research Center for assistance in the course of this project. Particularly to be thanked are Dr. E. N. Adams, Mrs. M. E. Barrett, J. Burns, Miss M. Miller, and Dr. C. K. Tung.

REFERENCES

[1] C. T. Abraham, "Techniques for thesaurus organization and evaluation," *Proc. American Documentation Inst.*, vol. 1, pp. 485–495, October 1964.
[2] R. E. Bonner, "A 'logical pattern' recognition program," *IBM J.*, pp. 353–360, July 1962.
[3] H. C. Chen, *Modern Chinese Vocabulary*. Society for the Advancement of Chinese Education, Shanghai: Commercial Press Ltd., 1939.
[4] C. P. Sha, *A Chinese First Reader*. Berkeley, Calif.: University of California Press, 1947.
[5] T.-T. Hsia, *China's Language Reforms*. New Haven, Conn.: The Institute of Far Eastern Languages, Yale University Press, 1956.
[6] G. W. King and H.-W. Chang, "Machine translation of Chinese," *Sci. American*, pp. 124–135, June 1963.
[7] "Survey of the need for language translation," Planning Research Corp., IBM Survey Rept. RC-634, March 12, 1962.
[8] R. J. Potter, "An optical character scanner," *J. Soc. of Photographic Instrumentation Engrs.*, vol. 2, pp. 75–78, February–March 1964.
[9] Y. R. Chao and L. S. Yang, *Concise Dictionary of Spoken Chinese*. Cambridge, Mass.: Harvard University Press, 1961.
[10] *Optical Character Recognition*. G. L. Fischer, Jr. et al., Eds. Washington, D. C.: Spartan Books, 1962.

Corrections

Y. C. Ho and R. L. Kashyap, authors of the paper "An Algorithm for Linear Inequalities and Applications," which appeared on pages 683–688 of the October, 1965, issue of these TRANSACTIONS, has called the following to the attention of the Editor.

It was established in this paper that the algorithm converges to the solution (if one exists) in a finite number of steps and that the convergence is exponential. We would like to point out that, for any finite algorithm, exponential convergence is superfluous since it is always possible to find an exponential trajectory which dominates any strictly decreasing and finite trajectory. The particular proof for exponential convergence in the paper is, however, erroneous. Nevertheless, this fact will not disturb any of the results, theoretical or experimental, mentioned in the paper.

We would like to thank C. C. Blaydon for this correction and Prof. T. M. Cover for his suggestions during the review of the paper.

Peter Fellgett, author of the short note "Logical Design of Analog-to-Digital Converters," which appeared on pages 740–741 of the October, 1965, issue of these TRANSACTIONS, has called the following to the attention of the Editor.

On page 740, column 2, the next to the last line, there is a comma after "the most significant bit" which should be deleted.