Feature analysis for symbol recognition by elastic matching

by Jerome M. Kurtzberg

A technique has been developed for the recognition of unconstrained handwritten discrete symbols based on elastic matching against a set of prototypes generated by individual writers. The incorporation of feature analysis with elastic matching to eliminate unlikely prototypes is presented in this paper and is shown to greatly reduce the required processing time without any deterioration in recognition performance.

Introduction

In References [1–3] a dynamic-programming technique for recognition of discrete handwritten symbols is described. The method is based on modeling symbols in terms of functions of input data points (termed model parameters) and performing elastic matching of these time-sequenced parameters of unknown symbols against those of a set of prototypes established by individual writers. A composite of the results of the pattern matchings is used for discrimination during recognition. This technique is useful for other applications. It was applied to speech recognition at least as early as 1968 [4–6] and to signature verification in 1978 [7].

In this paper the use of symbol features is studied to prune unlikely matches from the set of prototypes prior to the elastic matching of the unknown symbol, thereby greatly reducing the required computation time. This incorporation of feature analysis with correlation matching enables recognition processing in real time, even with the use of a highly refined computational model having several parameters to improve the resolution of symbols.

Recognition system

The data-gathering recognition system consists of a digitizing tablet, a computer with interfacing electronics, and graphic displays. The writer uses a ball-point pen (stylus) and writes on paper affixed to an electronic digitizing tablet. The trajectory of the moving pen, given in terms of a sequence of x and y coordinates, is detected at discrete points where the stylus physically touches the paper.

The writer is free to define the particular form (prototype) for each symbol, as well as the individual manner of generating it. Guidelines—a pair of parallel horizontal line segments—are established; these determine the space for the body, ascenders, and descenders of the symbols. Any token for a symbol, however, may exceed or fall short of the guidelines. As the term suggests, they are merely guides for the writer.

The tablet has a resolution of 0.005 inches (0.0127 cm), and the data are sampled at approximately 80 samples per second. Suitable filtering and smoothing are employed to reduce the effect of inaccuracies in the data-gathering hardware. For example, extra, spurious data can be recorded due to pen up-down detection problems, and there can be "jitter" due to hesitation and pausing of the writer. These are eliminated. Also, segmentation of the input symbols is performed. For the purpose of this study, it is assumed that there is no overlapping of symbols.

Elastic matching

Elastic matching is a dynamic-programming correlation technique for the comparison of data vectors having different lengths and nonlinear distortions.

Normal time-to-time differences in individual writing rate with repeated writing cause variation in the length and nonlinear time variation in the shape of a symbol. A simple correlation technique based upon dividing each symbol in time into the same number of time points addresses the length variation but does not handle the time nonlinearity. The elastic-matching procedure, however, does solve both problems by providing a mechanism to optimally match the unknown symbol token that is to be recognized against all possible elastic stretchings and compressions of each prototype (within a given range).

Symbol modeling

The modeling of the symbols (the symbol parameters) is derived from the time-sequenced point data produced by the dynamic trace of the stylus. Specifically, elastic matching takes place on a symbol model having four parameters [3]:

1. The magnitude of the angle of the tangent at each point of the curve produced by the trace (to enable discrimination of identically produced uppercase and lowercase symbols differing only in their relative vertical position).
2. The magnitude of the height of each point normalized to the given writing-space guidelines.
3. The x-offsets from the center of gravity of each symbol (to incorporate information about the relative horizontal positions of the strokes comprising the symbols).
4. The y-offsets from the center of gravity of each symbol (to enable discrimination independent of symbol position relative to the baseline; this affords the writer flexibility in positioning the symbol when the exact vertical position is not important).

Elastic matching is performed separately on each parameter. All taken together form the measure for recognition in the symbol-discrimination procedure. All parameters in this model are scaled to contribute the same weight in the composite measure [1, 2].

Pruning features

If the number of prototypes is large, computation time is a problem. The use of symbol features that are simple to compute yet efficient in eliminating unlikely prototypes from consideration as candidates for an unknown symbol can handle this difficulty. Accordingly, the following seven features are used for the pruning of the prototype set:

1. The number of strokes comprising a symbol (where a stroke is defined to be the sequence of points generated from the time the stylus is down on the tablet to the time it is removed).
2. The number of points in the symbol (after appropriate filtering of the point sequence to remove inaccuracies due to hardware peculiarities and to writer hesitation).
3. The number of points per stroke (as a sharper measure than the previous feature).
4. The height of the lowest point.
5. The height of the highest point.
6. The height of the lowest point per stroke.
7. The height of the highest point per stroke.

Additional, more complicated features, such as curvature and number of abrupt segment changes, may also be used, but the above set appears to be sufficient.

The pruning features need be computed only once for each prototype and can then be stored with the associated parametric model for each symbol. Prior to elastic matching, the features of the unknown symbol are compared to those of the prototype to ensure that *all* of the respective pairings are within acceptable tolerances. Since the elastic matching comprises the vast majority of the recognition time, and the pruning-feature computation is insignificant, the computation time for recognition is greatly reduced.

Experimental results, presented in the next section, show that there can be significant pruning with no decrease in the symbol-recognition rate, or extreme pruning with only a slight drop in the recognition rate.

Experimental results

To evaluate the effects of pruning, experiments were conducted on the same data used to evaluate the elastic-matching approach operating on a refined multiparameter model [3]. The symbol set consisted of the English uppercase and lowercase letters, the digits, and ten punctuation symbols (a total of 72 symbols), namely

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
( ) ! : ; " ' , . ?

The particular data used in the evaluation consisted of eight tokens of each symbol from a left-handed writer (not the author) who sequentially generated eight copies of the symbol set over a few sessions. In the first experiment, the prototypes consisted of the first complete set of tokens, thereby forming 72 prototypes. The last four sets of data, consisting of 288 unknowns, are matched against the prototypes. This experiment is designated by the notation *A(1;5678)*, where "1" indicates data from the first data set and "5678" from sets 5, 6, 7, and 8.

The set of prototypes is then updated by training on the second set of tokens and incorporating the incorrectly recognized tokens in the prototype set. The resulting

Table 1 Recognition with pruned prototypes.

Experiment	*No. of prototypes*	*No. of errors*	*Percentage recognized*	*Percentage pruned*	*Candidate prototype*
A(1;5678)	72	16	94.4	60.6	28
A(12;5678)	94	5	98.3	61.4	36
A(123;5678)	110	4	98.6	60.4	44
A(1234;5678)	121	3	99.0	61.5	47

Table 2 Baseline comparison—no features.

Experiment	*No. of prototypes*	*No. of errors*	*Percentage recognized*	*Percentage pruned*	*Candidate prototype*
A(1;5678)	72	17	94.1	26.1	53
A(12;5678)	95	5	98.3	26.3	70
A(123;5678)	111	4	98.6	25.0	83
A(1234;5678)	122	3	99.0	24.4	92

Table 3 Baseline comparison—constrained warping path.

Experiment	*No. of prototypes*	*No. of errors*	*Percentage recognized*	*Percentage pruned*	*Candidate prototype*
A(1;5678)	72	23	92.0	47.3	38
A(12;5678)	93	11	96.2	47.3	49
A(123;5678)	110	8	97.2	45.6	60
A(1234;5678)	123	7	97.6	44.9	68

augmented set of prototypes is matched against the same last four sets of tokens as before. This is designated by *A(12;5678)*. The process of updating the prototype set by training and subsequent matching is continued for the third and fourth sets of tokens, namely *A(123;5678)* and *A(1234;5678)*, respectively.

Table 1 presents the results for recognition with pruning features in which there is no degradation of recognition accuracy. The number of prototypes in the augmented prototype set is listed for each updating along with the percentage of prototypes pruned and the resulting number of candidate prototypes for the elastic-matching procedure. Note that the percentage of prototypes pruned was relatively constant for each updated prototype set.

The specific tolerances for the pruning features in this experiment are that

- The number of strokes for the prototype and for the unknown may differ by one.
- The difference between the overall highest and lowest points, respectively, must be within approximately one third of the unit distance established by the guidelines.
- The corresponding numbers of total points (the lengths) must be within a difference of 75% of each other, i.e., lie within a range of 3/4 to 4/3 of each other.

Per-stroke tolerances are set high so as to be ignored by the algorithm.

Table 2 shows the baseline comparison in which no pruning features are used. The elastic-matching procedure used can stretch or compress the symbol to be matched by no more than double or half itself; hence, matching can take place only if the prototype and the unknown differ in length within this range. There is, thus, a necessary elimination of all prototypes that lie outside this range. The percentage of prototypes thereby eliminated (or pruned) is shown along with the resulting reduced number of candidate prototypes. Note that for the same recognition accuracy of 99.0% as obtained with the use of pruning features, almost double the number of prototypes must be evaluated via elastic matching without features.

An attempt to improve the computation time of elastic matching by limiting the expansion and compression to within 50% has been made in the past [8]. In effect, this reduces the permissible "warping path." **Table 3** presents the baseline comparison in which no pruning features are employed, but the warping path is constrained to a comparatively narrow range associated with the 50% constraint. Note that with this technique the recognition accuracy for the updated prototypes decreases to 97.6%, and the number of candidate prototypes to be evaluated

Table 4 Sharply pruned prototypes.

Experiment	*No. of prototypes*	*No. of errors*	*Percentage recognized*	*Percentage pruned*	*Candidate prototype*
A(1;5678)	72	31	89.2	87.9	9
A(12;5678)	87	22	92.4	87.8	11
A(123;5678)	103	18	93.8	88.1	12
A(1234;5678)	112	13	95.5	88.6	13

Table 5 Recognition vs. pruning for *A(1;2345678)*.

Experimental matching	*No. of errors*	*Recognition accuracy*	*Percentage pruned*	*Candidate prototype*
Case 1	28	94.4	61.6	28
Case 2	31	93.9	64.8	25
Case 3	36	92.9	67.4	23
Case 4	51	89.9	79.9	14
Case 5	54	89.3	88.0	9
Case 6	58	88.5	89.0	8
Baseline runs				
No features	29	94.3	26.0	53
Constrained warping path	37	92.7	47.2	38

increases considerably over the case of light pruning in which features are used (Table 1).

Table 4 gives an illustration of recognition with sharply pruned prototypes. The final recognition accuracy drops to 95.5%, but with pruning of 88.6%, only 13 prototypes need be evaluated.

The tolerances for this specific experiment are that

- The numbers of strokes must be the same for the prototype and the unknown.
- The difference between the overall highest and lowest points, respectively, on both a total and per-stroke basis, must be within approximately one third of the unit distance established by the guidelines.
- The corresponding numbers of total points (the lengths) must be within a difference of 60% of each other and 70% on a per-stroke basis.

Table 5 shows the trade-off between recognition accuracy and prototype pruning for *A(1;2345678)*. In these experiments, no updating was performed, and the first set of 72 prototypes was matched against seven sets of data; each set was composed of 72 tokens for a total of 504 unknown symbols, which implies 36288 prototype examinations. Tighter tolerances for the pruning features were set for each successive experiment.

Specifically, the first, second, and third matchings (i.e., Cases 1–3) varied in the tolerances for the corresponding total lengths of the prototype and the unknown, allowing them to be, respectively, within 70%, 60%, and 50% of each other. The other tolerances were held constant, namely

- The numbers of strokes were permitted to differ by one.
- The per-stroke tolerances were set high so as to be ignored by the algorithm.
- The differences between the highest and lowest points, respectively, were within approximately one third of the guideline distance.

Case 4 continued to allow the numbers of strokes to differ by one, the corresponding total and per-stroke lengths by 70%, and the corresponding total and per-stroke differences of the highest and lowest points, respectively, to be within approximately one third of the guideline distance.

In Cases 5 and 6, the number of strokes was constrained to be the same; the total-length tolerance was held to 60% and 50%, respectively. All of the per-stroke and highest- and lowest-point tolerances were the same as in the previous experiment.

The two baseline runs, described previously, are shown at the bottom of the table. Interestingly, "light" pruning (Case 1) removed a prototype that the elastic-matching procedure had incorrectly selected as its first choice, thereby in this instance slightly improving the recognition accuracy.

Conclusions

This study demonstrates that significant savings in computation time can be achieved in correlation-type recognition techniques, such as elastic matching, by the introduction of simple feature analysis. Actual results show that pruning of the candidate prototypes yields a doubling of recognition speed with no loss in accuracy. (A much sharper pruning with looser tolerances resulted in an increase by a factor of seven in processing speed but a loss of 3.5% in recognition accuracy.)

This combination of feature and correlation analysis provides a powerful approach for a real-time handwriting-recognition system. Other recognition problems may also benefit from the use of these techniques.

Acknowledgments

I wish to express my appreciation to C. C. Tappert for discussions and aid in incorporating the pruning routines into the symbol-recognition system, and to M. Karnaugh for encouragement and support in this work.

References

1. J. M. Kurtzberg and C. C. Tappert, "Symbol Recognition System by Elastic Matching," *IBM Tech. Disclosure Bull.* **24,** No. 6, 2897–2902 (November 1981).
2. C. C. Tappert and J. M. Kurtzberg, "Elastic Matching for Handwritten Symbol Recognition," *Research Report RC-9988*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, May 23, 1983.
3. M. Karnaugh, J. M. Kurtzberg, and C. C. Tappert, "Improved Parameter Set for Adaptive Symbol Recognition," *IBM Tech. Disclosure Bull.* **24,** No. 1B, 769–771 (June 1981).
4. T. K. Vintsyuk, "Speech Discrimination by Dynamic Programming," *Kybernetika* (*Cybernetics*) **4,** No. 1, 81–88 (1968).
5. V. M. Velichko and N. G. Zagoruyko, "Automatic Recognition of 200 Words," *Int. J. Man-Machine Studies* **2,** 223–234 (1970).
6. H. Sakoe and S. Chiba, "A Dynamic Programming Approach to Continuous Speech Recognition," *Proceedings of the 7th International Congress on Acoustics*, August 1971, Paper No. 20, p. C13.
7. M. Hanan, N. M. Herbst, and J. M. Kurtzberg, "Finite State Machine Decision Procedure Model for Signature Verification," *IBM Tech. Disclosure Bull.* **20,** No. 8, 3355–3360 (January 1978).
8. C. C. Tappert and S. K. Das, "Memory and Time Improvements in a Dynamic Programming Algorithm for Matching Speech Patterns," *IEEE Trans. Acoust., Speech & Signal Proc.* **ASSP-26,** No. 6, 583–586 (December 1978).

Received July 8, 1986; accepted for publication August 14, 1986

Jerome M. Kurtzberg *IBM Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598.*

Mr. Kurtzberg received the B.S. degree in mathematics from Temple University, Philadelphia, Pennsylvania, in 1954 and the M.S. degree in electrical engineering from the University of Pennsylvania, Philadelphia, in 1961. From 1955 to 1958 he was with Univac and from 1958 to 1960, with Auerbach Corporation, Philadelphia. In 1960 he joined the staff of the Burroughs Research Center, Paoli, Pennsylvania, where he was manager of research in design automation. From 1964 to 1967 he was a consultant at Auerbach Corporation and Kettelle Associates, specializing in operations research problems. Since 1967 he has been a research staff member at the IBM Thomas J. Watson Research Center, where his research interests have included design automation, handwriting recognition, and the modeling and design of computer systems. Concurrent with his duties at the Watson Research Center, he served in 1968 as a consultant to the Executive Office of the President, Office of Emergency Planning. Mr. Kurtzberg is a member of the Scientific Research Society of America and a senior member of the IEEE; he was Chairman of the IEEE Technical Committee on Design Automation in 1966 and 1967 and an Associate Editor of *Management Science* from 1976 to 1980. In 1969 he was a Lecturer for the IEEE Distinguished Visitors program. He is a coauthor of *Microelectronics in Large Systems*, Spartan Books, 1965, and of *Design Automation of Digital Systems: Theory and Techniques*, Prentice-Hall, 1972.