XVII. COGNITIVE INFORMATION PROCESSING

Academic and Research Staff

Prof. Murray Eden
Prof. Jonathan Allen
Prof. Barry A. Blesser
Prof. Francis F. Lee
Prof. William F. Schreiber
Prof. Donald E. Troxel
Prof. Ian T. Young
Dr. Jack E. Bowie
Dr. Ralph W. Gerchberg
Dr. Uri F. Gronemann
Dr. Martin J. Hewlett
Dr. Conrade C. Jaffe*
Dr. Perry L. Miller†
Dr. Mary J. Naus‡
Dr. David M. Ozonoff
Dr. Stephanie E. Sher
Francis X. Carroll
Ali M. Fakhr
Jeffrey A. Hagerich
M. Sharon Hunnicutt
Eric R. Jensen
Kenneth A. Okin
Robert A. Piankian
Kenneth P. Wacks

Graduate Students

Robert P. Bishop
Brian E. Boyle
Becky J. Clark
Michael A. Cohen
Charles H. Cox
Caleb J. Drake
James R. Ellis, Jr.
Irvin S. Englander
Steven G. Finn
Harry S. Forsdick
Walter E. Gibbons
Alan M. Gilkes
Richard S. Goldhor
René M. Haas
David B. Harris
Marvin E. Jernigan
Clayton K. S. Kau
Richard C. Kinnaird, Jr.
John W. Klovstad
Theodore T. Kuklinsky
Sze-Ping Kuo
Donald S. Levinstone
Charles W. Lynn
Lee C. Makowski
Dennis V. Marsicano
Douglas O'Shaughnessy
John E. Ostrowski
Douglas B. Paul
Roger S. Putnam
Robert J. Shillman
Robert D. Solomon
Jan A. Spriet
Christopher E. Strangio
Hoo-min D. Toong
Bik-Kwoon Tye
Joseph E. Wall, Jr.
Allen W. Wiegner
Harvey M. Wolfson
Kenway J. Wong
Tonny Y. Wong
Gregory W. Zack

A. A LETTER STRUCTURE GRAMMAR, WITH APPLICATIONS TO MULTIFONT CHARACTER RECOGNITION JS

National Science Foundation (Grant GK-3373X1)
Joint Services Electronics Program (Contract DAAB07-71-C-0300)
National Institutes of Health (Grant 5 PO1 GM14940-07)

Charles H. Cox, Barry A. Blesser, Murray Eden

1. Introduction

Research on multifont, automatic character-recognition machines has been in progress for ten years. With more than 3000 type fonts in common use (including 300 typewriter fonts) there is enormous variability in the representation of any character. One approach to the multifont problem is to augment the recognition algorithms with a contextual processor. This method is usually implemented by incorporating linguistic context constraints such as diagram statistics[1,2] or restricting the output to be an element

*Research Affiliate from the Department of Biology.

†Picker Foundation Fellow.

‡Also Assistant Professor of Psychology at Wellesley College. JS

JS of a dictionary.[3-5] In this approach it is assumed that the given recognition algorithm is nearly optimum, in the sense that all available information has been extracted from the image. There are common situations in which such an approach is completely useless, for example, in the recognition of numbers, which have little if any context in the conventional sense.

Another approach to the multifont problem is to consider all possible representations of a character and from these to find the average character or archetype.[6-9] Work then proceeds to recognize the archetype as well as possible. This is basically an engineering approach: averaging to reduce the effects of local variability. Indeed, such an averaging approach has proved to be quite powerful, for example, in communication theory. Note, however, that this approach is most powerful when the local values of the variable parameters are inherently unpredictable.

An alternative approach, which we report here, is based on the assumption that the variability of characters is due to two sources: the type font to which the character belongs, and noise. Consider the copies of the English letters shown in Fig. XVII-1. Clearly, the style of any given letter, say (B), is different in each type font. But within any one type font certain aspects of the style of (B) are identical to stylistic aspects of other characters in that font. Compare, for example, the upper and lower corners on the left side of (B) and (D). While there may be no way of accounting for the various forms of any one character among type fonts, within a given type font the variations of each character from its respective archetype evidently follow some implicit design convention. The repetition of stylistic features among characters within a type font will be termed the stylistic consistency of a font. A framework within which this repetition of stylistic aspects in a type font can be expressed explicitly may be used subsequently to augment the recognition algorithms.

2. Stylistic Consistency

For an intuitive feeling of the basis of stylistic consistency, consider a set of characters drawn from a single roman type font. Since it is a roman font, serifs are part of the stylistic consistency. Furthermore, the shape or design of the serifs is a part. Of course, not all of a font's style is necessarily carried by serifs: the design of the strokes composing the characters would also seem to be part of a font's stylistic consistency. Thus, one part of a font's stylistic consistency is a set composed of each serif and stroke style.

Knowledge of the serif and stroke designs, however, is not enough to describe completely the roles of serifs and strokes in the stylistic consistency of a given font. For example, this information does not make it possible to tell whether serifs occur at such places as the top of an (A) or the lower end of a (C). Another set must be formed which contains the relations between each serif and the stroke to which it is joined.
JS

JS

Fig. XVII-1. Upper-case characters from 8 different type fonts, depicting the wide variability among fonts.

Formally, stylistic consistency will be defined as those algorithms which take characters as input, and will output the members of two sets, the set of elements and the set of relations. The set of elements contains two subsets. A suitable description of each marker forms one subset of the set of elements and we shall refer to it as the set of serifs. A suitable description of each stroke style forms the other subset and we shall refer to it as the set of strokes. The set of relations contains the relationships that are present among serifs and strokes of the characters under consideration.

The task of expressing the stylistic consistency of any machine or hand-printed character is enormous. Shillman, Kuklinski, and Blesser[10] have presented results of an initial investigation of the stylistic consistency of hand-printed characters. Here we deal only with machine-printed, upper-case characters composed of two stroke styles at most, taken from roman or sans-serif families. The set of elements is assumed to be given because the determination of the elements is basically a question of detection. JS

JS Hence it can be discussed more appropriately when an implementation is under consideration, rather than as part of a theoretical study such as this. Therefore, the main emphasis of this work is to develop the members of the set of relations. We found it convenient to express the required relations implicitly in the form of two main groups of grammarlike rules. One group, Stroke Style rules, predicts the placement of different stroke styles for a given letter. For example, given a "skeleton" letter (A), i.e., one with no serifs and with all strokes initially of the same width, the stroke rules predict the strokes that should be made thicker so that the given (A) would look as though it belongs to a roman type font. The other group of rules, Serif Placement rules, predicts the placement of serifs for a given letter. In our example, serif rules predict where serifs should be added to the skeleton (A) to make it consistent with an (A) from a roman type font.

3. Stroke Style Rules

The rules for recording which strokes will have a given style are made with reference to Fig. XVII-2. A useful notion is that of having two sets of orientations; this provides the flexibility that is needed in dealing with skewed type fonts, as with italic, for example. The orientations suggested by such large-scale phenomena as the outline of the page or the run of the type are objective orientations: for example, objective vertical O_v and objective horizontal O_h. The orientations defined by the individual characters themselves are relative orientations; for example, relative vertical R_v, and relative horizontal R_h. Lines oriented between R_v and R_h are assigned orientations R_{vh} or R_{hv}.

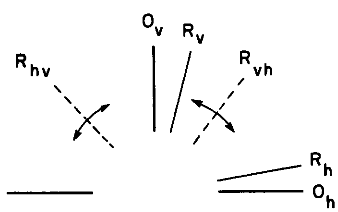


Fig. XVII-2.
Orientation conventions for stroke style rules. Objective orientations are denoted "O"; relative orientations, "R".

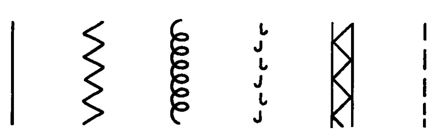


Fig. XVII-3.
Typical straight strokes.

In the rules discussed below the terms "straight" strokes and "curved" strokes refer to notions broader than those conventionally associated with these terms. A straight stroke is one to which a single orientation is ascribed. Geometrically straight strokes readily fulfill this extended notion of straight. This definition also allows strokes such as JS those shown in Fig. XVII-3 to be included. Curved strokes, then, are strokes that are not straight or, equivalently, strokes to which more than one direction can be ascribed. Note that either style of stroke may be either straight or curved. JS

In an analogous manner, when two distinct styles are required the styles are given the generic titles of "thick" and "thin." When the stroke styles do indeed come in two physically different thicknesses the generic titles take on their usual interpretation. Stroke combinations such as plain/decorated, single/combination will also be assigned the appropriate thick or thin designation. Any of the stroke styles shown in Fig. XVII-3 may be either thick or thin.

The <u>main</u> <u>stroke</u> <u>style</u> <u>rules</u> are to be applied in the following order.

1. Every letter has at least one thick stroke.
2. Two or more thick strokes never meet.
3. Straight strokes parallel to orientation R_h are always thin.
4 (a). Curved strokes are thin where they are parallel to R_h.
   (b). Curved strokes are thick where they are parallel to R_v.
5 (a). Straight strokes parallel to R_{hv} are always thick.
   (b). Straight strokes parallel to R_{vh} are thin (unless this violates rule 1, in which case they are thick).
6. Straight strokes parallel to R_v are thick (unless this violates rule 2, in which case they are thin).

Figure XVII-4a shows a type font that follows all rules as stated. The type font shown in Fig. XVII-4b follows all rules with the exception that (U) is not predicated correctly. A comparison of (U) in Fig. XVII-4a and 4b reveals that the style of (U), which

Fig. XVII-4.
Two type fonts, with stroke assignments.

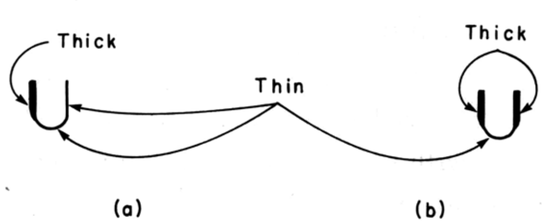


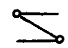
Fig. XVII-5.
Two basic forms of upper-case (U).

is seemingly independent of the style of the remaining characters in the font, appears in at least two different forms, as shown in Fig. XVII-5. JS

JS

With so much variation in design among type fonts, idiosyncratic deviations of an individual character in a particular type font are to be expected. To deal with these font-dependent deviations from the main rules, we introduce another related set of rules, termed stroke style dialect rules. These dialect rules pertain to specific letters and when used override the stroke conventions established by the main rules. The stroke style of a font is thus determined by the stroke style rules and any applicable dialect rules.

For the fonts studied, three dialect stroke style rules are required.

1. The right vertical stroke of (U) is thin.
2. The basic structure of (S) is , with the stroke style in transition regions (indicated by circles) arbitrary.
3. The basic structure of (Y) is .

4. Marker Placement Rules

Before presenting rules for recording where the members of the marker set occur, a few conventions, as well as another definition, that of a corner, are required. A corner is the intersection of a straight stroke end and either another straight stroke end or a curved stroke end; that is, a corner is exactly two stroke ends coming together, one of which is straight. Table XVII-1 gives the convention for determining the orientation of the marker relative to the stroke end to which it is attached. The orientation of a marker is defined by the orientation of the major axis of an ellipse that circumscribes it.

Table XVII-1. Marker-to-Stroke Orientation Convention.

Stroke Orientation	Marker Orientation
straight R_v	R_h
straight R_h	R_v
curved	R_v

The members of the marker set are primary and secondary. The number of primary markers is limited to two, slab and bracket serifs, and they are used for virtually all markers. Usually the slab serif has a simple geometric outline and abuts the stroke directly. The bracket serif often has rounded or curved edges that form a kind of bracket with the stroke to which it is attached. As with the stroke terms thick and thin, the marker terms slab and bracket are adopted as generic titles for the two kinds of primary markers. Typically, but not necessarily, the design of the primary markers conforms to this description.

JS

There may be any number of secondary markers, although frequently there are only

two of these. These secondary markers usually occur only at one place in an entire font. JS
Typically, one location of a secondary serif is the bottom of a (J); another secondary serif occurs on the bottom right leg of the (R). Should additional secondary markers be required, they may be added to the marker set, provided that rules appropriate for their use are added to the set of marker rules.

In some fonts degeneracies may occur where one or more of the distinct markers listed above takes on the same design. Clearly, this causes no problems with the rules. Furthermore, the rules have the ability to modify the appearance of a marker. For example, the most common modification will be the removal of a portion of a marker.

The main serif placement rules are listed as follows.

1. All stroke ends have markers:
   (a) strokes parallel to R_h (including curve stroke ends) receive a bracket serif;
   (b) strokes parallel to R_v receive a slab serif.
2. All vertical strokes terminating in a corner receive a slab serif.
3. Markers on the right leg of (R) and the bottom of (J) are secondary.

The dialect serif placement rules are formulated as follows:

1. Markers on horizontal stroke ends (including curve stroke ends) do not extend beyond the run of the type.
2. Markers on corners composed of one R_{hv} (R_{vh}) stroke extend only to the left (right) of the vertical stroke.
3. Markers on curve stroke ends extend only to the left (where left is determined by standing on the stroke and facing the end).
4. Markers on all thin strokes are bracket serifs.
5. A marker on the lower end of (C):
   (a) extends only to the right;
   (b) is not present.
6. For markers on the corners of (N):
   (a) the one on the lower right corner is not present;
   (b) neither marker is present.
7. There is a secondary marker on the top of (A).
8. There is a secondary marker on the middle corner of (W).

(a) ABCDEFGHIJKLMNOPQRSTUVWXYZ

(b) ABCDEFGHIJKLMNOPQRSTUVWXYZ

(c) ABCDEFGHIJKLMNOPQRSTUVWXYZ

Fig. XVII-6. Three type fonts whose marker rules have been given. JS

JS

For the type font shown in Fig. XVII-6a, in addition to the main rules, dialect rules 1, 2, 3, 5b, and 6a are in effect. In contrast, for the type font shown in Fig. XVII-6b, only dialect rule 6b is in effect. Finally, for the type font shown in Fig. XVII-6c dialect rules 1, 2, 3, 4, 5b, 6a, and 8 are in effect.

5. Conclusion

It seems apparent that the stylistic consistency of each character in a font relative to the stylistic expression of the same character in another font offers us more information than has previously been used. That the information conveyed by a font's stylistic consistency will be useful follows from the obvious fact that all characters of any one word are almost always printed in the same font.

This consistency has interesting implications for the rest of the recognition process. The feature detection portion of the recognition process is typically preceded by a fixed mode processor such as a normalization or lowpass filter operator. These preprocessors have been reasonably successful when the input is limited to a few type fonts, but they are not suitable for multifont input. For example, it is entirely possible that the degree of lowpass filtering that works well for a roman font would completely obscure the lines of a sans-serif font. The stated aim of preprocessing is to reduce the variability with which feature-detection algorithms must deal. In light of the work reported here, it seems quite reasonable to envision a modifiable preprocessor. One mode of operation for such a modifiable preprocessor would be to analyze elements of the input to determine certain aspects of their stylistic consistency. This information would then be made available to the feature-detection and recognition algorithms. Such a modifiable preprocessor would be capable of processing each string of characters in a nearly optimum manner. It would provide a means for dealing with a wide variety of variations, including the subtle variations among type fonts which often prove not to be subtle to the feature-detection and recognition algorithms.

The work reported here is a digest of a paper, entitled "The Application of Type Font Analysis to Automatic Character Recognition," which is to be presented at the Second International Conference on Pattern Recognition, Lyngby-Copenhagen, Denmark, August 13-15, 1974.

References

1. J. Raviv, "Decision Making in Markov Chains Applied to the Problem of Pattern Recognition," IEEE Trans., Vol. IT-13, No. 4, pp. 536-551, October 1967.
2. E. M. Riseman and R. W. Ehrich, "Contextual Word Recognition Using Binary Diagrams," IEEE Trans., Vol. C-20, No. 4, pp. 397-403, April 1971.
3. Y. T. Chien and R. Ribak, "A New Data Base for Syntax-Directed Pattern Analysis and Recognition," IEEE Trans., Vol. C-21, No. 7, pp. 790-801, July 1972.

JS

4. Carl S. Christensen, "An Investigation of the Use of Context in Character Recognition...," Cornell University Center for Applied Mathematics, November 1968. JS

5. R. B. Thomas and M. Kassler, "Character Recognition in Context," Inform. Contr. 10, 43-64 (1967).

6. A. I. Frank, "Multiple-Font Reading, a System Description," IEEE Systems Design, Vol. 1, No. 3, pp. 21-25, 1964.

7. S. B. Gray, "A Highly Flexible, Many-Font General-Purpose Page Reader," American Documentation Institute, 26th Annual Meeting, Chicago, Illinois, October 1963, pp. 85-86.

8. L. A. Kamentsky and C. N. Liu, "Computer-Automated Design of Multifont Print Recognition Logic," IBM J. Res. Develop., Vol. 7, No. 1, pp. 2-13, January 1963.

9. C. N. Liu and G. L. Shelton, Jr., "An Experimental Investigation of a Mixed-Font Recognition System," IEEE Trans., Vol. EC-15, No. 6, pp. 916-925, December 1966.

10. R. J. Shillman, T. T. Kuklinski, and B. A. Blesser, "Experimental Methodologies for Character Recognition Based on Phenomenological Attributes" (submitted to the Second International Conference on Pattern Recognition, Lyngby-Copenhagen, Denmark, August 13-15, 1974).

11. B. Blesser, R. Shillman, C. Cox, T. Kuklinski, J. Ventura, and M. Eden, "Character Recognition Based on Phenomenological Attributes," Visible Language, Vol. 7, No. 3, pp. 209-223, 1973. JS