

**DES PRIMITIVES AUX LETTRES - UNE METHODE
STRUCTURELLE DE RECONNAISSANCE
AUTOMATIQUE D'ECRITURE CURSIVE**

H.OULHADJ, G.LORETTE, E.PETIT, J.LEMOINE,
M.GAUDAIRE

LABORATOIRE DE GENIE ELECTRIQUE DE CRETEIL - UNIVERSITE PARIS VAL-DE-MARNE
Avenue du Général de Gaulle - 94010 CRETEIL Cédex - FRANCE

RÉSUMÉ

Nous exposons, dans cet article, une méthode structurelle de lecture automatique d'écriture cursive. Une phase d'apprentissage permet d'établir un dictionnaire de mots et une base de données. Celle-ci contiendra les lettres de l'alphabet codifiées par un couple de primitives (P1, P2) caractérisant la forme de leur tracé. Le mot à reconnaître est segmenté par la primitive P1. A chaque segmentation, l'extraction de P2 permet d'identifier la lettre associée au couple (P1, P2). Ceci génère des listes de lettres que le dictionnaire oriente sur les seuls mots du vocabulaire enregistré. Parmi l'ensemble des candidats, celui qui respecte le mieux l'enveloppe moyenne de l'écriture est alors retenu.

SUMMARY

We present a structural method for automatic cursive script recognition. By a training step, a dictionary and a database are defined. This database contains a representation of each alphabetical letter coded by meanth of two primitives (P1, P2) which are deduced from each drawing. In the recognition step, the input word is segmented with primitive P1. Then, P2 extraction allows to identify the letter associated with (P1, P2) couple in database. Thus, we obtain a sequences of identified letters which define candidates words in the recorded dictionary. Finally, we retain the candidate which has the best respect to meanth space factor of handwriting.



I - INTRODUCTION

La lecture automatique d'écriture cursive manuscrite pose un problème de reconnaissance des formes qui reste encore non résolu. Actuellement, deux démarches peuvent être adoptées : soit reconnaître globalement un mot, soit reconnaître les caractères qui le forment.

La première approche (1, 2, 3, 4) nécessite la mise en mémoire de toutes les représentations des mots de la langue étudiée. Celles-ci sont évidemment nombreuses, d'autant qu'un mot donné peut être écrit différemment, non seulement par deux scripteurs distincts, mais aussi par un scripteur unique. Cette méthode est précise pour des vocabulaires de petites tailles mais devient vite très lourde pour des vocabulaires plus importants : on est amené à manipuler beaucoup de données et les mots mal orthographiés ne sont pas identifiés car ils n'existent pas dans le dictionnaire considéré.

La seconde approche (5, 6, 7, 8) consiste à segmenter au préalable le mot et à identifier ensuite les caractères qui le forment. La segmentation s'effectue en général au niveau de l'intersection de deux traits, des arches ou des points de rebroussement qui indiquent que l'on est à la limite entre deux caractères ou deux composantes d'un caractère. Cette méthode semble plus réaliste mais pose le problème de la segmentation du mot: opération sujette à des erreurs inévitables. De nombreux auteurs ont étudié le problème mais les solutions proposées restent encore insuffisantes.

Notre étude vise à montrer que la deuxième approche est encore possible et peut être améliorée si le problème de la segmentation est contourné. La démarche que nous proposons ne suppose pas une segmentation du mot a priori. Celui-ci sera segmenté au fur et à mesure de l'identification des lettres dans son tracé. Ainsi, la segmentation du mot en caractères découle plus de la reconnaissance des lettres qu'elle n'y contribue.

II - APPRENTISSAGE

II-1 ETABLISSEMENT DU DICTIONNAIRE

Les mots à reconnaître sont classés d'après la méthode de décision de l'arbre binaire. Le dictionnaire est donc organisé en arbre. Les noeuds représentent une lettre de l'alphabet et les feuilles, le mot formé par le parcours d'une branche de l'arbre. Reconnaître un mot consiste alors à trouver un chemin possible dans le dictionnaire.

II-2 ETABLISSEMENT DE LA BASE DE DONNEES

II-2-1 Saisie de traces, extraction de primitives et codage.

La saisie des caractères s'effectue sur tablette graphique numérique. Une opération de filtrage angulaire et en distance, appliquée sur les tracés, permet de réduire les distorsions et la redondance de l'information (9, 10). L'identité d'une lettre est déterminée par un couple de primitives (P1, P2). P1 est une séquence de codes directionnels définissant l'évolution temporelle des tracés. P2 est un invariant définissant l'encombrement spatial de chaque caractère.

L'obtention de P1 se fait en deux étapes. En un premier temps, on associe au tracé le code de Freeman 16 DIRECTIONS (11). Ce dernier est ensuite ramené à un code 4 DIRECTIONS grâce à une base de connaissances contenant des règles de réécriture obtenues expérimentalement (cf. figure 1). Le but de cette opération est d'obtenir un code simple, invariant par transformation géométrique, ce qui évite de normaliser ou de redresser les lettres avant de les coder (12).

Lors de l'apprentissage, P2 est introduite au clavier pour indiquer l'appartenance de chaque lettre à une classe parmi 4 préalablement définies. Elle peut prendre la valeur 1, 2, 3 ou 4 correspondant aux situations suivantes : lettre sans hampes (e), à une seule hampe (l), à deux hampes (f), ou à hampe vers le bas (j).

II-2-2 Classification des caractères et enrichissement de la base de données.

Les lettres sont classées d'après la méthode de décision de l'arbre binaire. La base de données est donc organisée en arbre. Les noeuds représentent les codes directionnels de la primitive P1 et les feuilles, l'ensemble : primitive P2 et lettre associée au couple (P1, P2). La base de données s'enrichit par de nouvelles représentations de lettres en considérant d'une part plusieurs scripteurs au cours de l'apprentissage (système multiscripteurs), et en appliquant d'autre part sur les lettres prototypes un générateur de formes basé sur un modèle de dégénérescence (apprentissage par présentation de modèles). Dans ce dernier cas, les nouvelles représentations obtenues sont pondérées par un coefficient de vraisemblance qui indique leur degré de parenté avec le prototype qui les a générées.



III - ALGORITHME DE RECONNAISSANCE

III-1 PRINCIPE GENERAL

La reconnaissance d'un mot s'effectue selon une stratégie de prédiction/vérification appliquée sur les lettres et une opération de décision appliquée sur les mots. La prédiction des lettres consiste à analyser les codes directionnels d'un tracé d'après l'arbre de décision binaire permettant de classer les caractères dans la base de données. Leur vérification consiste à analyser les listes générées d'après l'arbre de décision permettant de classer les mots dans le dictionnaire. L'opération de décision consiste enfin, à sélectionner parmi l'ensemble des mots candidats, celui dont l'aspect global respecte le mieux le tracé du mot d'entrée.

III-2 PREDICTION DES LETTRES

Après plusieurs traitements (filtrage, extraction des extrema et estimation de l'enveloppe moyenne d'écriture), le mot d'entrée est représenté par trois attributs : une liste de minima, une liste de maxima et une séquence de codes 4 directions. La reconnaissance peut alors s'effectuer en balayant la séquence de codes dans le sens de l'écriture. A chaque pas du balayage, nous recherchons une correspondance entre les noeuds de la base de données et les codes du mot d'entrée. Quand une branche est entièrement parcourue dans la base de données, une primitive de type P1 est reconnue. Celle-ci permet alors de remonter à P2 et à la lettre représentée par le couple (P1, P2). Au cours de ce balayage, on scrute la position d'un minima et d'un maxima chaque fois que le code directionnel "3" est rencontré (le code 3 correspond à un passage entre un maxima et un minima dans le tracé). Cette opération permet d'extraire une primitive P'2 associée localement au tracé. Le résultat de la comparaison de P'2 à P2 définit un coefficient d'ambiguïté dont est affectée la lettre probable. Après validation, la somme du coefficient d'ambiguïté et du coefficient de vraisemblance, déterminée par le modèle de dégénérescence, donne un poids de reconnaissance à toute lettre identifiée dans le tracé.

III-3 VERIFICATION (OU VALIDATION) DES LETTRES

Au fur et à mesure que les lettres sont détectées, nous recherchons une correspondance entre celles-ci et les noeuds du dictionnaire. Si la corres-

pondance est possible, la lettre est retenue sinon elle est rejetée. La lettre retenue permet de définir un point de segmentation, dans le tracé analysé, à partir duquel l'identification de la lettre suivante sera tentée. Quand une branche est entièrement parcourue dans le dictionnaire, un mot est alors reconnu. Le dictionnaire permet ainsi de réduire l'explosion combinatoire des lettres identifiées. Autrement dit, il se comporte comme une table de vérité qui n'autoriserait que les associations de lettres conduisant aux mots du vocabulaire enregistré.

III-4 OPERATION DE DECISION

L'analyse d'un tracé aboutit à deux types de listes de caractères :

- les listes terminées, soit par une lettre impossible à détecter, soit par une lettre identifiée mais rejetée par le dictionnaire. Ces listes ne correspondent à aucun mot candidat.
- les listes correspondant aux mots candidats.

Dans les listes de candidats, chaque mot réalise un score calculé d'après la somme des poids de reconnaissance attribués à chaque lettre identifiée. Le candidat dont le score est le plus grand est alors retenu.

IV - CONCLUSION

Le système est implanté en Pascal sur un micro-ordinateur APPLE III. Avec un dictionnaire de 100 mots et un apprentissage sur 5 scripteurs, il permet des taux de reconnaissance de 95%, avec des temps de traitement ne dépassant pas la dizaine de secondes. Ceci donne une vitesse de reconnaissance beaucoup plus grande qu'une méthode de programmation dynamique implantée sur le même système.

L'implantation en cours sur un micro-ordinateur de type IBM-PC permettra de diviser par 8 les temps de traitement et d'augmenter sensiblement la taille du dictionnaire. De plus, le taux de reconnaissance actuel peut être amélioré par suite d'un apprentissage permanent des écritures propres à chaque scripteur. Cette méthode s'avère être un bon outil d'analyse de tracé en vue de leur reconnaissance. Elle aboutit à un système simple dont les performances sont satisfaisantes.

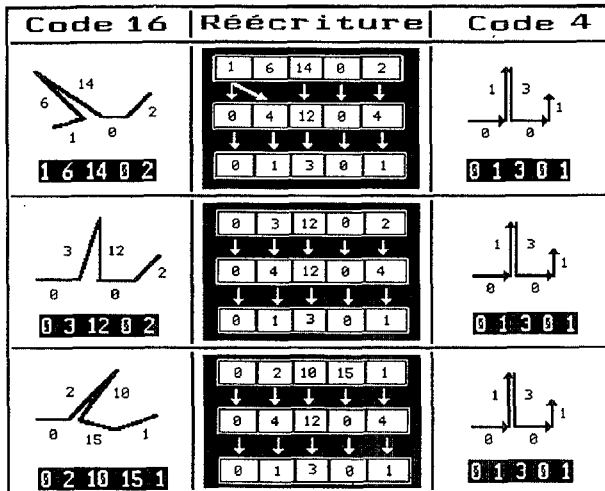


Figure 1 : Exemple illustrant la "réécriture" de 3 codes "16 directions" représentant la lettre *i* en un code "4 DIRECTIONS" invariant par translation et homothétie, et tolérant des inclinaisons de $\pm 45^\circ$.

BIBLIOGRAPHIE

- (1) FRISHOPF LS., HARMON L.D.
"Machine Reading of Cursive Script" - Information Theory, Symposium on Information Theory-London 1960. C. CHERRY ed. Washington: Butterworths, 1961, P. 300-316.
- (2) MERMELSTEIN P., EDEN M.
"Experiment on Computer Recognition of Connected Handwritten Words." - Information and Control, Vol 7, juin 1964, P. 255-270.
- (3) EHRICH R.W., KOEHLER K.J.
"Experiment in the Contextuel Recognition of Cursive Script." - I.E.E.E. Trans. Comput, Vol. C 24, fev 1975, P. 182-194.
- (4) A.S. FOX and C.C. TAPPERT
"On-Line External Word Segmentation for Handwriting Recognition." - Proceeding of the Third International Symposium on Handwriting Computer Application - Montreal, Canada, July 1987, P. 53-55.
- (5) EARNEST L.D.
"Machine Reading of Cursive Script." - IFIP Congress 1962, Amsterdam: The Netherhands, North-Holland, 1963, P. 462-466.
- (6) MILLER G.M.
"Real Time Classification of Handwritten Script Words." - IFIP Congress 1971, Amsterdam. North Holland Publishing Company, 1972, P. 218-223.
- (7) LORETTE G.
"Méthode de traitement automatique de points et de lignes en reconnaissance des formes - Application à l'analyse de scènes, à la reconnaissance de signatures et de mots manuscrits." - Thèse de Doctorat d'Etat à l'Université Paris Val-de-Marne, 1984.
- (8) CHEN C., LORETTE G., GAUDAIRE M.
"Hierarchical Primitives Matching - A Structural Method for Cursive Script Recognition." - 4th SCIA, Trondheim, June 1985.
- (9) BERTHOD M.
"Expérimentation sur l'échantillonnage de tracés manuscrits en temps réel." - 1er Congrès AFCET-IRIA, Reconnaissance de Formes et Traitement des Images, 1978, P. 560-567.
- (10) BERTHOD M., JANCEN P.
"Le prétraitement des tracés manuscrits sur une tablette graphique." - 2ème Congrès AFCET-IRIA, Reconnaissances des Formes et Intelligence Artificielle, 1979, P. 195-209.
- (11) FREEMAN H.
"Computer Processing of Line Drawing Images." - Computing Surveys, Vol. 6, N°1, march 1974, P. 57-97.
- (12) BURR D.J.
"A Normalizing Transform for Cursive Script Recognition." - 6th IJCP, Munchen, Germany, P. 1027-1030, oct. 1982.