Segmentation et reconnaissance "en-ligne" de mots manuscrits

Sophie Bercu, Bernard Delyon

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Segmentation et reconnaissance "en-ligne" de mots manuscrits

Sophie BERCU
Bernard DELYON

N° 1858
Février 1993

PROGRAMME 4

Robotique, Image
et
Vision

IRISA

INSTITUT DE RECHERCHE
EN INFORMATIQUE
ET SYSTÈMES ALÉATOIRES

Campus Universitaire de Beaulieu
35042 RENNES CEDEX FRANCE
Tél. (33) 99 84 71 00 - Télex : UNIRISA 950 473 F
Télécopie : (33) 99 38 38 32

SEGMENTATION ET RECONNAISSANCE « EN-LIGNE » DE MOTS MANUSCRITS

Sophie BERCU, Bernard DELYON

IRISA/INRIA,
Université de Rennes 1,
Campus de Beaulieu 35042, Rennes Cedex,
FRANCE

Publication Interne n° 700 - Février 1993 - 32 pages - Programme 4

Résumé

La reconnaissance de l'écriture cursive manuscrite est à l'heure actuelle en pleine expansion. La première partie de ce rapport donne une présentation générale de ce domaine de recherche qui comprend deux modules : le prétraitement qui code le signal écrit et la reconnaissance de mots.
Dans notre étude, le premier module est abordé à l'aide d'une nouvelle description simple et robuste de l'écriture cursive en termes de boucle, pic, arc et sens du tracé. Cette description qui tient compte des mécanismes de l'écriture permet d'intégrer un grand nombre de variantes. Le prétraitement consiste alors, après un lissage et un filtrage du signal initial, à fournir une séquence de primitives (boucle, pic, arc) de la longueur du mot. Ce codage permet une réduction de la quantité d'information d'un facteur 100.
L'apprentissage et la reconnaissance des mots sont effectués à l'aide d'une modélisation statistique basée sur les chaines de Markov cachées. Un réseau stochastique combinant plusieurs sources de connaissance a priori décrit exhaustivement toute l'application (reconnaissance des mots un, deux, ..., dix). Le taux de reconnaissance obtenu en mode mono-scripteur est de l'ordre de 98.6%.

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE
(U.R.A. 227) UNIVERSITÉ DE RENNES I I.N.S.A. DE RENNES

INRIA
INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE
ET EN AUTOMATIQUE - UNITÉ DE RECHERCHE DE RENNES

On-line segmentation and recognition of handwriting words

Abstract

The cursive handwriting recognition is actually expanding. The first part of this report gives a general presentation of this research area that include two modules: the Preprocessing wich encodes the written signal and the word Recognition.
In our study, the first module introduces through a new simple and robuste description of the cursive handwriting in terms of loop, cusp, hump and the writing direction. This description that takes into account writing mecanisms, allows to integrate a large number of variations. The preprocessing consists then, after smothing and filtering the initial signal, to provide a sequence of primitives like loop, cusp and hump. This encoding reduces the information amount by a factor of one hundred.
The word training and recognition are done using a statistical model based on Hiden Markov Model. A stochastic network combining several a priori knowledge sources describes exhaustively the whole application (recognition of the words un, deux, ...,dix). The recognition rate reaches 98.6% in mono-scriptor mode.

Table des matières

1 Introduction **4**

2 Reconnaissance de l'écriture cursive **5**
2.1 Les hypothèses de travail 5
2.2 Le système de reconnaissance 5

3 Modélisation de l'écriture **8**
3.1 Le problème de la segmentation 8
3.2 Mécanisme de l'écriture 8
3.3 Description des lettres 10
3.4 Description des ligatures et des mots 10

4 Prétraitement **14**
4.1 Introduction 14
4.2 Le lissage 14
4.3 Le filtrage 15
4.4 Extraction des primitives 16
4.4.1 Détection des points d'intersection 16
4.4.2 Détection des points de rebroussement 16
4.4.3 Détection des points d'inflexion 17

5 Reconnaissance **20**
5.1 Chaîne de Markov cachée 20
5.2 Application à la reconnaissance de l'écriture 21

6 Mise en oeuvre **23**
6.1 Construction du réseau 23
6.2 Apprentissage et reconnaissance 24

7 Résultats préliminaires **25**
7.1 Apprentissage 25
7.2 Tests de reconnaissance et discussion 25

8 Conclusion **29**

1 Introduction

Pour faciliter davantage l'utilisation des ordinateurs, de nouvelles interfaces de dialogue homme-machine ont été créées [1] . Elles n'utilisent plus le clavier ni la souris mais un stylet relié à une tablette graphique permettant d'entrer des données sous forme de textes, de tableaux ou de graphiques. L'utilisation de cet outil nécessite des modules de reconnaissance *dynamique* qui doivent interpréter en temps réel ce que l'utilisateur trace sur la tablette. Dans le cadre de la reconnaissance de l'écriture manuscrite, le module de reconnaissance peut être mono-scripteur (un seul utilisateur), multi-scripteur (plusieurs utilisateurs) ou omni-scripteur. Les deux premiers cas nécessitent l'apprentissage de l'écriture des scripteurs concernés alors que dans le dernier cas, l'interface est utilisable par tout le monde et sans adaptation préalable.

Notre étude concerne exclusivement la reconnaissance de l'écriture cursive et plus particulièrement celle des mots dont certaines parties sont liées ou espacées.
La principale difficulté de la reconnaissance de l'écriture est due à la grande variabilité de l'écriture manuscrite. Celle-ci est en effet différente pour chaque individu et également pour un même individu sur des périodes différentes. Elle dépend aussi de l'environnement de travail (instrument utilisé et surface d'acquisition). Cependant, l'homme est capable de lire presque tous les documents manuscrits écrits dans sa langue d'origine. Pourquoi? Il est difficile de répondre, le contexte doit sûrement nous aider ainsi que des repères particuliers que nous fabriquons... En tout cas, nous le faisons inconsciemment par l'intermédiaire de notre cerveau. Malheureusement, la capacité des ordinateurs est fort différente de celle des cerveaux humains. Il faut donc, trouver des chemins détournés des notres pour réussir à faire lire automatiquement un ordinateur.

A l'heure actuelle, les systèmes les plus performants restreignent le problème de la reconnaissance de l'écriture en se limitant à la reconnaissance de lettres séparées et souvent écrits en caractères batons. Bien sûr, ce type de systèmes n'est pas satisfaisant pour rendre les interfaces de traitement de textes totalement conviviales.

Le but de notre travail est de concevoir et de réaliser un système de reconnaissance automatique de mots d'écriture manuscrite. Pour cela, une étude préalable sur la génération de l'écriture telle qu'elle est enseignée aux enfants va nous permettre de comprendre le processus de l'écriture et d'y adjoindre une modélisation convenable.

Après la présentation de la méthode générale de reconnaissance de l'écriture et des hypothèses de travail, nous proposons dans la première partie de cette étude une nouvelle modélisation de l'écriture qui s'appuie sur les mécanismes liés à la formation des lettres, des liaisons entre lettres et des mots. Dans la seconde partie, nous décrivons le prétraitement effectué, puis la mise en oeuvre de la segmentation.

2 Reconnaissance de l'écriture cursive

2.1 Les hypothèses de travail

Notre étude concerne la reconnaissance « en-ligne » de l'écriture, c'est-à-dire, acquise dynamiquement à l'aide d'une tablette graphique à digitaliser. Celle-ci possède une résolution de 2048×2048 points et échantillonne à une vitesse de 170 points par seconde. Elle est posée sur un écran EGA qui permet de visualiser le tracé instantanément. Un temps d'adaptation est nécessaire au scripteur pour pouvoir écrire à peu près correctement. Malgré tout, le tracé obtenu n'est pas parfaitement lisse. Le bruit est provoqué par le scripteur mais aussi par le matériel et la transmission des données. Bien que les constructeurs tentent de faire des efforts dans ce sens, le signal que nous obtenons est toujours fortement bruité.
Actuellement, les mots doivent être écrits en lettres minuscules mais sans autre contrainte. Selon la classification de [2], l'écriture peut être soit :

1. en caractères séparés scripts
2. en script pur
3. en cursif pur
4. en un mélange de script et de cursif

D'autre part, nous imposons une contrainte faible sur l'orientation du tracé. Celle-ci ne doit pas varier de plus ou moins 45 degrés par rapport à l'horizontale mais les lettres peuvent être inclinées.

2.2 Le système de reconnaissance

L'acquisition des mots s'effectue en écrivant sur une tablette reliée à un ordinateur. Un signal $(x(t), y(t))_{t \in N}$ correspondant aux coordonnées des points successifs du tracé et régulièrement espacés dans le temps est transmis à l'ordinateur. Les levers de crayon sont aussi repérés. A partir de ce signal, le système (voir le schema figure 1) de reconnaissance de l'écriture se divise en deux parties distinctes :

1. prétraitement du signal écrit,
2. décodage de l'information extraite (reconnaissance du mot).

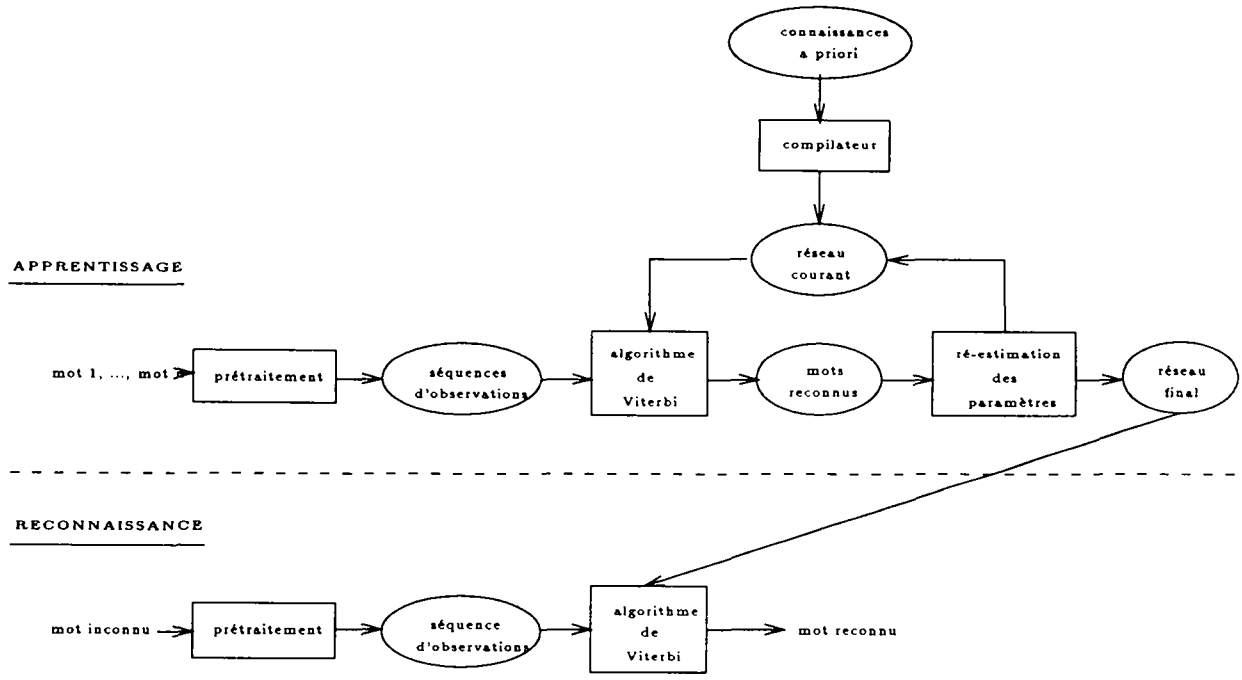


Figure 1 : synoptique général

Le rôle de la première partie consiste à effectuer un certain nombre de mesures sur le signal écrit pour en extraire des informations significatives. Le tracé subit deux traitements :

- Un lissage. Cette étape consiste à éliminer les bruits survenus lors de l'acquisition.
- L'extraction de primitives. Celle-ci nécessite le choix de descripteurs contenant des informations pertinentes et robustes permettant à la fois de discriminer au mieux les différents mots ou lettres et de reconnaitre une large variété de styles d'écriture. Ces descripteurs doivent être invariants par translation, taille, rotation de plus ou moins 30 degrés et indépendants les uns des autres.

De l'ensemble de points d'entrée, plusieurs types d'information peuvent être extraits [2, 3, 4, 5]. Ce sont essentiellement des informations de nature :

1. morphologique et topologique, renseignant sur la forme et les dimensions du tracé : boucles, segments de droite, rebroussements avec leur direction, leur longueur, largeur et hauteur propre ou relative, la courbure, la concavité, l'information de sortie de corps ...
   Ces informations sont extraites à partir des positions $x(t), y(t)$ et de la dérivée première dy/dx du tracé.
2. dynamique : ordre dans lequel les différentes composantes (tracé entre deux levers de crayon) ont été exécutées les unes par rapport aux autres. D'autre part, l'ordre des points à l'intérieur de chaque segment est également prépondérant pour déterminer le sens de la courbure.

Les positions successives et la dérivée seconde d^2y/dx^2 permettent d'obtenir ces informations.

3. cinématique : vitesse d'écriture obtenue à partir des dérivées par rapport au temps $dy(t)/dt, dx(t)/dt$. Le signal étant régulièrement échantillonné, la vitesse en un point du tracé correspond à la distance de ce point au point suivant.

4. ...

C'est l'application que l'on veut faire qui doit conditionner le choix des informations à extraire. Bien entendu, une erreur de jugement lors de cette extraction peut entraîner une erreur au niveau de la reconnaissance si le décodeur ne tient pas compte de l'hypothèse selon laquelle une erreur peut toujours survenir durant la phase d'extraction.

Ces informations sont ensuite envoyées à la partie reconnaissance qui, par comparaison avec des représentations connues, obtenues lors d'une phase d'apprentissage, détermine finalement le mot tracé.
Il existe deux principales approches pour la partie reconnaissance :

1. L'approche analytique : elle utilise une segmentation a priori du tracé en unités de la taille d'une lettre en général. Chaque unité est identifiée puis les lettres sont regroupées pour former les mots.
   Dans ce cas, l'apprentissage s'effectue uniquement sur les lettres. Cette méthode autorise donc un vocabulaire de travail étendu. Toutefois, la segmentation des mots en lettres est une opération complexe et sujette à des erreurs qui sont fatales pour la reconnaissance des mots.

2. L'approche globale : elle peut être abordée de deux façons différentes. Soit les mots sont segmentés en primitives et la représentation globale du mot est donnée par la somme des informations obtenues sur les différents segments du tracé. Soit le mot à reconnaitre est caractérisé par son aspect global et reste intact en fin d'analyse.
   L'apprentissage est effectué sur des mots appartenant à un vocabulaire choisi.

Dans ce rapport, nous décrivons la partie prétraitement du système de reconnaissance en justifiant le choix des primitives utilisées. La modélisation de l'écriture manuscrite cursive que nous proposons pouvant être utilisée ultérieurement pour faire de la reconnaissance analytique ou globale avec segmentation. Nous n'avons abordé dans la suite que la seconde approche qui est, de par sa nature, plus tolérante aux déformations de l'écriture et donc plus performante pour un vocabulaire limité.

3 Modélisation de l'écriture

3.1 Le problème de la segmentation

Que ce soit dans le cadre de l'approche analytique ou de l'approche globale avec segmentation, le tracé est toujours découpé en unités graphiques délimitées par des points de segmentation. Il existe deux stratégies de segmentation :

1. Une segmentation externe. Elle est effectuée avant la reconnaissance. Des caractéristiques sont extraites des segments et analysées lors de la reconnaissance.

2. Une segmentation interne. Les unités sont segmentées et reconnues simultanément. Dans le cas où les unités sont les lettres de l'alphabet, le point initial du tracé est le premier point de segmentation. Le suivant est celui qui permet au reconnaisseur de découvrir une lettre entre ces deux points, et ainsi de suite jusqu'à la fin du tracé. Certains systèmes permettent cependant de remettre en question des décisions de segmentation [6].

Les points de segmentation peuvent être détectés aux levers de stylet, aux points où la vitesse verticale est minimale ou nulle, aux points d'ordonnée ou d'abscisse extrémale, aux points de forte courbure, de forte variation angulaire, d'inflexion, d'intersection...
Quelle que soit la stratégie adoptée, il subsiste toujours des problèmes. Dans le cas de la segmentation interne, ceux-ci sont localisés au niveau des liaisons entre les lettres. En fonction de l'écriture (cursive ou scripte), ces liaisons peuvent être importantes ou inexistantes. D'autre part, lorsque l'on fait de la reconnaissance aveugle (sans connaissance au niveau du contexte et au niveau lexical), la segmentation d'un mot en lettres est certes une opération naturelle, mais elle est extrêmement complexe car la segmentation en lettre n'est pas unique. Par exemple la lettre 'd' peut être segmentée en 'c' puis 'l', 'a' en 'c' et 'e'. Il existe beaucoup d'autres exemples, en particulier si l'on ne prend pas en compte le point sur la lettre 'i' parce qu'il a été omis ou parce qu'il n'est pas juste au dessus du 'i', tous les jambages montants peuvent être confondus avec le 'i'.
Une erreur lors de la segmentation peut entraîner une erreur au niveau de la reconnaissance. C'est pourquoi nous avons implementé une autre méthode de segmentation, qui nous semble plus robuste que celles déjà existantes en nous basant sur la génération de l'écriture.

3.2 Mécanisme de l'écriture

Pour écrire, nous appliquons à la pointe du stylo deux mouvements :

1. un mouvement de translation de la gauche vers la droite, dû au poignet et à l'avant bras,

2. un mouvement plus rapide de bas en haut, produit par certains doigts de la main.

La combinaison de ces deux mouvements, avec des vitesses différentes pour chaque composante, permet de former l'écriture occidentale. En fait, cette combinaison de mouvements permet de tracer des courbes arrondies et des segments de droites. Les courbes possèdent des points de rebroussement que nous appellerons des pics et des zones continues que nous appellerons des arcs. Lorsqu'une même courbe se rencontre à nouveau sans qu'il n'y ait eu d'interruption de tracé, nous l'appellerons une boucle. Sur une période du signal de la vitesse, des informations relatives aux passages par zéro d'une composante (horizontale ou verticale) de la vitesse par rapport à l'autre composante nous renseignent sur la forme du tracé correspondant à cet intervalle de temps [7]. La figure suivante montre l'allure des vitesses correspondantes aux trois formes de base: pic vertical, boucle verticale et arc concave:

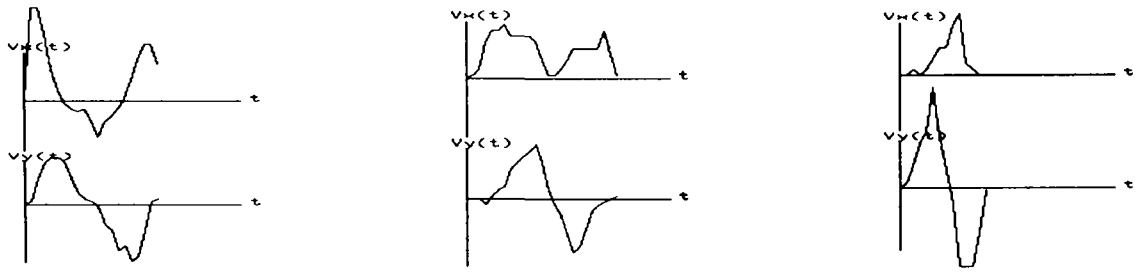


Figure 2 : Boucle: $V_x(t)$ change de signe avant et après un changement de signe de $V_y(t)$. Pic: $V_y(t)$ change de signe et $V_x(t)$ possède un minimum au même instant. Arc: seul $V_y(t)$ change de signe.

Les deux mouvements (de bas en haut et de gauche à droite) peuvent évoluer dans un repère variable en fonction des scripteurs.

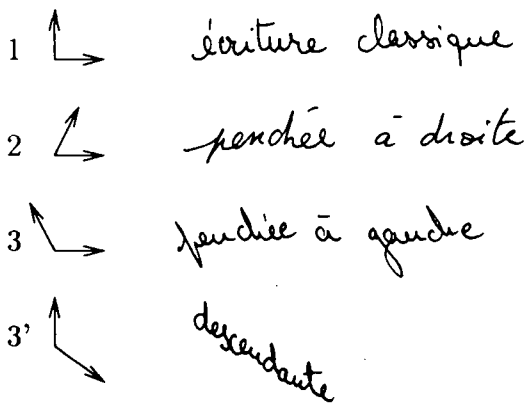


Figure 3 : repères d'écritures

On obtient aussi tous les repères provenant d'une rotation de ces repères. Il peut y avoir également des variantes à l'interieur d'un même mot. Par exemple, les lettres

sortant du corps du mot peuvent être écrites dans le repère 2 et les autres dans le repère 1 : mélange de repères

Toutes autres combinaisons peuvent évidemment se produire. La combinaison des deux mouvements (poignet/avant bras et main) produit donc une succession de boucles, d'arcs et de pics. Nous allons voir maintenant comment l'on peut décrire les lettres et les mots à partir de ces informations, de façon simple, unique et robuste.

3.3 Description des lettres

Nous nous interessons principalement aux lettres minuscules scriptes ou cursives et parfois scriptes et cursives à la fois.
Les lettres parfaitement bien tracées telles que nous les apprenons à l'école peuvent être segmentées en boucle, arc et pic :

 : arc . pic

 : boucle . arc . boucle

Cette segmentation amène à une description simple mais pas nécessairement unique. Les lettres a, d et q possèdent la même description, les lettres e et l, g et z également. Si on ajoute l'information de sens d'écriture (+ pour le sens trigonométrique et − pour le sens inverse) et de hauteur relative des primitives les unes par rapport aux autres, alors on peut distinguer les lettres a, d et q entre elles. De même, pour les lettres g et z.

 : arc^+ . pic^+

 : arc^+ . (pic^+, haut)

 : arc^+ . (pic^+, bas)

Seules les lettres e et l ne sont distinguables qu'en fonction du contexte. Dans le cadre des lettres isolées et parfaitement bien tracées, les caractéristiques citées sont pertinentes car elles permettent de discriminer ces lettres entre elles.
Malheureusement, pour les lettres imparfaites de l'écriture courante, beaucoup de variations peuvent intervenir. Nous allons donc utiliser une description stochastique nous permettant de prendre en compte la variabilité de l'écriture.

3.4 Description des ligatures et des mots

A partir de l'écriture apprise à l'école, dont l'enseignement est relativement commun à tous, le scripteur va progressivement au cours des ans transformer ces modèles parfaits en fonction de sa personnalité et en particulier pour augmenter sa vitesse d'écriture. Ceci se traduit à l'interieur des lettres mais aussi par l'apparition de liaisons entre ces lettres. Pour être plus efficace, on peut penser qu'il faut lever le stylo le moins possible mais certaines personnes préfèrent néanmoins conserver une écriture très segmentée. Les levers de stylo se font plutôt dans le but d'avoir un mouvement plus souple ; ce point de vue étant propre à chaque scripteur. Lors de

l'écriture des mots, les lettres subissent deux types de dégradations, celles-ci sont dues à :

1. la personnalité et à la vitesse d'écriture du scripteur,
2. l'influence des lettres adjacentes.

Le premier type de dégénérescence se produit à l'intérieur des lettres et provient d'une variation des vitesses horizontale et verticale du mouvement du stylo et d'une variation de la combinaison de ces vitesses. Par exemple, une écriture très « arrondie » est causée par un mouvement de transation horizontale plus lent par rapport au mouvement de bas en haut à l'inverse d'une écriture « pointue ». Ces dégradations peuvent transformer une boucle en pic, un pic en arc et vice versa, en fonction du déphasage entre les composantes horizontale et verticale de la vitesse de l'écriture. Les profils des vitesses verticales $V_y(t)$ montrés en figure 2 sont similaires pour les trois formes de base alors que les profils des vitesses horizontales sont distinctes. Plus on écrit rapidement et moins on a tendance à revenir en arrière, ce qui entraîne la disparition des boucles. Par contre cela traduit également une diminution de la lisibilité.

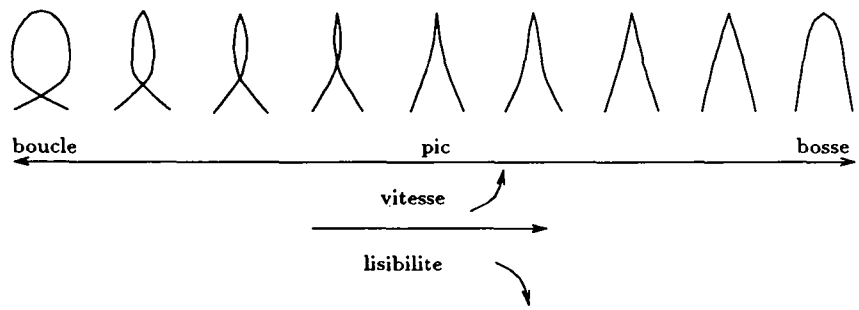


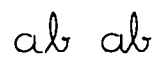
Figure 4 : transformation boucle, pic, arc

Le scripteur est donc obligé de moduler son écriture pour qu'elle soit rapide tout en restant lisible. Pour la lettre 'a', on peut rencontrer les variantes suivantes :

a : $\text{arc}^+ . \text{pic}^+$
a : $\text{arc}^+ . \text{boucle}^+$
a : $\text{arc}^+ . \text{arc}^-$

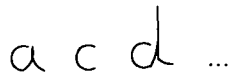
Pour tous les allographes d'une lettre, c'est-à-dire les différentes formes possibles issues d'un même modèle de lettre, il existe une description commune. Dans l'exemple ci-dessus, la première partie du 'a' cursif est toujours tracé grâce à un arc^+ et la seconde partie grâce à un pic^+ ou ses deux transformations possibles : boucle^+ et arc^-.

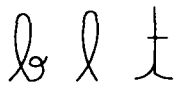
L'influence des lettres adjacentes se traduit par l'apparition :

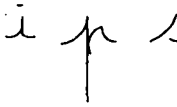
- d'une **amorce**. En début de mot ou juste après un lever de stylo au milieu d'un mot,
- d'une **terminaison**. En fin de mot ou juste avant un lever de stylo,
- d'une **ligature**. Entre deux lettres. La forme des ligatures dépend des lettres voisines. On peut distinguer trois classes principales :
  - invisible (lever de stylo ou lettres collées) 
  - neutre (un arc relie les deux lettres)
  - non neutre (la juxtaposition de deux lettres modifie leur forme) 

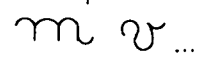
Ceci a pour conséquence de déformer le début et la fin des lettres en y supprimant ou ajoutant une boucle, un pic ou un arc. Les lettres sont partitionnées de deux façon, des groupes de lettres commençant de la même façon et des groupes de lettres finissant de la même façon.

DEBUT

Cd1 

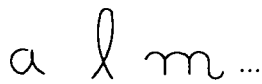
Cd2  Cd2' 

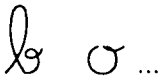
Cd3  Cd3' 

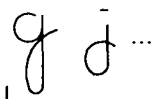
Cd4  Cd4' 

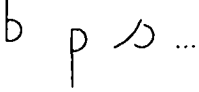
...

FIN

Cf1 

Cf2 

Cf3 

Cf4 

A chaque paire de lettres, on peut associer une forme de ligature en fonction des groupes d'appartenance des deux lettres. Par exemple,

- Si Cf1 est suivie de Cd1 alors il y a apparition d'une ligature non neutre qui a pour effet d'ajouter une boucle, un arc ou un pic au début de la lettre de la classe Cd1.
- Si Cf1 est suivie de Cd2 alors la ligature entre les deux lettres est soit invisible soit neutre.
- Si Cf2 est suivie de Cd3 alors il y a apparition d'une ligature non neutre qui a pour effet de supprimer l'amorce de la lettre de la classe Cd4.
- Si Cf4 est suivie de Cd1 alors il y a apparition d'une ligature non neutre qui a pour effet d'ajouter une bouble, un arc ou un pic au début et à la fin des lettres des classes Cf4 et Cd1 respectivement.

Ces règles devraient être toujours valables pour des écritures parfaitement soignées. Dans le cas contraire, des problèmes liés à l'omission ou l'ajout d'éléments de tracé élémentaires perturbent ces règles. C'est souvent le cas lorsque l'on écrit la paire 'br' où les petites boucles du 'b' et du 'r' cursifs sont confondus. Il faudra donc, dans la mesure du possible, tenir compte de ces aléas lors de la reconnaissance.

Les mots cursifs sont décris en termes de boucles, pics, arcs, levers de stylo, informations de sens et de hauteur relative à partir des lettres en y ajoutant des règles de dégradations.
La description que nous avons choisie est simple et robuste à la fois. Il s'agit maintenant de segmenter automatiquement les mots tracés sur la tablette puis les résultats de la reconnaissance nous permettront de valider cette segmentation.

4 Prétraitement

4.1 Introduction

Nous avons vu dans le chapitre précédent qu'une erreur au niveau de l'extraction des primitives pouvait être fatale pour la suite de l'application. Il est donc primordial de commencer par réduire les bruits d'acquisition. En effet, la vitesse d'échantillonnage de la tablette, l'imprécision des indications de la pointe du stylo, les tremblements de la main du scripteur, son inexpérimentation à utiliser ce genre d'outil ... sont autant de facteurs perturbant le tracé.

4.2 Le lissage

Cette étape consiste à supprimer tous les artefacts survenus lors du tracé sans toutefois modifier l'aspect initial du signal (voir exemples Figures 6 et 10).
Le lissage proposé [8] se compose de deux étapes: la première étape consiste à repérer les points mal échantillonnés et à les repositionner correctement. La seconde étape consiste à sous échantillonner le tracé.

Intéressons nous tout d'abord à la première étape. On supposera qu'un point est mal échantillonné lorsqu'il entraîne une brusque variation de la pente de la tangente au tracé.
Le principe en est le suivant: on calcule,

D_{i+1} : la distance entre le point i et son projeté orthogonal $pr_{i+1,f'(i)}(i)$ sur la droite de pente $f'(i)$ (pente de la tangente en i) et passant par le point $i+1$.

D_{i-1} : distance de i à $pr_{i-1,f'(i)}(i)$, projeté orthogonal de i sur la droite de pente $f'(i)$ et passant par $i-1$.

Si le $\sup(D_{i+1},D_{i-1})$ est supérieur à un seuil α, alors le point i est mal échantillonné. On le repositionne en son projeté, $pr_{i+1,f'(i)}(i)$ ou $pr_{i-1,f'(i)}(i)$, le plus proche. Le seuil α dépend de la résolution du capteur graphique.
Le tracé réel étant constitué d'une suite de points plus ou moins bruités, la tangente en un point i du tracé est calculée par la formule:

$$f'(i) = \frac{1}{2}(\alpha_{i-1,i+1} + \alpha_{i-2,i+2})$$

où $\alpha_{i,j}$ est la pente de la droite passant par les points i et j.
On peut noter que l'estimation de la tangente en un point est calculée independemment de la position de ce point mais qu'elle dépend de ses deux voisins suivants et précédents.
La figure ci-dessous montre le résultat d'un calcul de tangente et du lissage ainsi réalisé.

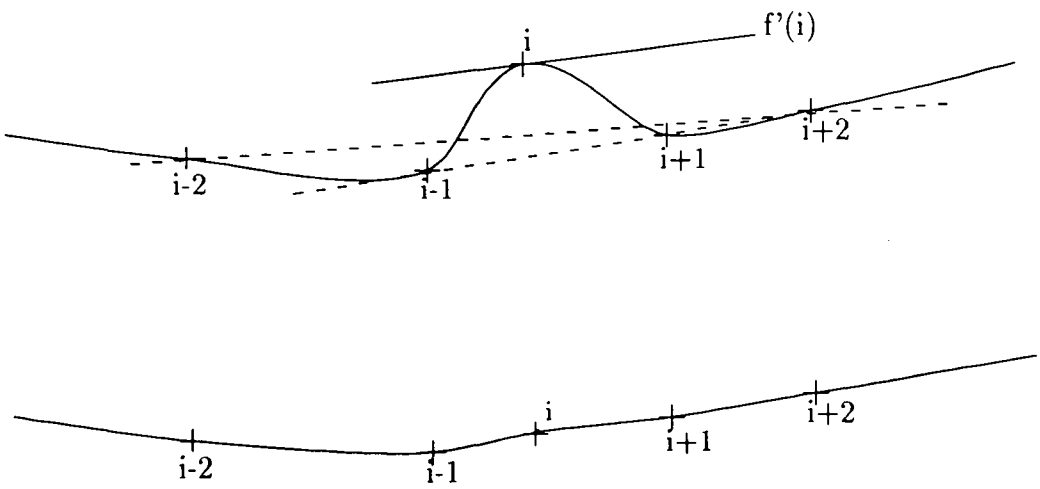


Figure 5 : Effet du lissage

On peut remarquer que ce traitement diminue bien les variations des pentes des demi-tangentes à droite et à gauche au point i.
Ce lissage peut entraîner une transformation de pic en arc. Cependant, de part la segmentation choisie, cela n'a aucune conséquence sur la suite du traitement. De plus, du fait du nombre important de points, de telles transformations sont rares.
Cette méthode de lissage possède également l'avantage important de conserver intacte la vitesse d'écriture.

L'écriture cursive étant en fait une succession de courbes plus ou moins arrondies, la seconde étape du lissage consiste à extraire de l'ensemble des points d'échantillonnage, des doublets tels que la courbure du tracé entre ces points soit relativement constante. Ces points vont être sélectionnés grâce à un filtrage.

4.3 Le filtrage

Ce filtrage vise deux objectifs (voir les Figures 7,8,11,12):

1. Approximer le tracé par une succession d'arcs de cercle et de segments de droite. Le nombre d'arcs de cercle et de segments de droite doit être le plus petit possible pour avoir un minimum d'informations à traiter simultanément. Paradoxalement, ce nombre doit être suffisament élevé pour que le tracé engendré soit le plus proche possible du tracé original.

2. Effectuer une présegmentation. Les arcs de cercles et les segments de droite représenteront le premier niveau de description. Nous verrons dans le chapitre suivant le second niveau de description représenté à partir du premier niveau par des pics, boucles et arcs. Les niveaux suivants seront les lettres et ligatures constituées de pics, arcs et boucles et le mot constitué de lettres et de ligatures.

Le filtrage utilisé [3, 9] permet de conserver après traitement les points anguleux, les zones de faible courbure assimilables à des droites et les zones de forte voir très forte courbure. Ces informations étant très importantes pour la reconnaissance de l'écriture.
Ce filtrage est un filtrage angulaire dont le principe consiste à rééchantillonner un point chaque fois que la pente de la tangente au tracé a varié d'un angle supérieur à ε par rapport à la pente de la tangente au point précédemment rééchantillonné. Ce traitement commence avec le rééchantillonnage du premier point du tracé et se termine avec celui du dernier point du tracé. La pente de la tangente au tracé est calculée comme précédemment.
L'échantillonnage ainsi réalisé est invariant par similitude et la densité linéaire des points échantillonnés est relativement proportionnelle à la courbure. Ainsi, les zones de forte courbure sont bien préservées. Si l'on ajoute à cet ensemble de points, les points du tracé correspondant à une inflexion suffisament importante, alors le sens de courbure est lui aussi conservé. Les points d'inflexion sont détectés aux points où la dérivée seconde change de signe, avec la contrainte supplémentaire que les trois points précédents ont une dérivée seconde de même signe, de même pour les trois points suivants. Cette procédure permet d'être robuste vis-à-vis du bruit. On obtient alors un ensemble de points de présegmentation entre lesquels on peut approximer le tracé par des arcs de cercle ou des portions de droite.
Dans [9], Berthod se contentait, pour l'approximation, du tracé des segments reliant les points de réechantillonnage. Néanmoins, les arcs de cercle permettent d'avoir une représentation plus fidèle du tracé original et de ce fait, d'avoir recours à un nombre inférieur de points de présegmentation. Grâce à ce filtrage, la quantité d'information (en espace mémoire) est réduite d'un facteur 100 par rapport aux données initiales.

4.4 Extraction des primitives

Nous avons donc choisi de segmenter les mots en un ensemble de primitives de types : boucles, pics et arcs (voir les Figures 9,13). Dans [5], on trouve le même choix de primitives mais les motivations ne sont pas les mêmes que les notres.
Ces trois formes sont reconnues respectivement à partir des points d'intersection, des points de rebroussement et des points d'inflexion du tracé et sont délimités par des points de présegmentation spécifiques issus de l'étape de filtrage.

4.4.1 Détection des points d'intersection

Pour chaque segment, on cherche un point d'intersection avec les n segments qui lui sont consécutifs. En pratique, on prend n=5.
La primitive boucle est délimitée par les points de présegmentation qui précèdent et qui suivent le point d'intersection.

4.4.2 Détection des points de rebroussement

Les points de rebroussement sont des points de discontinuité de la fonction f'. Si en un point, la différence entre les tangentes à droite et à gauche est supérieure à $5\pi/8$ alors un point de rebroussement est détecté.

La primitive pic commence au point de présegmentation qui précède le point de rebroussement jusqu'au point de présegmentation suivant.

4.4.3 Détection des points d'inflexion

Un point d'inflexion correspond à un changement de convexité dans le tracé.
La primitive arc est délimitée, soit par deux points d'inflexion, soit par un point de segmentation et un point d'inflexion, soit par deux points de segmentation.

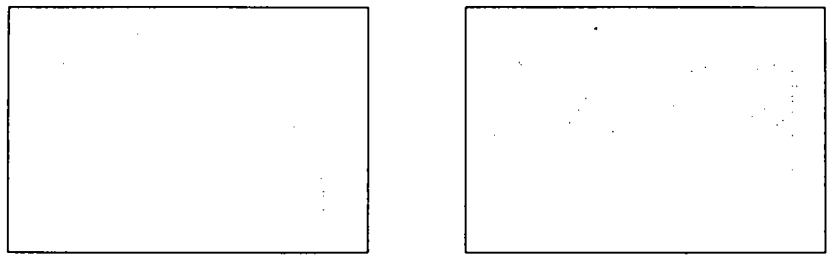


Figure 6 : a- signal initial, b- signal lissé

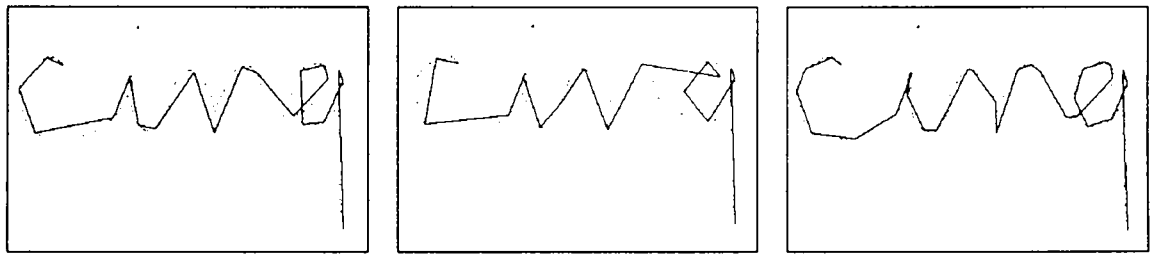


Figure 7 : filtrage angulaire $\epsilon = 3\pi/8$, $\epsilon = \pi/2$, $\epsilon = \pi/4$

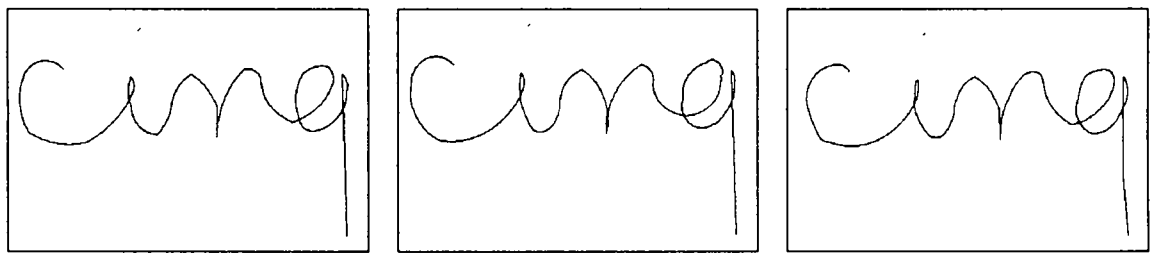


Figure 8 : approximation à l'aide d'arcs de cercle. $\epsilon = 3\pi/8$, $\epsilon = \pi/2$, $\epsilon = \pi/4$

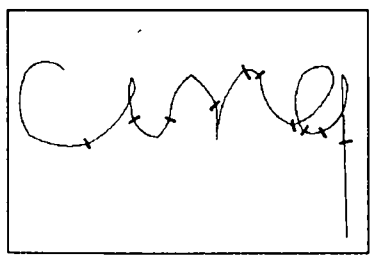


Figure 9 : segmentation finale : $\begin{matrix} \text{arc}^+ \text{ boucle}^+ \text{ arc}^+ \text{ arc}^- \text{ pic}^- \text{ arc}^- \\ \text{arc}^+ \text{ boucle}^+ \text{ arc}^+ \text{ boucle}^+ \text{ arc}^+ \end{matrix}$

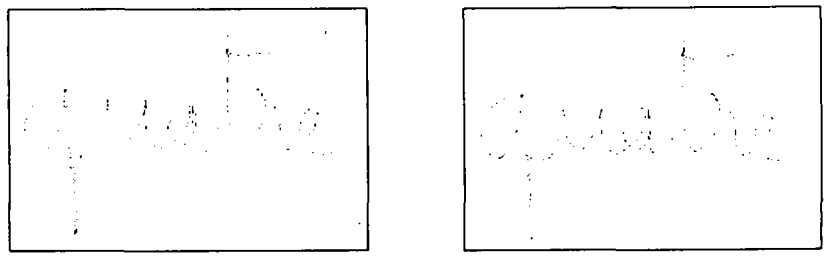


Figure 10 : a- signal initial, b- signal lissé

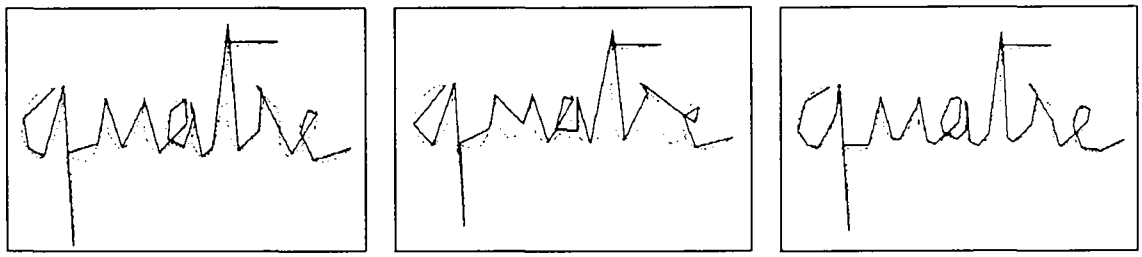


Figure 11 : filtrage angulaire $\epsilon = 3\pi/8$, $\epsilon = \pi/2$, $\epsilon = \pi/4$

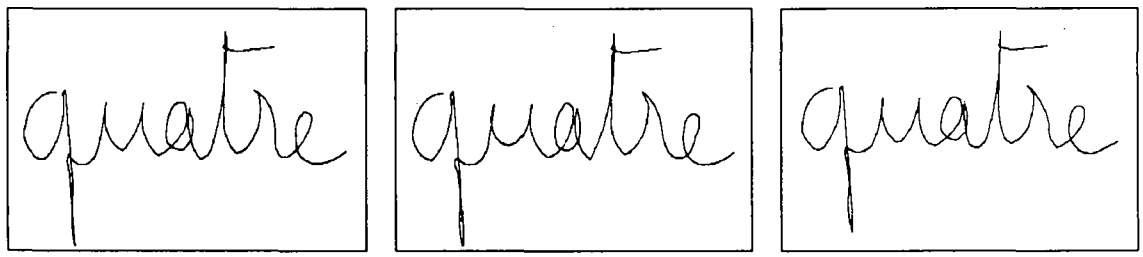


Figure 12 : approximation à l'aide d'arcs de cercle. $\epsilon = 3\pi/8$, $\epsilon = \pi/2$, $\epsilon = \pi/4$

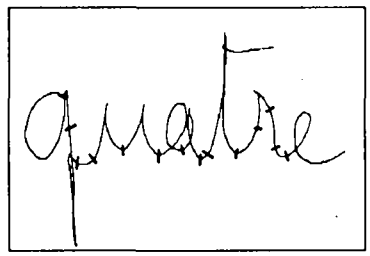


Figure 13 : segmentation finale : arc^+ arc^- $boucle^-$ arc^+ pic^+ pic^+ $boucle^+$ $boucle^+$ arc^+ pic^+ arc^+ pic^+ arc^- arc^+ $boucle^+$ arc^+

5 Reconnaissance

Pour tenir compte des fluctuations du tracé décrites précédemment, nous utilisons une modélisation statistique basée sur les chaînes de Markov cachées. Les modèles de Markov cachés (HMM) sont utilisés avec succès en reconnaissance de la parole où les taux de reconnaissance des nombres de 0 à 999 atteignent 95% en mode indépendant du locuteur [10, 11, 12]. En reconnaissance de l'écriture, des études utilisant des modélisations stochastiques commencent à apparaitre. Que ce soit dans la cadre de la reconnaissance « hors ligne » [13, 14, 15], ou dans le cadre de la reconnaissance « en ligne » des caractères chinois [16] ou des lettres isolées [17].
Avant de préciser l'application de ces modèles à la reconnaissance de l'écriture, nous allons définir les HMM.

5.1 Chaîne de Markov cachée

Une chaîne de Markov cachée est un double processus (X_t, O_t) [10] où

- $(X_t)_{t\geq 1}$ est une chaîne de Markov sur un espace d'états fini E, de cardinal N. Elle est définie par :
  - Sa loi initiale
  $$\Pi = \{\pi_i\}$$
  - Sa matrice de transition
  $$A = \{a_{ij}\} \quad i,j = 1...N$$

- $(O_t)_{t\geq 1}$ est le processus des observations (vecteurs des paramètres liés aux primitives) associé à chaque transition.
  Pour $e_1, ..., e_{t+1}$ dans E et $o_1, ..., o_t$ dans l'espace des paramètres, le processus vérifie la propriété suivante :
  $$\begin{aligned} & P(O_t = o_t | X_1 = e_1, \ ..., \ X_{t+1} = e_{t+1}, \ O_1 = o_1, \ ..., \ O_{t-1} = o_{t-1}) \\ = \ & P(O_t = o_t | X_t = e_t, X_{t+1} = e_{t+1}) \\ = \ & b_{e_t, e_{t+1}}(o_t) \end{aligned}$$
  où $b_{ij}(k)$ est la probabilité d'émettre l'observation k en effectuant la transition de l'état i à l'état j.
  Cette propriété traduit le fait que O_t est indépendant de $(O_s)_{s<t}$ et ne dépend que de (X_t, X_{t+1}).
  $$B = \{b_{ij}(.)\} \quad i,j = 1...N,$$
  est une famille de lois de probabilités.

On note $\lambda = (A, B, \Pi)$ un modèle de Markov caché.

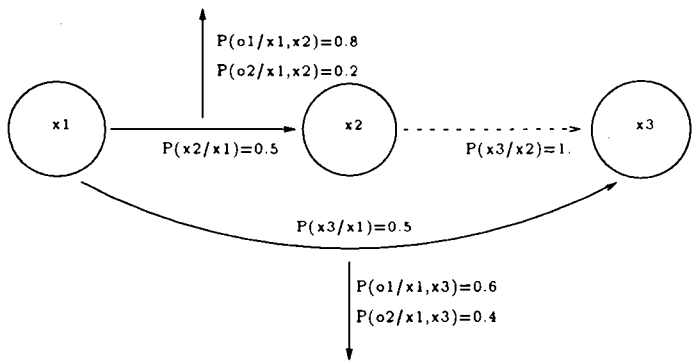


Figure 14 : Exemple de chaîne de Markov cachée à 3 états et 2 paramètres.

La chaîne (X_t) (c'est à dire la succession des états visités) n'est pas directement observable. Les chaînes que nous utilisons ne possèdent qu'un seul état initial et un seul état final. Elles sont dites « gauche-droite » car les seules transitions permises vont d'un état d'indice inférieur à un état d'indice supérieur. L'ordre d'écriture des segments est ainsi conservé. Une transition peut être vide lorsqu'il n'y a pas de vecteur O_t lié à cette transition. Ceci permet de prendre en compte des variations d'écriture comme par exemple la présence d'une ligature (transition non vide) ou l'absence d'une ligature (transition vide) entre deux lettres.

5.2 Application à la reconnaissance de l'écriture

Le problème de la reconnaissance de forme en général, se compose de deux étapes :

1. L'apprentissage. Un modèle est créé et estimé pour chaque unité (mot) à reconnaitre.

2. La reconnaissance. L'observation à reconnaitre est comparée aux modèles créés lors de la phase d'apprentissage.

Chaque mot du lexique est modélisé par un HMM. La structure des modèles (nombre d'états et transitions vides et non vides entre états) sont fixés a priori.

A l'issue de la phase de prétraitement, chaque segment est paramétrisé par deux caractéristiques : le genre et le sens d'écriture. $(O_t)_{t\geq 1}$ est alors un vecteur aléatoire discret à valeur dans $\{\text{boucle}^+, \text{boucle}^-, \text{arc}^+, \text{arc}^-, \text{pic}^+, \text{pic}^- \}^K$ où K est le nombre de segments de l'unité à modéliser ou à reconnaitre.

$E = \{e_1, ..., e_N\}$ où N est le nombre de segments maximum de tous les allographes de l'unité modélisée.
Toutes les trajectoires de (X_t), depuis l'état initial jusqu'à l'état final, correspondent à un allographe de l'unité associé au modèle, chaque transition de la chaîne étant associée à une fonction de densité de probabilité b_{ij}.

L'apprentissage consiste, à partir d'un échantillon de l'unité considérée et d'un modèle initial, à calculer à l'aide d'un algorithme itératif les paramètres (lois de transition de X_t, lois d'émission des symboles) du modèle. Soit $O^* = \{O^1, ..., O^R\}$ R écritures de l'unité à modéliser et θ_n un ensemble de paramètres du modèle λ. Les algorithmes d'apprentissage assurent, à chaque itération, l'augmentation de la probabilité d'émission des observations :

$$\prod_r P(O^r|\lambda(\theta_{n+1})) \geq \prod_r P(O^r|\lambda(\theta_n))$$

Etant donné une suite d'observations $(o_t)_{1\leq t\leq T}$, la reconnaissance consiste à trouver le modèle $\hat{\lambda}$ qui maximise $P_\lambda(O_1, ..., O_T)$ au sens du maximum de vraisemblance. Ce problème revient à déterminer un chemin optimal parmi tous les modèles. Ce chemin est obtenu grâce à l'algorithme de Viterbi qui permet de calculer de façon simple et rapide l'argument de

$$\max_{e_1 ... e_T} P_\lambda(X_1 = e_1, ..., X_T = e_T, O_1 = o_1, ..., O_T = o_T)$$

où T est le nombre de segments du mot et λ le modèle associé.

6 Mise en oeuvre

L'application visée est la reconnaissance des nombres de un à dix écrits littéralement. Elle est décrite de façon exhaustive par un réseau stochastique.
Les modèles correspondant à tous les mots du lexique sont connectés par un état initial et un état final communs, afin de former un réseau unique représentant toute l'application.

6.1 Construction du réseau

L'application est décrite de façon hiérarchique. Chaque niveau de connaissance est modélisé :

- Une description lexical est faite au niveau le plus élevé.
- Au second niveau, le lexique est décrit à l'aide de lettres.
- Au niveau le plus bas, chaque lettre, ligature et pseudo-lettre définies en fonction du contexte, est associée à un modèle de Markov élémentaire dont la structure est déterminée par des connaissances a priori sur la forme des lettres et où les observations sont celles qui proviennent du prétraitement.

La structure d'un modèle peut être commune à plusieurs lettres mais les fonctions de densité de probabilité sont bien sûr différentes. Par contre, une même distribution peut être associée à plusieurs transitions.

Une lettre commune à plusieurs mots est décrite par un unique modèle. Un modèle différent est associé à une même lettre en fonction de sa position dans le mot. Une lettre située en début, en fin ou en milieu d'un mot ne s'écrit pas de la même manière.

Les groupes de lettres définis dans le paragraphe 3 sont spécifiés et des règles de contexte concernant la forme des ligatures sont fournies. En fonction de la forme de deux lettres adjacentes, une règle est donnée. Par exemple :

(a . b) devient (a . (lig_inv + lig_neut) . b)
+ signifie 'ou', . signifie 'et'

Un modèle est associé à chaque ligature. Toutefois, si deux lettres sont trop fortement liées, une nouvelle lettre est définie.

Un compilateur traite toutes ces données pour fournir un unique modèle de Markov caché où tous les allographes de tous les mots du lexique sont représentés.

6.2 Apprentissage et reconnaissance

Tous les mots servant à l'apprentissage subissent le prétraitement décrit plus haut. Pour chaque mot, une séquence de primitives est trouvée. L'algorithme de Viterbi est appliqué sur un réseau initial pour chacunes des séquences de primitives. Toutefois, la recherche d'un chemin optimal se fait uniquement sur la portion du réseau correspondant au mot à apprendre.
Les paramètres du réseau sont estimés à partir des taux de passages obtenus entre deux états et des séquences de primitives. Cette ré-estimation entraîne la formation d'un nouveau réseau. A partir de ce réseau et toujours des mêmes observations, une nouvelle itération de l'algorithme de Viterbi permet la ré-estimation de paramètres « optimaux ».

Le mot à reconnaitre subit également le prétraitement. Ensuite, l'algorithme de Viterbi appliqué au nouveau réseau fournit la séquence d'états la plus probable. Cette séquence permet alors de désigner un des mots du dictionnaire.

7 Résultats préliminaires

Nous avons évalué notre réseau sur un dictionnaire de 10 mots : un, deux, ... ,dix où 16 lettres différentes sont présentes.

7.1 Apprentissage

La structure du réseau et l'association des lois d'émission des observations aux transitions sont décrites complètement par des connaissances a priori. Au dernier niveau de la description du modèle (cf paragraphe 6.1) les connaissances sur l'écriture sont de deux types. Soit elles sont pragmatiques et basées sur toutes les informations données dans le chapitre 3. Soit, elles sont liées à la connaissance du bruit propre à la tablette à digitaliser utilisée. Ce bruit extrêmement important entraîne des changements de sens du tracé involontaires de la part du scripteur (surtout s'il n'a pas suffisamment d'entraînement). Au niveau de la segmentation, cela se traduit par des arcs supplémentaires que l'on ne peut pas repérer avec le prétraitement mis en oeuvre et par des arcs dont le sens d'écriture n'est pas celui attendu. Finalement, 20 modèles de Markov cachés et 50 fonctions de densité de probabilité d'émission des observations sont nécessaires pour décrire toutes les lettres et types de ligature intervenant dans le lexique choisi. Deux modèles différents sont associés à la lettre 's' en fonction de sa position dans le mot. En effet, la ligature entre cette lettre et une autre entraîne la formation d'une boucle supplémentaire à la fin du 's' qui n'apparait pas si celui-ci se trouve en fin de mot. De même, deux modèles sont associés à la lettre 'q'.
Le compilateur génère ensuite un réseau constitué de 213 états et de 492 transitions (dont 128 transitions vides).

L'apprentissage est réalisé sur 20 banques comprenant chacune les dix mots du lexique et écrites par une seule personne.

7.2 Tests de reconnaissance et discussion

La reconnaissance effectuée est monoscripteur car les tests ont été réalisés sur l'écriture de la personne ayant écrit les banques d'apprentissage. 29 banques (soit 290 mots) ont servi pour faire la reconnaissance.
Pour évaluer l'influence de l'initialisation et de l'apprentissage sur les performances du modèle, deux expériences ont été menées. La première consiste à prendre à l'initialisation des lois uniformes pour toutes les fonctions de densité d'émission des observations et à mesurer les taux de reconnaissance après chacune des itérations d'apprentissage. La seconde expérience consiste à initialiser les paramètres des fonctions de densité à partir des connaissances décrites précedemment.
Ces expériences permettent également d'évaluer la pertinence des primitives choisies. Les résultats des taux de reconnaissance sont donnés à l'aide des matrices de confusions suivantes :

Lois uniformes :

%	1	2	3	4	5	6	7	8	9	10
1	82.7					3.4				13.8
2		100								
3	6.9		72.4			10.3		3.4	3.4	3.4
4				100						
5					100					
6						79.3	20.7			
7						17.2	82.7			
8					3.4			96.5		
9								6.9	93.1	
10	3.4									96.5

1 itération, taux global : 90.3%

%	1	2	3	4	5	6	7	8	9	10
1	79.3					3.4				17.2
2		100								
3			79.3			10.3		3.4	3.4	3.4
4				100						
5					100					
6	3.4					96.5				
7						13.8	86.2			
8								100		
9								6.9	93.1	
10										100

2 itérations, taux global : 93.4%

%	1	2	3	4	5	6	7	8	9	10
1	82.7									17.2
2		100								
3	10.3		86.2					3.4	3.4	
4				100						
5					100					
6		3.4				96.5				
7							96.5			3.4
8								100		
9								3.4	96.5	
10										100

3 itérations, taux global : 95.8%

%	1	2	3	4	5	6	7	8	9	10
1	82.7									17.2
2		100								
3	3.4		89.6					3.4	3.4	
4				100						
5					100					
6		3.4				96.5				
7							96.5			3.4
8								100		
9								6.9	93.1	
10										100

4 itérations, taux global : 95.8%

%	1	2	3	4	5	6	7	8	9	10
1	82.7									17.2
2		100								
3			93.1					3.4	3.4	
4				100						
5					100					
6		3.4				96.5				
7							96.5			3.4
8								100		
9								3.4	96.5	
10										100

5 itérations, taux de reconnaissance global : 96.5%

Au dessus de 5 itérations les taux de reconnaissance ne varient plus.

Lois a priori :

%	1	2	3	4	5	6	7	8	9	10
1	93.1									6.9
2		100								
3			93.1		6.9					
4				100						
5					100					
6						100				
7							100			
8								100		
9									100	
10										100

A la première itération le taux de reconnaissance global est de 98.6% $\pm$ 1.3% au niveau de confiance 95%. Ici, le taux maximal est atteint dès la première itération.

Discussion

Ces deux expériences permettent de vérifier les propriétés liées à l'algorithme d'apprentissage [12] utilisé.
La première expérience montre que d'une itération à l'autre, le taux de reconnaissance augmente pour atteindre le maximum de 96.5%.
Le seconde expérience montre que en choisisant une bonne initialisation pour les paramètres du modèle, la convergence vers le modèle optimal peut être très rapide (1 itération dans notre cas) et meilleure que dans le cas d'une initialisation arbitraire. En effet, dans la première expérience, l'algorithme d'apprentissage a été piégé dans un maximum local.

En ce qui concerne la reconnaissance, les erreurs les plus fréquentes se produisent pour les mots 'un' et 'trois'. 'un' étant confondu avec 'dix' et 'trois' avec 'cinq', 'huit', et 'neuf'. Bien que visuellement ces mots nous semblent très distincts, leur confusion est très compréhensible lorsque l'on regarde les séquences d'observations obtenues après le prétraitement. Les séquences d'observations pour 'u' et 'di' étant identiques : arc^+ . $boucle^+$. arc^+ . $boucle^+$. Cette exemple montre que la description effectuée manque de précision. En ajoutant des renseignements supplémentaires sur la forme, l'orientation et les dimensions relatives des primitives, nous devrions pouvoir encore améliorer les taux de reconnaissance.

Un dernier test, donné uniquement à titre indicatif, a été effectué sur 4 banques provenant de deux autres scripteurs. Avec l'apprentissage précedent, on obtient la matrice de confusions suivante :

%	1	2	3	4	5	6	7	8	9	10
1	100									
2		75		25						
3			25	25				25	25	
4				100						
5					100					
6		25				50	25			
7							75	25		
8								100		
9								25	75	
10		25								75

taux de reconnaissance global : 77.5%.

Le nombre de banques est insuffisant pour que les résultats soient significatifs et les erreurs proviennent surtout de la mauvaise qualité des signaux obtenus après l'acquisition. Les scripteurs n'étant pas habitués à écrire sur une tablette, le tracé est très hésitant et bruité.

8 Conclusion

Le prétraitement effectué et le choix des primitives semblent, d'après les résultats obtenus, être bien adaptés au problème de la reconnaissance de l'écriture manuscrite.

Le lissage proposé donne de bons résultats visuels et ne perd pas la dynamique du tracé.

Le filtrage atteint bien le but désiré qui est de réduire au maximum la quantité d'information tout en conservant les caractéristiques significatives de l'écriture.

La segmentation du tracé en termes de boucles, arcs et pics est à la fois simple et robuste. Elle permet de comparer des écritures d'inclinaisons et de tailles variables, ce qui est indispensable lorsque l'on veut faire de la reconnaissance omni-scripteur. Enfin, elle permet une utilisation simple et structurée des modèles de Markov cachés.

Pour améliorer davantage le taux de reconnaissance, nous sommes en train d'extraire d'autres informations significatives telles que la détection des hampes et des jambages et nous essayons de mieux caractériser la forme des boucles, arcs et pics.

Enfin, la topologie du réseau devra être améliorer en tenant compte davantage de connaissances pragmatiques et ainsi permettre d'utiliser au mieux les capacités des modèles stochastiques.

References

[1] Higgins C., Ford M. "Stylus driven interfaces - The electronic paper concept." *First International Conference on Document Analysis and Recognition*, Saint-Malo, pp 853-862, octobre 1991.

[2] Tappert C., Suen C., Wakahara T. "The State of The Art in On-line Handwriting Recognition." *IEEE transactions on pattern analysis and machine intelligence*, vol.12, No.8, pp 787-808 august 1990.

[3] Oulhadj H. "Des primitives aux lettres une méthode structurelle de reconnaissance avec un apprentissage continu." *Thèse de Doctorat*, Univ Paris Val-de-Marne, Créteil, 1990.

[4] Parizeau M. "Segmentation hiérarchique de l'écriture manuscrite cursive." *Rapport de stage franco-québécois*, IRISA-Scribens, 1990.

[5] Higgins C., Ford D., "On-line Recognition of Connected Handwriting by Segmentation and Template Matching", *ICASSP-92*, vol. 3, pp 200-203, 1992.

[6] Menier G., Lorette G., "Segmentation et reconnaissance en ligne d'écriture cursive à l'aide de plusieurs niveaux d'information contextuelle", *CNED-92*, pp 318-324, 1992.

[7] Schomaker L., Thomassen A., Teulings H., "A Computational Model of Cursive Handwriting", *Computer Recognition and Human Production of Handwriting*, pp 153-177, 1989.

[8] Bercu S., Delyon B., Lorette G., "Segmentation pour une Méthode de Reconnaissance d'Ecriture Cursive "En-ligne", *CNED-92*, pp 144-151, 1992.

[9] Berthod M., Jancen P. "Le prétraitement des tracés manuscrits sur une tablette graphique." *2ème congrès AFCET-IRA, Reconnaissance des formes et Intelligence Artificielle*, pp 195-209, 1979.

[10] Rabiner L. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition" *Proceedings of the IEEE*, vol. 77, 1989.

[11] Lee K., Hon H., Reddy R., "An Overview of the SPHINX Speech Recognition System", *IEEE Trans. on Acous.*, vol. 38, n. 1, pp 35-45,1990.

[12] Jouvet D., "Reconnaissance de mots connectés indépendamment du locuteur par des méthodes statistiques" *Thèse de Doctorat*, ENST, Lannion, 1988.

[13] Kundu A, Bahl P., "Recognition of Handwritten Script: A Hidden Markov Model Based Approach", *ICASSP-88*,pp 928-931, 1988.

[14] Vlontzos J., Kung S. "Hidden Markov Model for Character Recognition", *ICASSP-89*, pp 1719-1722, 1989.

[15] He Y., Chen M., Kundu A., "Handwritten Word Recognition Using HMM with Adaptive Length Viterbi Algorithm", *ICASSP-92*, vol. 3, pp 153-156, 1992.

[16] Hsieh C., Lee H., "A Probabilistic Stroke-based Viterbi Algorithm for Handwritten Chinese Characters Recognition", *ICPR*, vol. 2, pp 191-194, 1992.

[17] Farag R., "Word-level Recognition of Cursive Script". *IEEE Trans. on Comp.*, vol. C 28, n. 2, pp 172-175, 1979.

Liste des publications internes Irisa 1993

PI 694 MECANISMES D'ABSTRACTION DANS UNE REPRESENTATION DE LA CONNAISSANCE CENTREE OBJET (R.C.O.)
Stéphane LE PEUTREC, Sophie ROBIN
Janvier 1993, 40 pages.

PI 695 DETECTION DE SEQUENCES ATOMIQUES DE PREDICATS LOCAUX DANS LES EXECUTIONS REPARTIES
Michel HURFIN, Noël PLOUZEAU, Michel RAYNAL
Janvier 1993, 16 pages

PI 696 SEMI-UNIFIED CACHES
Nathalie DRACH, André SEZNEC
Janvier 1993, 18 pages.

PI 697 CONCEPTS ET PROBLEMES DE L'ALGORITHMIQUE REPARTIE
Michel RAYNAL
Janvier 1993, 18 pages.

PI 698 TEMPORAL PLANNER = NONLINEAR PLANNER + TIME MAP MANAGER
Eric RUTTEN, Joachim HERTZBERG
Janvier 1993, 18 pages.

PI 699 EVALUATION COMPARATIVE D'ALGORITHMES DE NORMALISATION DES SCHEMAS RELATIONNELS
Annie FORET, Placide FRESNAIS
Février 1993, 22 pages.

PI 700 SEGMENTATION ET RECONNAISSANCE "EN LIGNE" DE MOTS MANUSCRITS
Sophie BERCU, Bernard DELYON
Février 1993, 32 pages.

Unité de Recherche INRIA Rennes
IRISA, Campus Universitaire de Beaulieu 35042 RENNES Cedex (France)

Unité de Recherche INRIA Lorraine Technopôle de Nancy-Brabois - Campus Scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 VILLERS LES NANCY Cedex (France)
Unité de Recherche INRIA Rhône-Alpes 46, avenue Félix Viallet - 38031 GRENOBLE Cedex (France)
Unité de Recherche INRIA Rocquencourt Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)
Unité de Recherche INRIA Sophia Antipolis 2004, route des Lucioles - B.P. 93 - 06902 SOPHIA ANTIPOLIS Cedex (France)

EDITEUR
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

ISSN 0249 - 6399