Segmentation incrémentale d'une base de mots multiscripteur en lettres

Incremental Letter–Segmentation of a Multiscriptor Database of Words

par Laurent DUNEAU, Bernadette DORIZZI
Institut National des Télécommunications
9, rue Charles Fourier, F-91011 Évry cedex

Résumé

Cet article présente un système de segmentation qui découpe automatiquement un ensemble de mots manuscrits en lettres. L'objectif de cette opération est de constituer une base d'apprentissage pour un système de reconnaissance en ligne de mots manuscrits. Les tracés de mots à segmenter ont été saisis sur une tablette à digitaliser puis convertis en une suite de vecteurs. Enfin, à chacun de ces tracés de mot, est associé le mot alphabétique correspondant. La description de notre système de segmentation est suivie d'un test expérimental portant sur le découpage en lettres d'une base de 10000 mots qui proviennent de 10 scripteurs différents.

Mots clés : Reconnaissance en ligne, écriture cursive, mots manuscrits, approche analytique, segmentation en lettres, apprentissage automatique.

Abstract

This paper presents a segmentation system that automatically splits into letters a set of handwritten words. This is made in order to built a learning base for an on-line recognition system dedicated to handwritten words. The drawn words to be segmented are entered on a digitizing tablet before being converted into a sequence of vectors. At last, the corresponding alphabetic word is associated to each of these drawn words. The description of our segmentation system is followed by an experimental test of letter segmentation applied on a database of 10000 words coming from 10 different scriptors.

Key words : *On–line recognition, cursive script, handwritten words, analytical approach, letter segmentation, automatic learning.*

1. Introduction

Nous nous plaçons ici dans le cadre de la reconnaissance analytique de mots anglais (pour limiter les problèmes causés par les accents et les cédilles) écrits sur tablette à digitaliser (entrée « en–ligne »). Le terme « analytique » signifie ici que l'on cherche d'abord à identifier les lettres des mots. L'identification des mots nécessite ainsi une modélisation des (tracés de) lettres, et l'utilisation d'une mesure du degré de « correspondance » entre des portions de tracé de mot et des modèles de lettres. Les systèmes de reconnaissance de mots décrits dans [Duneau.b], [Fujisaki], [Higgins], et [Morasso], par exemple, suivent cette approche. Les principaux systèmes proposés lors des décennies précédentes sont présentés dans [Tappert].

Les modèles de lettres sont, en général, déterminés par l'analyse statistique d'une base de tracés de lettres issus de mots. Celle–ci ne donne sa pleine mesure que si l'on dispose d'un ensemble suffisamment important de tracés de lettres. Un tel ensemble ne sera représentatif que s'il contient des lettres prises dans leur contexte, c'est à dire extraites d'un mot. Cela implique nécessairement une segmentation efficace et rapide d'un grand nombre de mots en lettres.

L'objet de notre étude est de réaliser un système qui effectue cette tâche de façon automatique et autonome (sans l'intervention d'un opérateur humain). Un tel système permet d'envisager la fabrication de bases comprenant plusieurs centaines de milliers de lettres, ainsi que l'adaptation automatique à une nouvelle écriture par la découverte de nouvelles sortes de lettres.

D'autres auteurs se sont penchés sur ce problème (voir [Teulings]). Les originalités de notre approche résident principalement dans le codage du tracé, la stratégie utilisée pour la segmentation des mots en lettres, le recours à deux niveaux de représentation pour le tracé du mot et surtout, l'introduction d'une mesure permettant d'estimer la « fiabilité » d'une référence de lettre.

Nous avons déjà réalisé un système effectif pour la segmentation de mots ([Duneau.a]) et les modifications apportées ici rendent ce système plus souple, tout en améliorant ses performances dans un contexte multiscripteur. Enfin, ce travail a donné lieu à des développements ultérieurs présentés dans [Duneau.b] et [Duneau.c].

Après une description générale de notre système de segmentation de mots, nous présenterons quelques préambules qui nous permettrons ensuite de le décrire en détail. La qualité de la segmentation sera testée en utilisant un système de reconnaissance de mots que nous présenterons. Enfin, des résultats expérimentaux permettront de juger l'intérêt de notre approche.

2. Principe

L'entrée de notre système est une base $\boldsymbol{B}$ contenant une liste de mots codés et étiquetés. Chacun des éléments de cette base est ainsi une suite $\boldsymbol{G_k}$ de vecteurs obtenue après codage du tracé d'un mot (voir § 3.1). Chaque suite de vecteur G_k est accompagnée de son **étiquette** $\boldsymbol{\Gamma_k}$. Γ_k est simplement la suite des lettres alphabétiques qui correspondent au mot codé G_k. La tâche à effectuer est donc, pour chaque élément de la base, de localiser dans G_k chacune des lettres $\boldsymbol{L_j}$ de son étiquette Γ_k, en déterminant la sous suite $(\boldsymbol{X_j})$ de G_k qui correspond à la lettre L_j.

Lorsque toutes les lettres d'un mot de B ont été localisées, celui ci est **segmenté**. On associe ainsi à B une liste σ qui contient les positions des lettres de chacun des mots segmentés de B. Cet ensemble σ est appelé **segmentation** de B. Le système décrit ici a donc pour but de calculer, pour une base B donnée, le meilleur ensemble σ possible, cette opération étant également nommée *segmentation* de B.

Comme le montre la figure 1, cette segmentation s'effectue en un certain nombre $\boldsymbol{N_p}$ de passes fixé à l'avance. A chaque passe i, le système dispose d'un ensemble $\boldsymbol{R_i}$ de références de lettres qui provient de la passe précédente, R_0 étant préalablement fourni. A partir de R_i, un module de reconnaissance tente de segmenter chacun des mots de la base. Les échantillons de lettre provenant des mots ainsi segmentés sont ensuite utilisés pour produire le nouvel ensemble R_{i+1}.

A chaque passe, le système découvre de nouvelles références de lettres, le nombre de références croît donc, ainsi que la proportion de mots segmentés. Le résultat de ce processus est la segmentation σ_{Np} obtenue lors de la dernière passe.

Ce qui précède ne décrit que les grandes lignes du système. Avant d'en présenter une description plus détaillée, nous avons jugé nécessaire d'introduire quelques préliminaires.

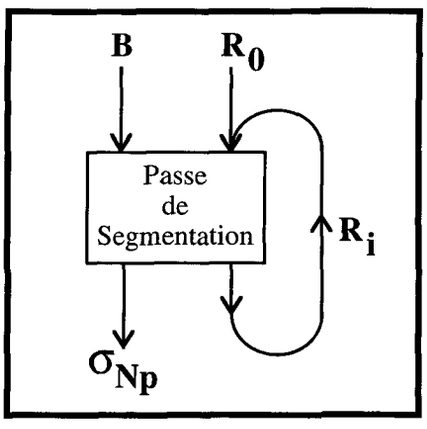


Fig. 1. – Principe du système de segmentation.

3. Préambules

Nous décrivons ici les éléments suivants :

1) Le codage du tracé d'un mot
2) La structure d'une fenêtre
3) Le calcul des références
4) L'activité d'une référence
5) La fiabilité d'une référence

3.1. CODAGE DU TRACÉ D'UN MOT

Le codage présenté ici transforme la suite des points acquis par la tablette à digitaliser en une suite de vecteurs $\boldsymbol{G}$ tout en conservant les informations pertinentes. Pour faire cela, la suite de points de chaque tracé de mot est d'abord découpée en sous–suites qui correspondent à des **portions de tracé**. Ce découpage du tracé repose en général sur une modélisation de celui–ci. Nous nous sommes inspirés de celle utilisée par Eden en 61 (cf.[Eden]). Les principales modélisations sont décrites dans [Plamondon].

Comme le montre la figure 2, les points de segmentation (représentés par des ronds) correspondent aux ruptures de tracé (levés de stylo et reprises de tracé) ainsi qu'aux extrema locaux suivant l'axe vertical. De plus, les portions de tracé qui présentent une hauteur trop importante sont systématiquement coupés en deux portions de même hauteur (ronds barrés dans la figure 2.).

Ce découpage sous–entend bien sûr que l'échelle du tracé est connue, ce qui se résout en imposant celle–ci au scripteur. Chacune des portions γ_i ainsi obtenue est ensuite convertie en un vecteur

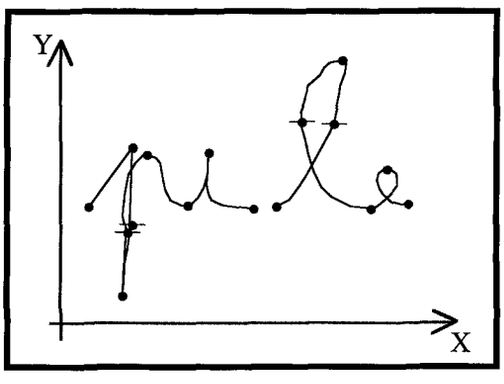


Fig. 2. – Exemple de découpage du tracé.

de représentation g_i de 4 composantes $g_i = [g_{i,1}, g_{i,2}, g_{i,3}, g_{i,4}]$ que nous désignerons par le terme **graphème**.

La figure 3 illustre le calcul des deux premières composantes d'un graphème.

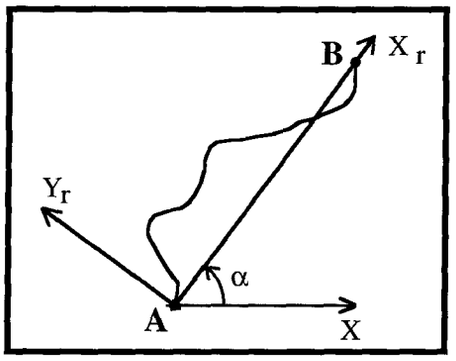


Fig. 3. – Codage d'un graphème. (α)

Si A est le premier point du tracé et B le dernier, la distance euclidienne ab entre A et B constitue la première composante $g_{i,1}$ de $g_{i,}.g_{i,2}$ est un codage de l'angle orienté α entre l'axe horizontal X et l'axe du tracé Xr.

Comme l'illustre la figure 4, on peut considérer que, dans le repère (Xr, Yr) lié au graphème, les coordonnés x_r et y_r de chaque point sont reliées entre elles par une fonction discrète F. En interpolant cette fonction F par une fonction continue f, linéaire, entre chaque point du tracé, il est possible de calculer les cœfficients de Fourier :

$$I_1 = \int_0^{ab} f(t) \cdot \sin(\frac{\pi}{ab} t)\, dt \qquad (1)$$

$$I_2 = \int_0^{ab} f(t) \cdot \sin(\frac{2\pi}{ab} t)\, dt \qquad (2)$$

qui sont les composantes $g_{i,3}$ et $g_{i,4}$.

Chacune de ces composantes $g_{i,j}$ est divisée par un cœfficient de normalisation Q_j puis discrétisée. Par conséquent, les cœfficients Q_j déterminent le niveau de discrétisation ainsi que l'importance relative de chaque composante i lors de la comparaison de deux graphèmes.

3.2. STRUCTURES ET FENÊTRES

Les graphèmes peuvent être de trois types :

« * » si le stylo est levé.

« + » si le tracé est ascendant.

« - » si le tracé est descendant.

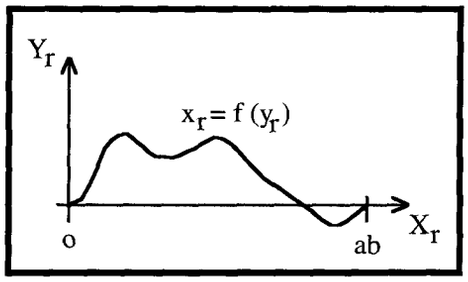


Fig. 4. – Codage d'un graphème (b).

Dans la suite, nous désignerons par **fenêtre** une suite de graphèmes — généralement une sous-suite de G. La **structure** d'une telle fenêtre est simplement la suite des types de ses graphèmes. La figure 5 présente la structure de la suite G après codage du tracé du mot anglais « mile ».

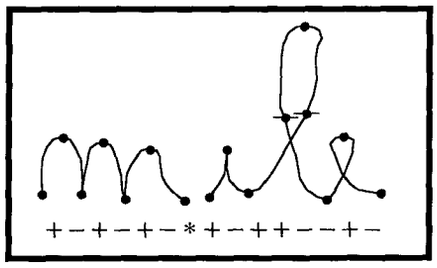


Fig. 5. – Exemple de structure.

Ces structures jouent un double rôle : accélérer le calcul des activations des références et permettre au module de segmentation de distinguer deux types de lettres « non reconnues ».

Les tracés de mots sont ainsi codés suivant deux niveaux de représentations. Le premier niveau est constitué par les composantes numériques des graphèmes. Il permet d'effectuer des comparaisons fines. Le second niveau est celui des structures. Il autorise des comparaisons plus grossières et plus rapides.

3.3. CALCUL DES RÉFÉRENCES

A chaque référence de lettre $\boldsymbol{r_k}$ sont associés trois éléments : une lettre alphabétique $\boldsymbol{L_k}$ $-$ $a, b, ...,$ ou $z -$, un prototype $\boldsymbol{P_k}$ qui est une fenêtre et une fiabilité $\phi_{\boldsymbol{k}}$. Cette section décrit le calcul des prototypes. Les fiabilités sont abordées au § 3.5.

Un ensemble $\boldsymbol{E}$ d'échantillons codé de lettres est obtenu à partir d'une base de mots $\boldsymbol{B}$ segmentée. Dans E, chaque échantillon de lettre est constitué d'une fenêtre associée à une lettre de l'alphabet. L'ensemble $\boldsymbol{E}$ est d'abord découpé en sous–ensembles $\boldsymbol{E(L,\Sigma)}$ contenant les échantillons de la lettre $\boldsymbol{L}$ et de structure . Ensuite, chaque sous–ensemble $E(L, \Sigma)$ est partitionné à son tour par regroupements successifs pour obtenir des ensembles $\boldsymbol{E(L, \Sigma, k)}$ qui seront chacun représentés par une référence r_k caractérisée elle même par un prototype P_k.

Cette construction des références est effectuée de façon incrémentale grâce à une variante de l'algorithme à seuil décrit dans [Belaid]. Les éléments de E sont considérés un par un. Pour chaque échantillon e de E, on cherche le premier ensemble $\boldsymbol{E(L, \Sigma, k)}$ tel que :

$$d_{\max}(e, f) < \Delta^c \text{ pour tout } f \in E(L, \Sigma, k)$$

où $\boldsymbol{d_{\max}(e,f)}$ désigne le plus grand écart entre les composantes de e et de f et $\boldsymbol{\Delta^c}$ est un seuil fixé à l'avance. Si un tel ensemble $E(L, \Sigma, k)$ existe, on lui ajoute e. Sinon, on construit un nouvel ensemble $E(L, \Sigma, k)$ à partir de e. Enfin, le prototype de chaque référence r_k est l'isobarycentre des éléments de $E(L, \Sigma, k)$.

Nous avons retenu cet algorithme de classement car il est très simple et très rapide. De plus l'utilisation du seuil de classification $\boldsymbol{\Delta^c}$ est plus souple qu'une stratégie où le nombre de références à construire est fixé à l'avance. On notera que le choix de $\boldsymbol{\Delta^c}$ reste empirique. Il s'agit, comme souvent, de réaliser un compromis. Un nombre restreint de références est un gage de rapidité alors qu'un nombre plus élevé permet une meilleure représentation des lettres, d'où un meilleur taux de reconnaissance. Notons qu'en général, le gain en terme de taux de reconnaissance est moins sensible que l'accélération induite.

3.4. ACTIVATION D'UNE RÉFÉRENCE

Soit $\boldsymbol{X}$ une suite de graphèmes et r_k une référence de lettre de prototype $\boldsymbol{P_k}$.

Définition 1 :

r_k est **active** pour X si et seulement si :

1) X et P_k ont même structure.

2) $d_{\max}(P_k, X) < \Delta^a$

Où $\boldsymbol{d_{\max}(P_k, X)}$ est le maximum des écarts entre les composantes de P_k et X, et Δ^a un seuil fixé à l'avance, nommé **seuil d'activation**.

L'utilisation d'un maximum pour déterminer l'activation d'une référence engendre une certaine imprécision. Malgré tout, nous avons retenu cette solution car elle accélère considérablement les calculs. En effet, la non–activation d'une référence peut être ainsi détectée dès les premières composantes des vecteurs X et P_k.

Définition 2 :

Si r_k est active pour X, l'**activité** de r_k est le réel :

$$A_k(X) = 1 - \frac{d_{\text{moy}}(X, P_k)}{d_0} \quad (3)$$

où $d_{\text{moy}}(P_k, X)$ est la moyenne des écarts entre les composantes de P_k et de X.

Le paramètre $\boldsymbol{d_0}$ est choisi empiriquement et est inférieur à Δ^a. Il est donc possible que l'activité d'une référence soit négative. Dans ce cas, cette référence sera considérée inactive. Ainsi, l'activité d'une référence (active) est comprise entre 0 et 1.

3.5. FIABILITÉ D'UNE RÉFÉRENCE

L'ensemble E peut contenir des échantillons de lettres douteux provenant de mots mal écrits ou d'une erreur de segmentation, ce qui conduit à des références de mauvaise qualité. De plus, la même lettre peut être écrite de différentes façons et donne ainsi lieu à plusieurs références plus ou moins représentatives. Il semble donc important de pouvoir estimer la « confiance » que l'on peut accorder à une référence de lettre. Pour cela, nous avons associé à chaque référence une mesure appelée **fiabilité**.

Soit L une lettre de l'alphabet et $R(L)$ l'ensemble des références de la lettre L. La fiabilité ϕ_i d'une référence r_i de L est obtenue par la formule :

$$\phi_i = \frac{\phi^0 + \log(\rho_i \cdot \delta_i)}{\phi^0} \quad (4)$$

Où ρ_i est la **représentativité** de r_i et δ_i sa **discriminance**.

ρ_i s'obtient par la formule :

$$\rho_i = \frac{\nu_i}{\boldsymbol{MAX}_{k\ tq\ r_k \in R(L)}\ \nu_k} \quad (5)$$

où $\boldsymbol{\nu_k}$ est le nombre d'activations de r_k pour les échantillons de L dans E. Pour chaque lettre L, il existe donc une référence dont la représentativité vaut 1, la représentativité des autres références de L étant mesurée par rapport à celle–ci.

$\delta_i$est obtenu en divisant ν_i par le nombre total d'activations de $\boldsymbol{r_i}$ sur l'ensemble des mots segmentés de B. La discriminance est donc une mesure de la probabilité (a posteriori) pour qu'une fenêtre X corresponde à la lettre L sachant que $\boldsymbol{r_i}$ est active.

Le paramètre ϕ^0 est un réel positif qui fixe la précision de notre mesure de fiabilité car les fiabilités négatives sont ramenées à 0. Nous avons retenu $\phi^0 = \log(10^{-7})$.

La fiabilité d'une référence est ainsi un nombre entre 0 et 1 qui reflète le caractère discriminant et représentatif d'une référence de lettre.

4. Système de segmentation de mots

4.1. ALGORITHME D'UNE PASSE DE SEGMENTATION

Comme nous l'avons expliqué au paragraphe II, la segmentation automatique de la base B s'effectue en plusieurs passes, chacune d'elles étant composée des trois étapes représentées dans la figure 6.

Comme on le voit dans la figure 6, de nouveaux échantillons de lettres sont découverts lors de l'étape i (ensemble d'échantillons R_i). Ensuite, ils donnent lieu à un nouvel ensemble de références

R_i^0 plus complet que R_{i-1}. Enfin, R_i^0 est filtré et seules les références de R_i^0 dont la fiabilité (calculée à l'étape 2) est supérieure à un seuil Φ_i sont conservées dans R_i.

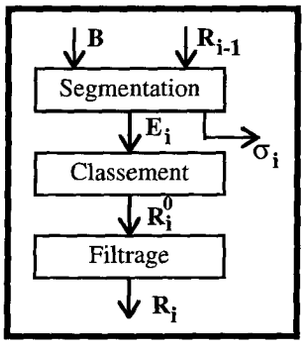


Fig. 6. – Schéma d'une passe de segmentation.

Bien sûr, une base de référence R_0 est nécessaire pour « amorcer » le processus de segmentation automatique. Celle-ci provient de quelques mots, d'écriture « standard », préalablement segmentés à la main. Chacune de ces trois étapes est décrite plus précisément dans les paragraphes suivants.

4.2. ÉTAPE DE SEGMENTATION

Avant de décrire davantage cette étape, nous allons expliciter dans les trois prochains paragraphes comment on calcule le niveau de confiance d'une *hypothèse* $H(L_i, X_i)$ selon laquelle X_i correspond à L_i. De même, le niveau de confiance d'un *chemin d'hypothèses* associant l'ensemble d'un tracé de mot codé à son étiquette sera défini.

4.2.1. Hypothèses réelles

Une fenêtre X de G est associée à une lettre L s'il existe une référence r_k de L active pour X. On obtient alors une hypothèse $H(L, X, k)$ dont le niveau de confiance N est :

$$N(L, X, k) = \phi_k \cdot A_k(X) \tag{6}$$

Où ϕ_k est la fiabilité de r_k et A_k son activité pour X.

Soit r_m la référence ayant le plus grand niveau de confiance pour L et X.

On pose :

$$H(L, X) = H(L, X, m)$$
$$N(L, X) = N(L, X, m)$$

$N(L, X)$ est alors le niveau de confiance de l'hypothèse $H(L, X)$ selon laquelle la fenêtre X correspond à la lettre L.

4.2.2. Hypothèses virtuelles

Un chemin peut comporter des hypothèses $H(L, X)$ virtuelles qui ne correspondent à aucune activation d'une référence de L (lettre « non reconnue »). Les hypothèses sont ainsi de 3 types :

h_0 : Hypothèses *réelles*
Celles du paragraphe précédent. Elles représentent une correspondance fine entre une fenêtre et une lettre de l'alphabet.

h_1 : Hypothèses *virtuelles primaires*
Aucune référence de L n'est active pour X, cependant il existe une référence de L dont la structure est celle de X. Le niveau de confiance d'une telle hypothèse est une constant négative N_1. Les hypothèses virtuelles primaires rendent ainsi compte d'une correspondance grossière.

h_2 : Hypothèses *virtuelles secondaires*
La structure de X ne correspond à celle d'aucune référence de L. Le niveau de confiance N_2 d'une telle hypothèse ne dépend que de la taille de la fenêtre X afin de défavoriser les hypothèses de tailles extrêmes et est toujours inférieur à N_1. Les hypothèses virtuelles secondaires sont ainsi purement hypothétiques !

4.2.3. Chemins d'hypothèses

Soit $M = (M_1, \ldots, M_m)$ une étiquette de mot codé composée des m lettres M_i. Un **chemin** c pour M est une suite de m hypothèses $H(L_i, X_i)$ telles que :

1) $L_i = M_i$ pour $i = 1, \ldots, m$

2) Les X_i respectent les règles d'adjacence

3) c ne contient pas deux hypothèses virtuelles successives

Les règles d'adjacence définissent comment doivent se succéder les fenêtres des hypothèses. Ainsi, deux fenêtres peuvent se suivre ou être séparées par une rupture. La figure 7 présente une segmentation qui respecte les règles d'adjacence.

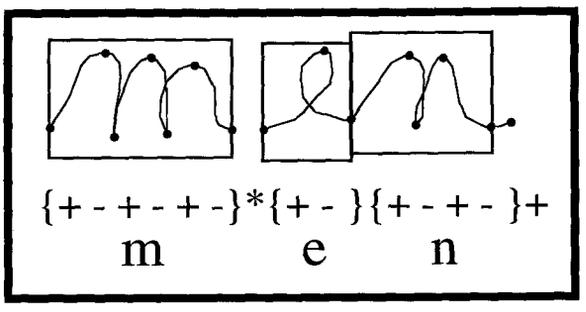


Fig. 7. – Exemple de segmentation.

Le **score** d'un chemin est simplement la moyenne des niveaux de confiance de ses hypothèses.

4.2.4. Segmentation

Le module de segmentation génère tous les chemins correspondant à l'étiquette M. Si aucun chemin n'existe, le mot n'est pas segmenté. Sinon, le chemin possédant le meilleur score est retenu, et les hypothèses qu'il renferme définissent la segmentation du mot. Les hypothèses virtuelles donnent toujours naissance à de nouvelles références de lettres. Les hypothèses réelles peuvent également engendrer de nouveaux modèles de lettres car le seuil d'activation Δ^a est supérieur au seuil de classement Δ^c.

4.3. CLASSEMENT ET CALCUL DES FIABILITÉS

Les mots segmentés lors de la passe i fournissent un ensemble E_i d'échantillons de lettres. Ensuite, l'ensemble de références R_i^0 est obtenu à partir de E_i grâce au classement exposé au paragraphe 3.3. Enfin, les fiabilités des éléments de R_i sont calculées avec les formules du paragraphe 3.5.

4.4. FILTRAGE DE LA NOUVELLE BASE DE RÉFÉRENCE

Le « filtrage » de l'ensemble R_i^0 consiste à ne conserver dans R_i que les références dont la fiabilité est supérieure à un seuil Φ_i lié à l'étape i (pour $i = 1, \ldots, N_p$). Notons que Φ_i décroît linéairement entre des valeurs Φ_i et Φ_{N_p} fixées à l'avance. On « admet » ainsi les références les plus fiables en priorité.

Les résultats expérimentaux exposés au paragraphe 6 montrent que cette progression régulière de Φ est satisfaisante car ainsi, le nombre de références découvertes à chaque passe garde le même ordre de grandeur.

5. Système de reconnaissance des mots

Le système de reconnaissance de mots fonctionne de façon analogue au système de segmentation. La principale différence est la suivante : le mot codé à reconnaître, possède non pas une, mais une multitude d'étiquettes possibles. Cet ensemble d'étiquettes possibles est en l'occurrence le dictionnaire du système (plusieurs dizaines de milliers de mots). Il s'agit donc de trouver les mots du dictionnaire dont les scores sont les meilleurs et de classer ceux–ci par score décroissant. Le score d'un mot du dictionnaire étant bien sûr le score de son meilleur chemin d'hypothèses. Le mot de meilleur score est la (première) réponse du système.

La recherche combinatoire se fait alors dans un espace plus grand, aussi est il nécessaire de limiter les hypothèses virtuelles à une seule par mot, voire même de les interdire.

6. Résultats expérimentaux

Les résultats que nous présentons ici sont donnés pour la segmentation d'une base de mots provenant de plusieurs scripteurs. Après avoir présenté cette base, nous décrirons le processus de segmentation utilisé. Ensuite nous analyserons la qualité de la segmentation obtenue, puis nous étudierons le nombre de références après filtrage en fonction du seuil de fiabilité. Enfin, nous donnerons les performances obtenues par notre système de reconnaissance après apprentissage sur la base segmentée.

6.1. BASE À SEGMENTER

Nous disposons d'un ensemble B de 10000 mots codés et étiquetés provenant de 10 scripteurs différents, chacun d'eux ayant écrit 1000 mots sur la tablette. Le lecteur pourra apprécier la qualité de cette base en se reportant à la page suivante qui contient un échantillon de la production des 4 premiers scripteurs.

B est découpée aléatoirement en deux parties $\boldsymbol{B_l}$ et $\boldsymbol{B_t}$ contenant respectivement 9000 et 1000 mots.

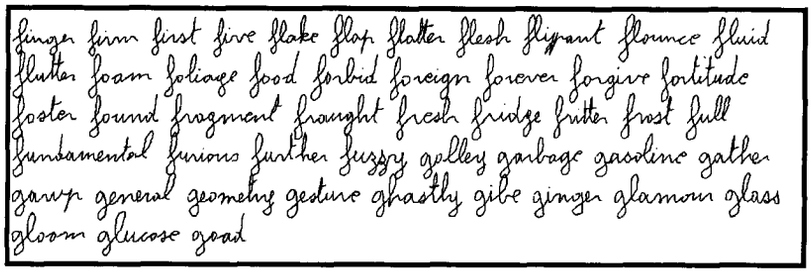
6.2. PROCESSUS DE SEGMENTATION

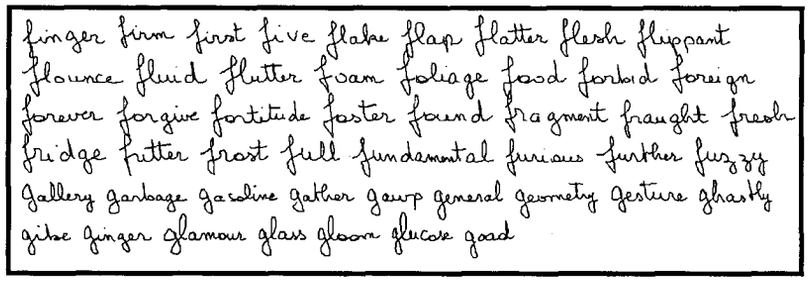
Seule la base B_l a été segmentée automatiquement. Le nombre de passes Np du processus de segmentation a été fixé expérimentalement à 20, 40 passes n'apportant pas d'amélioration significative du résultat. Le seuil de fiabilité ϕ_i utilisé pour le filtrage des références a varié linéairement entre les valeurs $0,5$ et $0,1$. On observe une progression régulière du nombre de références au cours des passes successives. On notera que l'ensemble de références initial R_0 provenait de mots segmentés en lettres à la main. Ces mots (fantaisistes) contenaient au total 5 échantillons de chacune des lettres de l'alphabet. Les 130 échantillons de lettres ainsi obtenus ont produit, après classement, 91 références de lettres.

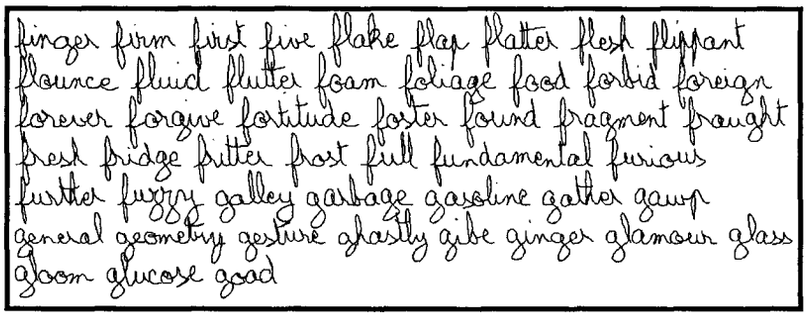
Le processus a duré environ 8 heures sur une station SUN Sparc 10. Une étude expérimentale montre que la durée du processus est liée à la variabilité de l'écriture et que le nombre de passes nécessaire ne varie pas significativement avec la taille n de la base. On constate ainsi que la durée d'une segmentation automatique est de l'ordre de $k \cdot n \cdot \log(n)$, la valeur de k étant liée à la « qualité » de la base. Contrairement à ce que l'on pourrait croire, l'étape la plus coûteuse en temps de calcul n'est pas la segmentation mais (et de loin) le calcul des fiabilités des références.

A la fin du processus, 99,9 % des mots de la base ont été segmentés.

Echantillon des 4 premières bases de mots







6.3. QUALITÉ DE LA SEGMENTATION OBTENUE

La base B_l utilisée ici n'a pas été soumise à un « nettoyage » manuel, elle contient ainsi des mots de mauvaise qualité avec des lettres mal formées ou des fautes d'orthographe.

Mesurer objectivement la qualité de la segmentation produite est une chose très difficile. Pour faire cela, il faudrait comparer la segmentation réalisée avec une segmentation idéale qui reste à définir. La solution que nous avons retenue est de juger nous même les segmentations proposées en essayant d'être le plus objectif possible (!).

L'examen des 9000 mots segmentés étant une tâche trop fastidieuse, nous n'avons étudié qu'un échantillon de 100 mots prélevés aléatoirement. La précision de nos mesures sera donc

faible mais suffisante pour avoir une idée des performances de notre système.

Le résultat de notre mesure est le suivant :

- Segmentations correctes : 88
- Erreurs inévitables : 2
- Segmentations incorrectes : 10
- Mots non segmentés : 0

Les 10 segmentations incorrectes constatées ne concernaient qu'un seul point de segmentation par mot (erreur sur une à deux lettres donc), ce point étant décalé de 1 graphème par rapport au choix idéal selon nous. La segmentation obtenue ne semble donc optimale que pour 8 à 9 mots sur 10, la plupart des erreurs ne portant que sur un graphème mal attribué.

6.4. SEUIL DE FIABILITÉ ET NOMBRE DE RÉFÉRENCES

La base B_l segmentée a été utilisée pour produire un ensemble R de références. La figure 8 donne le nombre d'éléments de R dont la fiabilité dépasse un certain seuil en fonction de ce seuil Φ.

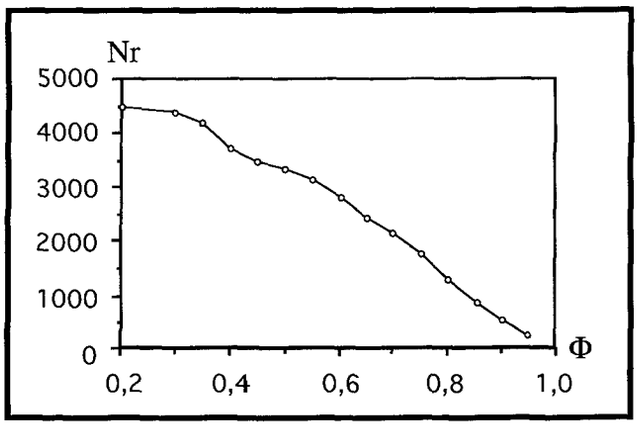


Fig. 8. – Nombre de références (Nr) en fonction du seuil.

On constate que le nombre de références varie à peu près linéairement avec le seuil de filtrage Φ. Voila pourquoi nous nous sommes contentés d'une variation linéaire de Φ_i lors de la segmentation automatique. Le nombre de nouvelles références conservées dans R_i lors de chaque passe i garde en effet le même ordre de grandeur.

6.5. SEUIL DE FIABILITÉ ET PERFORMANCES EN RECONNAISSANCE

Afin d'évaluer l'intérêt de notre segmentation automatique, nous avons étudié les performances de notre système de reconnaissance de mots, sa base d'apprentissage étant la base B_l (9000 mots) obtenue précédemment. Ces performances ont été mesurées sur la base B_t, (1000 mots) pour un dictionnaire de 23000 mots. Les résultats sont présentés dans les tableau 1, 2 et 3. Dans chacun de ces tableaux, la colonne « %***Rc*** » donne le pourcentage de mots reconnus, c'est à dire, classiquement, le nombre de mots reconnus divisé par le nombre de mots présentés au système. De même, la colone « %***Rf*** » donne la proportion de mots confondus, c'est à dire pour lesquels la réponse du système est erronée. Le taux de mots rejeté, c'est à dire ceux pour lesquels le système ne propose aucune réponse, est indiqué dans la colonne « %Rj ». La somme de ces trois colonnes est donc toujours égale à 100%. Enfin, nous avons indiqué, dans la quatrième colonne « %***Fi*** », la fiabilité moyenne des réponses du système, c'est à dire le nombre de réponses correctes divisé par le nombre de mots *non rejetés*.

Les premiers résultats, présentés dans le tableau 1, montrent l'intérêt de la prise en compte des fiabilités des références dans le calcul des niveaux de confiance des hypothèses réelles (cf. § 4.2.1). La première ligne du tableau donne les résultats obtenus lorsque les fiabilités ϕ_k sont toutes forcées à 1. Dans ces deux tests, seules les hypothèses réelles sont autorisées.

Tableau 1. – Prise en compte de la fiabilité.

	%***Rc***	%***Cf***	%***Rj***	%***Fi***
sans	**82,3**	7,9	9,8	91,2
avec	**85,0**	5,2	9,8	94,3

On constate tout d'abord que la proportion de mots correctement identifiés est relativement faible. Il ne faut cependant pas oublier que le système connaît alors dix scripteurs simultanément et que la taille du dictionnaire est plutôt importante, alors que seule la meilleure réponse du système est prise en compte. De plus, comme seules les hypothèses réelles sont autorisées, la robustesse du système est faible, d'où les taux de rejet importants. On obtient ainsi une fiabilité honorable. Enfin, les chiffres démontrent clairement l'apport des fiabilités. Celles–ci ont en effet permis de réduire les confusions de façon assez considérable.

L'expérience suivante donne une autre confirmation du bien fondé de notre mesure de fiabilité. Nous avons refait l'expérience correspondant à la seconde ligne du tableau 1, (fiabilités des références prises en compte) après élimination d'un quart des références au moyen du filtrage décrit au § 4.4. La première ligne du tableau 2 reprend les résultats de l'expérience précédente (sans filtrage) alors que la seconde ligne donne les résultats obtenus après élimination des références de moindre fiabilité (avec filtrage).

Tableau 2. – Effet du filtrage.

	%***Rc***	%***Cf***	%***Rj***	%***Fi***
sans	85,0	5,2	9,8	**94,3**
avec	68,0	2,4	29,6	**96,5**

On observe tout d'abord que le filtrage opéré ici engendre une chute importante du taux de reconnaissance. Cela n'est pas surprenant car nous effectuons ici un filtrage très important, si bien que de nombreuses références « utiles » sont éliminées, ce qui augmente considérablement le rejet. Par contre, il apparaît que les références restantes sont de meilleure « qualité » car elles donnent lieu à un système considérablement plus fiable (cf. colonne $\%Fi$).

Enfin, les derniers résultats que nous présentons dans le tableau 3 donnent les performances de notre système lorsque les hypothèses virtuelles sont autorisées (seconde ligne du tableau).

Tableau 3. – Intérêt des hypothèses virtuelles.

	$\%Rc$	$\%Cf$	$\%Rj$	$\%Fi$
sans	**85,0**	5,2	9,8	94,3
avec	**93,2**	6,7	**0,1**	93,3

L'introduction des hypothèses virtuelles élimine pratiquement le rejet, et ce, sans trop augmenter le nombre de confusions. On obtient alors un taux de reconnaissance de plus de 93% sur l'ensemble des dix scripteurs, ce qui est tout à fait satisfaisant. On notera que les bonnes performances du système de reconnaissance constituent une validation indirecte de notre système de segmentation automatique, car c'est lui qui est à la base de la production des références de lettres utilisées.

7. Conclusion

Nous avons présenté un système capable de découper automatiquement une base de mots manuscrits anglais codés et étiquetés en lettres à partir d'un petit ensemble R_0 de références de lettres préalablement établi. Ce système de segmentation est basé sur la découverte de nouvelles références par satisfaction de contraintes d'adjacence ainsi que sur une évaluation de la « fiabilité » de chacune de ces références. La qualité de la segmentation réalisée rivalise presque avec celle d'un expert humain et est suffisante pour permettre à un système de reconnaissance, l'apprentissage automatique d'une écriture à partir d'exemples de mots étiquetés et de quelques références de lettres.

8. Annexes

A.1. NOTATIONS

Les principales notations utilisées dans cet article sont, par ordre d'apparition :

- B : **Ensemble de mots à segmenter** appelé également base. Chaque mot de B est accompagné de son étiquette.
- G : **Mot codé**. G est une suite de graphèmes. G_k est le k–ième mot codé de B.
- Γ : **Étiquette de mot codé**. Γ est une suite de lettres alphabétiques. Γ_k est l'étiquette de G_k.
- L : **Lettre alphabétique**. L peut valoir « a », « b »,. . ., « y » ou « z »
- σ : **Segmentation de** B. σ indique la position de chacune des lettres de la base de mots B. σ est donc une liste de segmentations de mots s_k. On notera que s_k, la segmentation de G_k, est un chemin d'hypothèses.
- Np : **Nombre de passes**. Paramètre : nombre d'itérations de l'algorithme de segmentation automatique des mots en lettres.
- R_i : **Ensemble de Références de lettres**. R_i contient les références de lettres produites lors de la i–ième passe de l'algorithme de segmentation.
- γ_i : **Portion de tracé**. γ_i est la i–ième portion du tracé et correspond à une suite de points. On lui associe un vecteur $g_i = [g_{i,1}, \ldots, g_{i,4}]$ appelé **graphème**. gi est le i–ième élément de G. On associe également à γ_i un type τ_i qui est un symbole pouvant prendre l'une des trois valeurs « + », « − » ou « ∗ ».
- X : **Fenêtre**. X est une suite de portions —ou graphèmes. La **structure** de X est la suite des types des portions de X.
- r : **Référence**. r représente un allographe de lettre. r_k est la k–ième référence. Trois éléments sont associés à r_k : une lettre alphabétique L_k, un prototype P_k qui est une fenêtre et une fiabilité $\phi_k \in [0, 1]$.
- e : **Échantillon de lettre**. On associe à e, une fenêtre et une lettre alphabétique.
- E : **Ensemble d'échantillons de lettres**. E est partitionné en sous–ensembles $E(L, \Sigma)$. $E(L, \Sigma)$ contient les échantillons de la lettre L dont la fenêtre a la structure Σ. $E(L, \Sigma)$ sert à produire l'ensemble de prototype $R(L, \Sigma)$.
- Φ : **Seuil de fiabilité**. Les références qui ont une fiabilité inférieure à Φ sont supprimées. Φ_i est le seuil de fiabilité de la i–ème passe de segmentation.
- H : **Hypothèse**. $H(L, X)$ représente la mise en correspondance de la fenêtre X avec la lettre L. Une hypothèse peut être **réelle** (correspondance forte), **virtuelle primaire** (correspondance faible) ou **virtuelle secondaire** (pas de correspondance constatée). Le niveau de confiance accordé à $H(L, X)$ est noté $N(L, X)$. Il s'agit d'un réel inférieur ou égal à 1, pouvant être négatif.

BIBLIOGRAPHIE

[Belaïd] A. et Y. Belaïd, *Reconnaissance des formes*, InterEdition, 1992.

[Duneau.a] L. Duneau and B. Dorizzi, « Incremental building of an allograph lexicon », *Advances in handwriting and drawing*, Europia, 1994, pp. 39-54.

[Duneau.b] L. Duneau and B. Dorizzi, « On–line cursive script recognition : a system that adapts to an unknown user. », *International Conference on Pattern Recognition*, Vol. II, Jerusalem, October 94, pp. 24-28.

[Duneau.c] L. Duneau, « Etude et réalisation d'un système adaptatif pour la reconnaissance en ligne des mots manuscrits », *Thèse de doctorat de l'Université Technologique de Compiègne*, décembre 94.

[Eden] M. Eden, « Handwriting and pattern recognition », *IRE Transactions on Information Theory*, Feb. 1961.

[Fujisaki] T. Fujisaki, K. Nathan, W. Cho, H. Beigi, « On–line unconstrained handwriting recognition by a probabilistic method », *International Workshop on Frontiers in Handwriting Recognition III*, Bufallo, 1993.

[Higgins] C.A. Higgins and D.M. Ford, « On–line recognition of connected handwriting by segmentation and template matching », *Proc. of the 11th International Conference on Pattern Recognition*, The Hague,1992, pp. 200-203.

[Morasso] P. Morasso, L. Barberis, S. Pagliano, and D. Vergano, « Recognition experiments of cursive dynamic handwriting with self–organizing networks », *Pattern Recognition*, Vol. 26, N°3, 1993, pp. 451-460.

[Plamondon] R. Plamondon, « A model–based segmentation framework for computer processing of handwriting », *Proc. of the 11th International Conference on Pattern Recognition*, The Hague,1992, pp. 303-307.

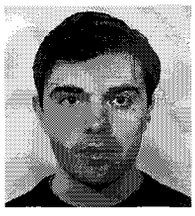
[Tappert] C.C. Tappert, C.Y. Suen, T. Wakahara, « The state of art in on–line handwriting recognition », *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 12, N°8, August 1990.

[Teulings] H.L. Teulings, L.R.B. Schomaker, « Unsupervised learning of prototype allographs in cursive script recognition », *From Pixels to Features III : Frontiers in Handwriting Recognition*, S. Impedovo and J.C. Simon (eds), 1992 Elsevier Science Publisher B.V., pp. 61-73.

Manuscrit reçu le le 15 Juin 1995.

LES AUTEURS

Laurent DUNEAU



Laurent Duneau, né le 25 décembre 1967 à Orléans, est docteur de l'Université Technologique de Compiègne (UTC) et diplômé de l'ENSI de Caen. La thèse qu'il a préparé à l'Institut National des Télécommunications (INT), de 1991 à 1994, a pour objet la reconnaissance des mots manuscrits.

Bernadette DORIZZI

Bernadette Dorizzi, Ecole Normale Supérieure de Fontenay, agrégée de Mathématiques, est actuellement professeur à l'Institut National des Télécommunications (INT) où elle anime un groupe de recherche sur les réseaux neuronaux et la reconnaissance des formes. Ses principaux centres d'intérêt sont la reconnaissance de l'écriture manuscrite et la prédiction de signaux temporels.