

Thèse

présentée pour obtenir le grade de docteur de l'Ecole
Nationale Supérieure des Télécommunications

Spécialité : Informatique

José Ruiz Pinales

Reconnaissance hors-ligne de l'écriture cursive par l'utilisation de modèles perceptifs et neuronaux

Soutenue le 21 septembre devant le jury composé de :

George Stamon	président du jury
Bernadette Dorizzi	rapporteur
Nicole Vincent	rapporteur
Eric Lecolinet	directeur de recherche
Jean Pierre Crettez	examineur
Claudie Faure	examineur
Jacques Labiche	examineur
Guy Lorette	examineur

Résumé

Cette thèse porte sur l'application de modèles perceptifs et neuronaux au problème de la reconnaissance hors-ligne de l'écriture cursive. La reconnaissance de l'écriture cursive est un problème très difficile à résoudre en sa forme générale du fait de la variabilité intrinsèque de l'écriture cursive. En revanche, les lecteurs humains sont capables de reconnaître n'importe quel texte, même dans des conditions difficiles. Il est donc vraisemblable que la modélisation du processus de lecture humain puisse apporter des éléments importants pour le développement des futurs systèmes de reconnaissance de l'écriture.

Dans un premier temps, nous abordons le problème de l'extraction de caractéristiques par l'utilisation d'une technique robuste de détection de droites. Puis, nous appliquons cette méthode au problème de la reconnaissance de caractères cursifs segmentés. Nous utilisons ensuite cette méthode pour la reconnaissance holistique de mots cursifs ainsi qu'à la reconnaissance de leurs premières et dernières lettres. Nous nous intéressons enfin à la mise au point d'un modèle perceptif pour la reconnaissance de mots cursifs.

Notre méthode d'extraction de caractéristiques est capable de capturer une grande partie de l'information contenue dans la partie singulière du mot cursif (ascendants, descendants et boucles) et présente une très bonne résistance au bruit et aux déconnexions du tracé.

Le modèle d'activation interactif est capable de modéliser l'accès lexical, l'effet de supériorité du mot et l'effet de fréquence. L'effet de supériorité du mot peut d'ailleurs être en partie attribué à la régularité des chaînes de lettres. Nous avons proposé l'utilisation d'une étape de pré-reconnaissance de chaînes de lettres et une étape de reconnaissance de mots basée sur un processus d'activation interactif afin de prendre en compte ces effets. La reconnaissance étant contrainte par les extrémités du mot, les lettres externes sont de ce fait mieux reconnues que les lettres internes. Ainsi, nous pouvons simuler un processus de reconnaissance allant de l'extérieur vers l'intérieur du mot.

Nous avons validé nos méthodes pour une tâche de reconnaissance de caractères cursifs segmentés et une tâche de reconnaissance de mots cursifs isolés pour lesquelles nous avons obtenu des résultats intéressants.

Remerciements

Je tiens tout d'abord à remercier les rapporteurs de ce travail, les professeurs Bernadette Dorizzi et Nicole Vincent. J'ai particulièrement apprécié l'intérêt que cette dernière a porté à ce travail. Les nombreuses suggestions qu'elles m'ont faites ont amélioré la qualité finale de ce document, et je leur en remercie sincèrement.

Ce travail n'aurait pas pu se faire sans la bienveillance de Eric Lecolinet, qui m'a guidé tout au long de cette thèse. Au début de ce travail, il insistait sur l'intérêt de l'application de la transformé de Hough pour l'extraction de caractéristiques.

Je suis très reconnaissant à George Stamon, professeur à l'Université Paris V, d'avoir accepté de présider le jury et pour son aide inestimable tout au long de cette thèse et du DEA IARFA.

Je remercie Claudie Faure, chargé de recherche au CNRS, Jean Pierre Crettez, ingénieur de recherche au CNRS, maintenant à la retraite, Jacques Labiche, professeur à l'université de Rouen, et Guy Lorette, professeur à l'université de Rennes, d'avoir accepté de participer au jury.

Je remercie aussi Hayette pour son amitié et pour l'aide précieuse qu'elle m'a donnée aux moments les plus critiques de mon travail.

Je remercie les membres de l'équipe de la bibliothèque de l'ENST Paris pour leur excellente service et leur attention face à mes nombreuses demandes.

Je remercie aussi à tous mes amis de la génération SFERE'96, et à tous les autres que j'ai connus à la maison du Mexique, à Paris VI et à l'ENST Paris.

Ce travail a été financé en partie par le Conseil National pour la Science et la Technolo-

gie (CONACyT) et la Société Française d'Exportation des Ressources Educatives (SFERE) dans le cadre d'un Programme de Coopération Scientifique et Technique franco-mexicain SFERE/CONACyT, patronné par le Ministère français des Affaires Etrangères. En particulier, je suis très reconnaissant à Anna Maneta pour son excellent travail et son attention.

Je remercie René Jaime Rivas, Roberto Rojas, José Amparo Andrade, Oscar Ibarra, Donato Valdez et Marc Ly pour leur amitié et leur soutien moral et financier durant cette dernière année de thèse.

Je remercie enfin tous ceux et celles qui ont contribué d'une manière ou d'une autre pour que je mène à bien mon travail de thèse, ils sauront se reconnaître.

Table des matières

1	Introduction	1
1.1	Le problème	3
1.1.1	Méthodes d'acquisition	4
1.1.2	Types d'écriture	5
1.1.3	Nombre de scripteurs	6
1.1.4	Taille du lexique	7
1.2	Applications	7
1.2.1	Traitement de formulaires	8
1.2.2	Traitement de chèques	8
1.2.3	Tri de courrier	9
1.3	Objectif de notre étude	11
1.4	Organisation de la thèse	12
2	État de l'art de la reconnaissance de l'écriture	15
2.1	Introduction	15
2.2	Reconnaissance de mots	16
2.2.1	Approches analytiques explicites	16
2.2.2	Approches analytiques implicites	19
2.2.3	Approches holistiques	21
2.2.4	Approches hybrides	22
2.2.5	Exemples de systèmes	22

2.3	Reconnaissance de caractères	24
2.3.1	Les diverses approches	24
2.3.2	Extraction de caractéristiques	26
2.3.2.1	Caractéristiques globales	27
2.3.2.2	Caractéristiques structurelles	27
2.3.2.3	Caractéristiques morphologiques	27
2.3.2.4	Caractéristiques de contour	28
2.3.2.5	Caractéristiques statistiques	29
2.3.2.6	Caractéristiques métriques	30
2.3.2.7	Caractéristiques adaptatives	30
2.3.3	Techniques de classification	31
2.3.3.1	Statistiques et neuronales	31
2.3.3.2	Stochastiques	32
2.3.3.3	Structurelles	32
2.3.4	Exemples de systèmes	33
2.4	Conclusions	34
3	Psychologie de la reconnaissance de l'écriture	37
3.1	Introduction	37
3.2	Modèles d'accès lexical	38
3.2.1	Modèles de recherche (Search models)	38
3.2.2	Modèles d'activation (threshold models)	39
3.3	Les différents modèles	40
3.3.1	Le modèle pandémonium	40
3.3.2	Le modèle Logogen	41
3.3.3	Le modèle de vérification	42
3.3.4	Le modèle d'activation interactif	42
3.3.5	Le modèle à deux voies en cascade (DRC model)	44

3.3.6	Le modèle triangulaire	45
3.4	Systèmes de reconnaissance de l'écriture	46
4	Extraction de caractéristiques	51
4.1	Introduction	51
4.1.1	Transformée de Hough	53
4.1.2	Variabilité	56
4.1.3	Préservation de l'information	57
4.1.4	Applications	59
4.1.5	La transformée de Radon	63
4.2	Méthodes développées	64
4.2.1	Extraction de caractéristiques directionnelles	65
4.2.2	Information locale	67
4.2.3	Autre méthode considérée	69
4.3	Caractéristiques de courbure	70
4.4	Généralisation	71
4.5	Conclusions	75
5	Reconnaissance de caractères cursifs	77
5.1	Introduction	77
5.2	Notre système	78
5.3	Extraction de caractéristiques	78
5.4	Reconnaissance	81
5.5	Résultats	82
5.5.1	La base de données	82
5.5.2	Conditions de test	83
5.5.3	Evaluation des caractéristiques	84
5.5.4	Résultats obtenus	85
5.6	Conclusions	87

6	Reconnaissance holistique de mots cursifs	91
6.1	Introduction	91
6.2	Architecture	92
6.3	Extraction de caractéristiques	92
6.4	Résultats	95
6.4.1	Base de données	95
6.4.2	Conditions de test	95
6.4.3	Résultats obtenus	97
6.4.3.1	Reconnaissance de la première lettre	97
6.4.3.2	Reconnaissance de mots cursifs	98
6.5	Conclusions	99
7	Reconnaissance perceptive de mots cursifs	105
7.1	Introduction	105
7.2	Fondements psychologiques	106
7.3	Architecture	110
7.4	Pré-traitement	110
7.5	Extraction de caractéristiques	112
7.6	Pré-reconnaissance de lettres	113
7.7	Reconnaissance de mots	115
7.7.1	Apprentissage	117
7.7.2	Reconnaissance	121
7.8	Résultats	121
7.9	Extensions possibles du système	123
7.10	Conclusions	126
	Conclusions générales	129
	Bibliographie	133

Table des figures

1.1	Ambiguïté de l'écriture. S'agit-il du mot "clump" ou du mot "dump" ? . . .	4
1.2	Types d'écriture [Tapp90].	6
1.3	Traitement automatique de formulaires [Lam93].	8
1.4	Traitement automatique de chèques [Gors99].	9
1.5	Traitement automatique de courrier [LeeCK99].	10
2.1	Schéma générique d'un système de reconnaissance de mots [Ste00].	16
2.2	Approches possibles [Case96].	17
2.3	Projection verticale.	18
2.4	Analyse de composantes connexes. En haut, segmentation en graphèmes [Leco91]. En bas, segmentation optimale en lettres. Les lettres 'T', 'u', 'n' ont été obtenues chacune par la fusion de deux graphèmes.	19
2.5	Segmentation à l'aide de la programmation dynamique.	20
2.6	Modèles de Markov cachés (HMM).	21
2.7	Caractéristiques morphologiques [Mitt89].	28
2.8	Caractéristiques statistiques [Itoh90].	29
2.9	Caractéristiques métriques.	30
2.10	Caractéristiques adaptatives.	31
3.1	Le modèle pandémonium [Self59].	40
3.2	Le modèle Logogen [Mort69].	41
3.3	Le modèle d'activation interactif [McCl81].	43

3.4	Codage des lettres.	43
3.5	Reconnaissance visuelle et auditive de mots [McCl81].	44
3.6	Le modèle à deux voies en cascade (DRC model) [Colt94].	45
3.7	Le modèle triangulaire [Seid89].	45
3.8	Le système PERCEPTO [Côté97].	49
4.1	Illusions optiques. a) Cette figure présente un carré et un cercle parfaits qui semblent être déformés. b) Les droites diagonales sont en réalité parallèles.	52
4.2	Paramétrisation d'une droite.	54
4.3	Problème de la longueur des segments. ($L_1 = 8\sqrt{2}$, $L_2 = 8$)	56
4.4	Variation de l'épaisseur. En haut, des images de test. En bas, les transformées de Hough respectives.	57
4.5	Effets dus à des transformations de l'image. En haut, des images de test. En bas, les transformées de Hough respectives. Les valeurs de l'accumulateur sont indiquées par des niveaux de gris (à droite de chaque image).	58
4.6	Reconstitution de l'image. a) Image de test, b) à e) sa reconstitution pour $N_\theta = 6, 8, 10, 16$ et 32 respectivement.	60
4.7	Exemple d'extraction de caractéristiques de contour [Taka91].	62
4.8	Cartes de caractéristiques directionnelles obtenues par notre méthode globale basée sur la FFT pour θ allant de 0 à 150° . A gauche, des images de test. A droite, les cartes de caractéristiques correspondantes.	67
4.9	Utilisation de l'information locale. A gauche, des images de test. A droite, les cartes de caractéristiques correspondantes. En haut, la méthode basée sur la transformée de Hough. En bas, la méthode locale basée sur la FFT pour $\lambda = 18$	68
4.10	Le noyau $G_\theta(u, v)$. ($\Delta\theta = \frac{\pi}{N_\theta}$, N_θ = nombre d'orientations)	70

4.11	Extraction de caractéristiques directionnelles par l'utilisation de secteurs symétriques. A gauche, des images de test. A droite, les cartes de caractéristiques correspondantes.	70
4.12	Extraction de caractéristiques de courbure. A gauche, des images de test. A droite, les cartes de caractéristiques correspondantes.	71
4.13	Détection de boucles et de cavités centrales.	73
4.14	Des caractéristiques capturant une partie de l'information contenue dans les points caractéristiques.	74
4.15	Estimation de l'épaisseur moyenne du tracé. A gauche, des images de test. A droite, les histogrammes correspondants (sans lissage).	75
5.1	Architecture du système.	78
5.2	Extraction de caractéristiques directionnelles à l'aide de la transformée de Hough. A gauche, des images de test. A droite, les cartes de caractéristiques correspondantes.	80
5.3	Division de l'image en zones [Taka91].	80
5.4	Caractéristiques obtenues après échantillonnage. A gauche, une image de test. A droite, les caractéristiques correspondantes.	80
5.5	Quelques exemples de la base de données.	83
5.6	Courbe de rejet pour la tâche de reconnaissance de caractères cursifs segmentés.	87
6.1	Architecture du système.	93
6.2	Extraction de caractéristiques directionnelles. A gauche, des images de test. A droite, les cartes de caractéristiques correspondantes.	94
6.3	Détection de boucles.	94
6.4	Quelques exemples de la base de données.	95
6.5	Exemple des images utilisées pour la reconnaissance de mots cursifs (normalisées par rapport à la zone centrale).	96
6.6	Exemple des images utilisées pour la reconnaissance de la première lettre.	97

6.7	Courbe de rejet pour la tâche de reconnaissance de mots cursifs.	100
7.1	Incertitude de la position des lettres. A gauche, normalisation par rapport à la zone centrale. A droite, normalisation en largeur.	108
7.2	Architecture possible pour la reconnaissance de chaînes de lettres.	109
7.3	Architecture du système.	111
7.4	Normalisation de la taille des images.	111
7.5	Normalisation de l'épaisseur du tracé. En haut, des images de test. En bas, les images normalisées.	112
7.6	Cartes de caractéristiques directionnelles. A gauche, des images de test. A droite, les cartes de caractéristiques respectives.	113
7.7	Cartes de boucles. A gauche, les images de test. A droite, les cartes de caractéristiques respectives.	113
7.8	Caractères composés produits par l'analyse de composantes connexes [Guil95a].	114
7.9	Architecture pour la pré-reconnaissance de lettres.	116
7.10	Détail des connexions entre les niveaux correspondant aux mots de 3 et 4 lettres. (Notons que le réseau ne comporte aucune connexion inhibitrice et qu'il n'existe pas de connexions entre niveaux.)	117
7.11	Influence du paramètre γ sur la forme de la fonction de transfert. a) La fonction de transfert $f(\eta)$ et b) sa dérivée ($M = 1, m = 0$ et $\theta = 0, 1$).	119
7.12	Nombre de mots appris en fonction de γ lorsque le réseau est utilisé comme mémoire associative. ($M = 1, m = 0, \theta = 0, 01, nunits = 162, noutputs = 32$)	120
7.13	Extension possible du système.	124

Liste des tableaux

5.1	Distribution des classes dans la base de données.	83
5.2	L'information mutuelle $H(C; F)$ en fonction de la taille du vecteur de caractéristiques ($N_s \times N_d \times N_\theta$). (Le nombre de partitions est de 200.)	85
5.3	Taux de reconnaissance.	86
5.4	Résultats obtenus par la validation croisée ($k = 10$).	86
5.5	La matrice de confusion.	89
6.1	Distribution des classes dans la base de données.	96
6.2	Résumé de résultats pour la reconnaissance de la première lettre de mots cursifs.	98
6.3	Matrice de confusion pour la reconnaissance de la première lettre.	98
6.4	Résumé de résultats pour la reconnaissance de mots cursifs.	99
6.5	Matrice de confusion pour la tâche de reconnaissance de mots cursifs. . . .	102
6.6	Matrice de confusion pour la tâche de reconnaissance de mots cursifs. (Continuation)	103
7.1	Cas spéciaux de la fonction de transfert $f(\eta)$	119
7.2	Performance du système pour des mots de différentes longueurs.	123

Chapitre 1

Introduction

L'homme inventa l'écriture vers l'an 3200 avant J.C. [Pommier93]. C'est dans des sociétés en plein développement, où l'essor du commerce et l'urbanisation font naître de nouveaux besoins que naît l'écriture. Sept siècles plus tard, l'homme inventa ainsi un nouveau système d'écriture : l'alphabet. Ce système de 22 lettres remplace donc les milliers de signes utilisés précédemment. Grâce à l'écriture, nous pouvons aujourd'hui connaître la vie et les croyances des hommes du passé.

Actuellement, l'écriture joue un rôle essentiel dans tous les domaines de notre vie quotidienne. Une grande partie des objets qui nous entourent comporte des traces écrites : les panneaux indicateurs, les notices d'emploi des produits et bien sûr les livres.

De nombreuses études ont montré que les tâches développées par notre cerveau sont très complexes. A première vue, la lecture humaine et la reconnaissance d'objets semblent être sujettes à des processus similaires. Des études en neurophysiologie ont permis de découvrir que la couche visuelle primaire de notre cerveau possède des neurones répondant sélectivement à l'orientation, la longueur ou la direction de mouvement de segments de droite. A cet égard, certaines théories postulent que le système visuel construit une représentation bidimensionnelle de la scène visuelle (raw primal sketch) en termes de primitives primaires (bords, segments de droite, régions noires) [Hube68].

D'autre part, des études en imagerie cérébrale ont permis d'identifier différentes zones

du cerveau impliquées dans certaines fonctions cognitives. Par exemple, certains travaux ont montré que l'accès lexical¹ semble avoir lieu dans une zone spécifique de l'hémisphère gauche du cerveau (extrastriate visual cortex) [Kuri98]. Cette zone présente une forte activation pendant la lecture des mots ou des chaînes régulières de lettres, mais aucune activation pour des chaînes irrégulières de lettres, des lettres isolées ou des pseudo-lettres. Dans la lecture normale (de gauche à droite), l'hémisphère gauche est typiquement plus rapide que l'hémisphère droit. D'ailleurs, le temps de réponse de chaque hémisphère est influencé par la longueur du mot. Enfin, dans la lecture rapide, les lecteurs ont tendance à fixer la partie gauche du mot (preferred window position). Ceci pourrait indiquer que c'est l'hémisphère le plus rapide qui traite la plupart des lettres afin de minimiser le temps de fixation.

D'autres études ont montré qu'une représentation à base de contours est souvent suffisante pour la reconnaissance d'objets. A ce sujet, certaines études ont montré que les points caractéristiques de l'image (points dominants : extrémités de segments, extremums de courbure) constituent une représentation complète [Attn54]. En effet, des figures, formées à partir des points caractéristiques, sont parfaitement reconnues par des sujets humains. Une représentation à base de points caractéristiques constitue donc une représentation plus compacte de l'image. Quant à l'écriture cursive, il a été montré que la plupart de l'information discriminante est contenue dans la partie singulière du mot cursif² (ascendants, descendants et boucles) [Simo92]. Par ailleurs, les boucles constituent les primitives les plus informantes de la zone centrale du mot.

Des études en psychologie expérimentale ont permis de découvrir que la reconnaissance de caractères faisant partie de mots est généralement plus rapide que la reconnaissance de caractères isolés [Reic69]. Ce phénomène est connu sous le nom de "effet de supériorité de mot" (word superiority effect). Ceci met en évidence l'importance de l'information contextuelle pour la reconnaissance humaine de lettres et de mots. Afin de comprendre des phénomènes comme celui-ci, différents modèles de lecture, validés expérimentalement,

¹Le processus par lequel les lecteurs humains associent l'image d'un mot avec son identité.

²La partie singulière correspond aux parties montantes et descendantes des lettres et la partie régulière correspond aux lettres sans projection.

ont été proposés. Parmi ceux-ci, le modèle d'activation interactif (IAM) suscite un intérêt particulier car il est capable d'expliquer certains de ces phénomènes. Ce modèle a d'ailleurs été appliqué avec un certain succès à la reconnaissance de l'écriture cursive [Côté97].

La reconnaissance de l'écriture est une tâche très difficile que les lecteurs humains sont pourtant capables d'effectuer sans effort apparent. Par contre, les machines conçues pour accomplir cette tâche sont loin d'atteindre des performances comparables. Certaines études ont pourtant montré qu'il est possible d'atteindre des performances compétitives dans certaines applications, à condition que l'on dispose de suffisamment de données d'apprentissage [Kimu97, Kner98b, Kim00]. D'autre part, l'importance de ce problème est illustrée par le fait qu'un nombre important de travaux sur le sujet sont publiés chaque année et que des machines capables de lire des numéros de téléphone, des codes postaux, des montants littéraux sont utilisées par un nombre croissant d'entreprises dans le monde.

1.1 Le problème

A la différence de l'imprimé où les caractères sont bien alignés et généralement bien séparés verticalement, les caractères présentent de nombreux types de variations et sont en général liés, fragmentés ou superposés dans le cas du manuscrit.

Chaque personne écrit de façon différente un même caractère selon son style particulier d'écriture. Ces variantes d'un même caractère sont appelées "allographes". La vitesse d'écriture est aussi un facteur important car l'écriture peut devenir illisible à cause d'une vitesse excessive.

La hauteur et la position verticale des lettres est parfois très importante. Par exemple, la lettre 'l' peut être reconnue comme 'e' car ces deux lettres diffèrent essentiellement par leurs tailles respectives. Un autre exemple d'ambiguïté est montré dans la figure 1.1. Dans ce cas, la seule manière de donner une réponse sûre est de considérer le contexte de la phrase.

La variabilité de l'écriture dépend ainsi des facteurs suivants :

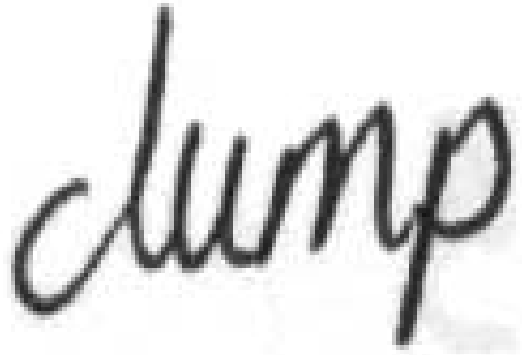


FIG. 1.1 – Ambiguïté de l’écriture. S’agit-il du mot “clump” ou du mot “dump” ?

- du style d’écriture de chaque personne,
- des caractéristiques et conditions des instruments d’écriture, et
- des variations de la qualité du papier.

De façon générale, on considère que la complexité de la tâche de reconnaissance dépend des facteurs suivants :

- la méthode d’acquisition,
- le type d’écriture,
- le nombre de scripteurs, et
- la taille du lexique.

1.1.1 Méthodes d’acquisition

Le premier critère de classification des systèmes de reconnaissance de l’écriture est basé sur la méthode d’acquisition des données. On distingue ainsi deux grandes catégories de systèmes :

- **systèmes en-ligne** (ou dynamiques). Ces systèmes utilisent comme donnée d’entrée une suite de coordonnées décrivant le mouvement d’un stylo électronique sur une tablette. Ces données sont représentées par une suite de positions de stylo échantillonnées à des intervalles de temps réguliers. Plus la vitesse est élevée, plus la distance entre points est grande. La reconnaissance peut s’effectuer au moment de saisie

de l'écriture ou ultérieurement. Après traitement, le système obtient ainsi une liste ordonnée de mots candidats. Cette liste est ensuite modifiée en considérant des informations syntaxiques ou sémantiques (**post-traitement contextuel**) afin d'obtenir une réponse unique.

- **systèmes hors-ligne** (ou statiques). Ces systèmes opèrent sur une image, usuellement binaire mais parfois en niveaux de gris, qui est généralement acquise à l'aide d'un scanner. L'information du tracé est extraite de l'image à l'aide de diverses méthodes de traitement d'images. Après traitement, le système obtient ainsi une liste ordonnée de mots candidats. Cette liste est traitée ultérieurement à l'aide d'information contextuelle afin d'obtenir une réponse unique. Certaines sociétés proposent actuellement des systèmes automatiques de traitement de formulaires, de chèques et de lettres [Trup97, Gors99, LeeCK99].

On considère que la reconnaissance hors-ligne est plus difficile pour les raisons suivantes :

- la détection de tracés. Un système hors-ligne doit chercher dans l'image les tracés décrivant le message écrit. Par contre, dans les systèmes en-ligne le tracé est déjà représenté comme une suite ordonnée de points.
- l'absence d'ordre et d'information temporelle. Dans les systèmes en-ligne, l'ordre du tracé aide à enlever en grande partie l'ambiguïté de la reconnaissance de certaines lettres.
- la variabilité de l'épaisseur du tracé. Ce problème ne se présente pas dans les systèmes en-ligne car dans ce cas les tracés sont filiformes.

1.1.2 Types d'écriture

Le type d'écriture à reconnaître constitue un autre critère de comparaison des systèmes [Tapp84]. On distingue les catégories d'écriture suivantes (Fig. 1.2) :

- **pré-casée** (*boxed discrete*). C'est l'écriture la plus contrainte, les scripteurs doivent écrire chaque caractère dans des cases spéciales, ce qui évite le problème de la localisation des caractères individuels (segmentation).

- **script** (*spaced discrete*). Cette écriture est moins contrainte que la précédente. Dans ce cas, les scribes doivent écrire chaque caractère individuellement en évitant la superposition.
- **continue** (*run-on discrete*). Dans ce cas, les scribes doivent écrire chaque caractère individuellement. Cependant, la superposition entre caractères est permise.
- **cursive** (*pure cursive script*). Dans ce cas, le mot doit être écrit d'un seul trait, sans levé de plume. Tous les caractères sont donc obligatoirement connectés.
- **mixte** (*mixed cursive and discrete*). Dans ce cas, les scribes ont la liberté de choisir leur style préféré d'écriture. C'est donc le type d'écriture le plus difficile, car toutes les variations provoquées par les scribes sont permises. Ce type d'écriture est aussi le plus courant à cause de sa simplicité et sa rapidité de tracé.

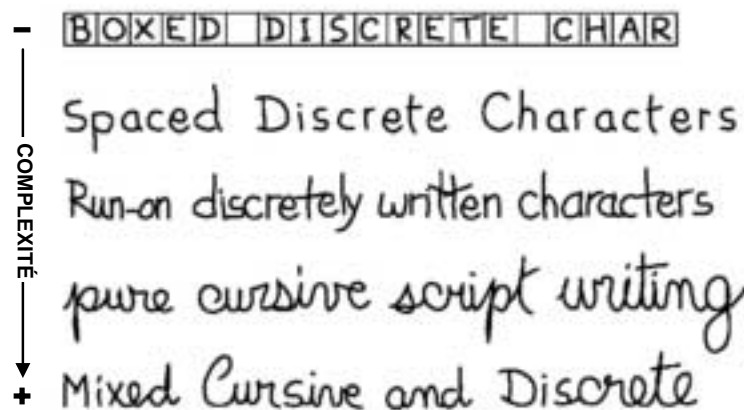


FIG. 1.2 – Types d'écriture [Tapp90].

1.1.3 Nombre de scribes

Chaque scribe possède un style particulier d'écriture. Pour cette raison, la reconnaissance de l'écriture de n'importe quel scribe (**système omni-scribe**) est plus difficile que la reconnaissance de l'écriture d'un seul scribe (**système mono-scribe**). Dans certains cas, il est même préférable de concevoir plusieurs sous-systèmes reconnaissant chacun un style particulier d'écriture (**système multi-scribe**).

Dans un système omni-scripteur, l'apprentissage se fait sur l'écriture de plusieurs scripteurs. Dans un système mono-scripteur l'apprentissage s'effectue sur l'écriture d'un seul scripteur, ce qui limite fortement l'ambiguïté et donne de meilleurs résultats.

1.1.4 Taille du lexique

Dans le cas de lexiques de grande taille le problème est dû principalement à l'existence de mots de plus en plus similaires. Plus il y a des mots similaires dans le lexique, plus il devient difficile de les discriminer. Pour cette raison, il est usuel de classer les systèmes par rapport à la taille du lexique.

On considère que la taille du lexique est :

- petite, lorsqu'elle est inférieure à 100 mots,
- moyenne, lorsqu'elle est comprise entre 100 et 500 mots,
- grande, lorsqu'elle est supérieure à 500 mots.

1.2 Applications

La reconnaissance hors-ligne de l'écriture cursive a des applications industrielles importantes dans le traitement automatique de formulaires, de chèques et de tri de courrier. Chaque jour, certaines entreprises doivent traiter un grand nombre de documents (lettres, chèques, formulaires, etc.) afin de satisfaire les besoins de leurs clients. L'utilisation de systèmes automatiques suscite donc un grand intérêt de la part des chercheurs, des industriels, et de tous ceux qui éprouvent le besoin de traiter des documents manuscrits. Certaines sociétés proposent actuellement des systèmes automatiques pour des applications telles que le traitement automatique de chèques, de lettres et de formulaires.

Le traitement d'un document consiste à identifier les différentes parties qui le composent, à reconnaître sa structure, à extraire chacun de ses composants et à reconnaître le message écrit.

1.2.1 Traitement de formulaires

Ce type d'application concerne typiquement le traitement de formulaires de déclaration d'impôts, de questionnaires, de bons de commande, de formulaires de recensement, etc. Il s'agit de systèmes plus ou moins complexes avec des performances relativement bonnes.

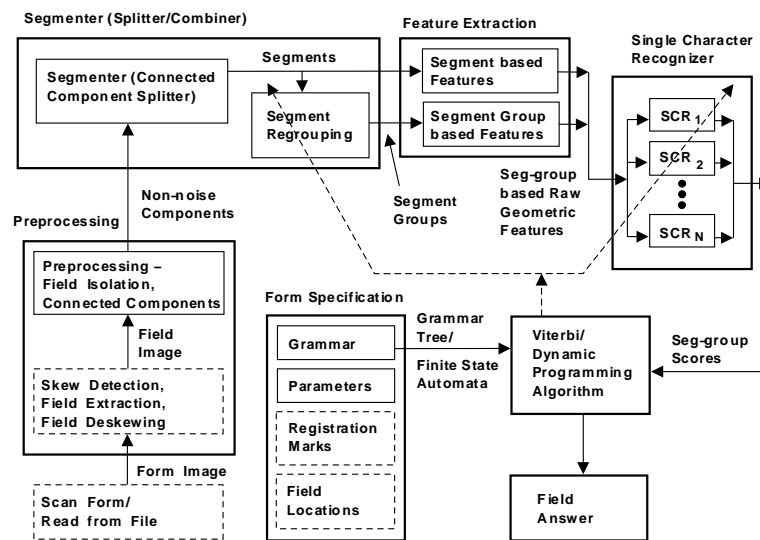


FIG. 1.3 – Traitement automatique de formulaires [Lam93].

Les formulaires se caractérisent par un ensemble prédéfini de champs. Chaque champ est usuellement encadré et son fond est imprimé de façon à faciliter la discrimination entre le texte et le fond. La structure du formulaire définit exactement le contenu de chaque champ (numérique), sa syntaxe (la date) et les relations entre champs (le nom de la société, son code, etc.).

La plupart des données à traiter comprennent des caractères manuscrits isolés mais aussi de l'écriture cursive (montants littéraux). La figure 1.3 montre le schéma d'un système de traitement automatique de formulaires [Lam93].

1.2.2 Traitement de chèques

Une application commerciale importante de l'écriture cursive hors-ligne est la lecture de chèques bancaires. Il s'agit de systèmes omni-scripteur à vocabulaire limité (environ 35 mots). A cause du grand nombre de chèques qui circulent chaque jour (plusieurs millions),

l'utilisation d'un système de ce type est donc très importante, même s'il n'est possible de vérifier que la moitié des chèques avec certitude.

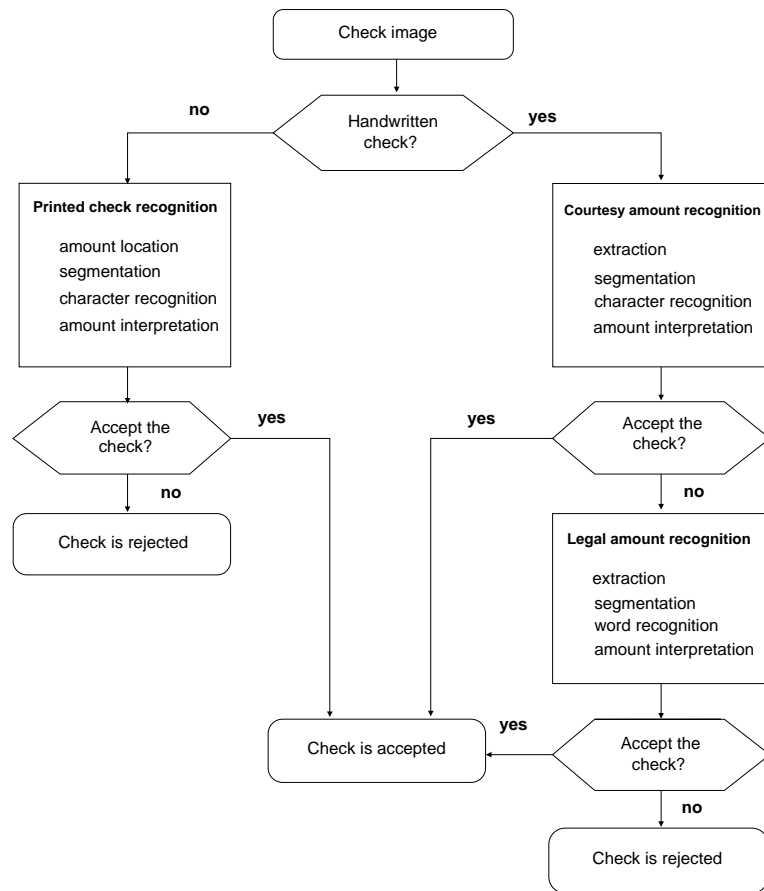


FIG. 1.4 – Traitement automatique de chèques [Gors99].

Certains systèmes de traitement de chèques se contentent de lire le montant numérique (courtesy amount) tandis que d'autres sont également capables de lire le montant littéral (legal amount) [Gors98] ou encore d'effectuer une vérification de signatures [Dima97]. Ces systèmes comprennent usuellement les étapes suivantes : localisation du montant, segmentation, reconnaissance et interprétation du montant. La figure 1.4 illustre un exemple de traitement automatique de chèques [Gors99].

1.2.3 Tri de courrier

La lecture d'adresses postales est considérée comme l'une des applications les plus abouties de la reconnaissance de formes. Il s'agit de systèmes hors-ligne de type omni-scripteur

à vocabulaire de grande taille (plusieurs milliers de mots). Cependant, il existe une redondance importante d'information entre le nom de ville et le code postal qui peut être exploitée afin d'améliorer leur performance.

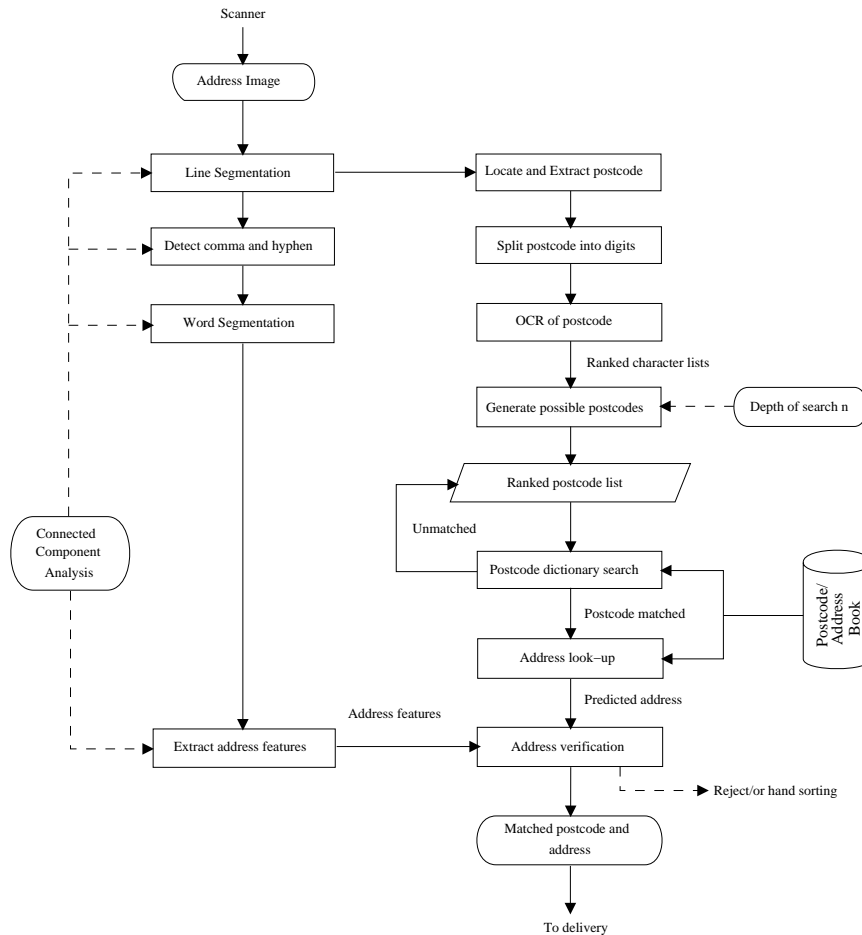


FIG. 1.5 – Traitement automatique de courrier [LeeCK99].

Une grande quantité de courrier est traitée chaque jour par les services postaux dans le monde. A titre d'exemple, en 1997 le service postal américain a dû traiter environ 630 millions de lettres par jour [Srih99]. Avec cette grande quantité de courrier, l'utilisation de systèmes automatiques de tri de courrier est donc primordiale. Des systèmes capables de reconnaître des caractères isolés sont déjà installés dans de nombreux bureaux de poste dans le monde, faisant partie de machines de tri de courrier. Une fois que le système a identifié le code postal sur l'enveloppe, le routage automatique du courrier peut être effectué. Des systèmes plus sophistiqués sont même capables de lire complètement l'adresse

du destinataire.

Les conditions requises pour cette application sont : une bonne vitesse (10 lettres/s), une bonne capacité (70 millions de lettres/jour) et un taux d'erreur réduit (moins de 1% pour l'adresse). Le texte à extraire peut comprendre : code postal, nom de ville, numéro et nom de rue, boîte postale, etc. Ces données peuvent comprendre soit du texte imprimé, soit du texte manuscrit isolé, soit du texte cursif.

Ce type d'application comprend les étapes suivantes : localisation de l'adresse, extraction de lignes de texte, de mots et de caractères, reconnaissance et post-traitement lexical. La figure 1.5 illustre un exemple de système de tri de courrier [LeeCK99].

1.3 Objectif de notre étude

L'objectif de notre étude est de développer et d'implémenter de nouvelles méthodes de reconnaissance hors-ligne de l'écriture cursive.

A partir d'une image binaire ou en niveaux de gris contenant un mot cursif, notre système doit reconnaître le mot inconnu parmi une liste de mots candidats. L'utilisation du modèle d'activation interactif constitue une option intéressante qui peut nous permettre de modéliser certains aspects de la reconnaissance humaine. D'ailleurs, ce modèle a été appliqué avec succès au problème de la reconnaissance de l'écriture cursive.

Comme la reconnaissance de caractères isolés est une tâche que les lecteurs humains sont censés être capables d'effectuer facilement, nous abordons dans un premier temps le problème de la reconnaissance de caractères cursifs segmentés. Par ailleurs, des méthodes basées sur des caractéristiques directionnelles ont obtenu des résultats très intéressants dans plusieurs études. Nous étudierons donc une méthode d'extraction de caractéristiques directionnelles biologiquement plausible basée sur une technique robuste de détection de droites.

Le phénomène de supériorité de mot peut être attribué en partie à la régularité de chaînes de lettres. D'ailleurs, les lettres externes sont mieux reconnues que les lettres in-

ternes. Nous considérerons certains de ces effets psychologiques dans la conception de notre système.

Bien que notre méthode de reconnaissance soit destinée à la reconnaissance hors-ligne de l'écriture, elle peut être adaptée pour d'autres tâches telles que la reconnaissance en-ligne, la reconnaissance d'autres systèmes d'écriture ou la reconnaissance de visages.

Nous disposons pour nos expériences de deux bases de données omni-scripteur contenant des images binaires de noms de villes et de montants littéraux provenant de chèques. La première base de données a été utilisée pour générer une base de données de caractères cursifs segmentés qui est destinée aux tests de nos méthodes de reconnaissance de caractères cursifs. La deuxième base de données est destinée aux différents tests de nos méthodes de reconnaissance de mots cursifs. Le nombre de classes de cette base est de 32. Nous sommes donc dans le cadre d'une application omni-scripteur à vocabulaire de petite taille.

1.4 Organisation de la thèse

L'organisation de la thèse est la suivante :

2. État de l'art de la reconnaissance de l'écriture.

Dans ce chapitre nous passons en revue les différentes approches de reconnaissance de mots et de caractères, ainsi que les différentes techniques utilisées par chacune de ces approches.

3. Psychologie de la reconnaissance de l'écriture.

Dans ce chapitre, nous passons en revue les différents modèles de lecture. Puis, nous présentons des exemples de systèmes de reconnaissance de l'écriture basés sur ces modèles.

4. Extraction de caractéristiques.

Dans ce chapitre, nous présentons une étude sur l'extraction de caractéristiques par l'utilisation de techniques résistantes au bruit et aux déconnexions du tracé. Nous présentons dans la première partie une brève introduction à la transformée de Hough.

Puis, nous présentons l'application de cette technique pour l'extraction de caractéristiques directionnelles. Finalement, nous présentons une méthode pour calculer différents types de caractéristiques biologiquement plausibles.

5. Reconnaissance de caractères cursifs segmentés.

Dans ce chapitre, nous présentons l'application d'une technique robuste de détection de droites au problème de la reconnaissance de caractères cursifs segmentés.

6. Reconnaissance holistique de mots cursifs.

Dans ce chapitre, nous présentons l'application d'une technique robuste de détection de droites au problème de la reconnaissance holistique de mots cursifs isolés.

7. Reconnaissance perceptive de mots cursifs.

Dans ce chapitre, nous présentons un résumé des différentes études concernant l'application de la psychologie de la lecture à la reconnaissance de l'écriture. Puis, nous présentons les fondements psychologiques de notre approche. Finalement, nous proposons un nouveau système perceptif de reconnaissance de mots cursifs.

Chapitre 2

État de l'art de la reconnaissance de l'écriture

2.1 Introduction

Les premiers systèmes automatiques de reconnaissance de l'écriture étaient des systèmes de lecture optique, c'est à dire, des systèmes de reconnaissance de caractères imprimés mono-fonte. Il existe aujourd'hui de nombreuses variantes commercialisées de ces appareils. La reconnaissance optique de caractères (OCR) étant ainsi aujourd'hui considérée comme pratiquement résolue.

Après le succès des recherches en OCR, le problème de la reconnaissance de l'écriture manuscrite a été abordé. Cependant, ce problème s'est avéré beaucoup plus difficile à résoudre. Cela est dû au fait que l'écriture manuscrite présente une variabilité très importante. De plus, le problème de l'identification de chaque caractère individuel conduit au délicat problème de la segmentation. Pour cette raison, les premiers travaux de recherche portaient essentiellement sur la reconnaissance de caractères manuscrits isolés.

Les premières approches consacrées à la segmentation de caractères utilisaient des techniques heuristiques de découpage. Pour cette raison, les erreurs de reconnaissance provenaient principalement de mauvaises segmentations. Des recherches ultérieures ont montré

que pour obtenir une segmentation correcte en caractères, il faut d'abord reconnaître le mot. Ce qui est paradoxal car pour reconnaître le mot il faut d'abord segmenter. La solution à ce problème a été donc de valider par la reconnaissance (de caractères ou de mots) chaque hypothèse de segmentation. Par exemple, le mot peut être segmenté en unités plus élémentaires qui peuvent être ensuite utilisées soit pour former des caractères reconnaissables, soit pour déterminer la segmentation optimale du mot à l'aide d'une technique de programmation dynamique.

2.2 Reconnaissance de mots

L'approche classique de reconnaissance de l'écriture est basée sur la décomposition de l'image en unités plus élémentaires [Case96] (**segmentation, dissection**).

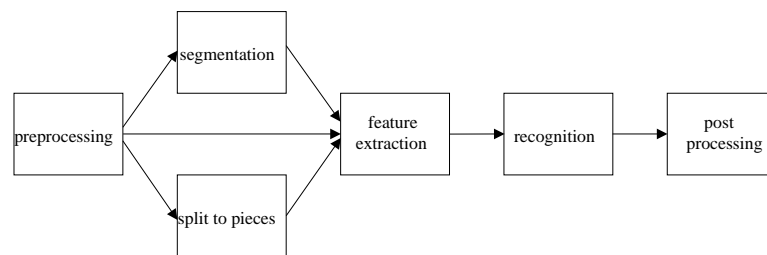


FIG. 2.1 – Schéma générique d'un système de reconnaissance de mots [Ste00].

Un système typique peut comprendre les étapes suivantes (Fig. 2.1) : pré-traitement, segmentation, extraction de caractéristiques, reconnaissance et post-traitement contextuel.

On distingue ainsi les approches suivantes [Case96] : **analytiques implicites**, **analytiques explicites**, **holistiques**, et **hybrides** (Fig. 2.2).

2.2.1 Approches analytiques explicites

Ces approches effectuent une segmentation basée sur des caractéristiques globales (dissection) suivie d'une classification des segments résultants. Dans ce cadre, la dissection est un processus intelligent au sens qu'elle effectue une analyse de l'image, mais sans effectuer une classification en symboles. Les frontières de segmentation sont déterminées en utilisant

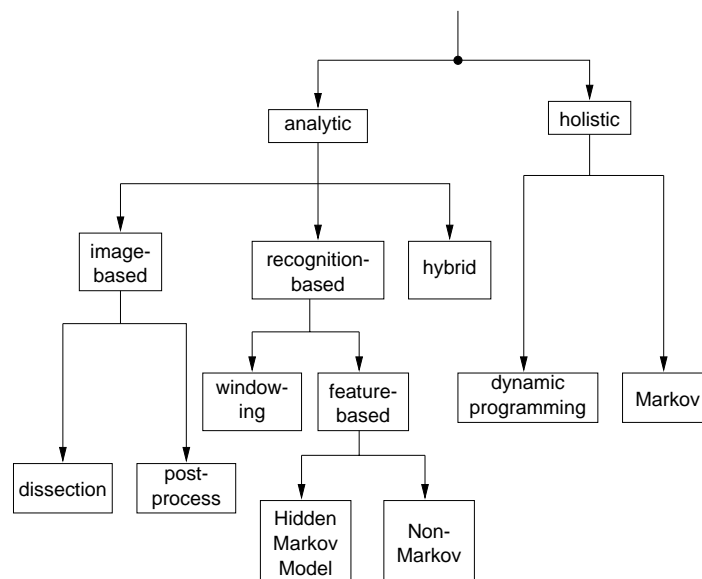


FIG. 2.2 – Approches possibles [Case96].

des mesures telles que : taille, largeur, espacement, position par rapport à la ligne de base, etc.

En général, il existe deux types de techniques de dissection : dissection directe en caractères et dissection avec post-traitement contextuel.

1. **Dissection directe en caractères.** Cette technique transforme l'image directement en symboles individuels (caractères). Selon les caractéristiques de l'écriture à reconnaître, différents types d'analyse sont possibles : analyse d'espacement, analyse des projections ou analyse des composantes connexes. Les deux premières techniques sont principalement applicables à l'imprimé tandis que la dernière technique est également applicable au manuscrit.

- **Analyse d'espacement.** Cette technique s'applique lorsque l'espacement entre caractères est pratiquement uniforme. Dans ce cas, les frontières de segmentation sont déterminées en se basant sur l'estimation de cette valeur quasi-constante [Henn68].
- **Analyse de projections.** La projection verticale (appelée aussi histogramme vertical) d'une ligne de texte consiste à compter les pixels noirs de chaque colonne (Fig. 2.3). Elle peut être utilisée pour estimer soit l'espacement entre caractères, soit pour détecter des tracés verticaux ou des lignes de texte. La projection verticale

présente usuellement un minimum au point de chevauchement de deux caractères. Donc, les colonnes où la valeur de la projection est inférieure à un seuil donné constituent des candidats potentiels pour la dissection de l'image [Ohta93].

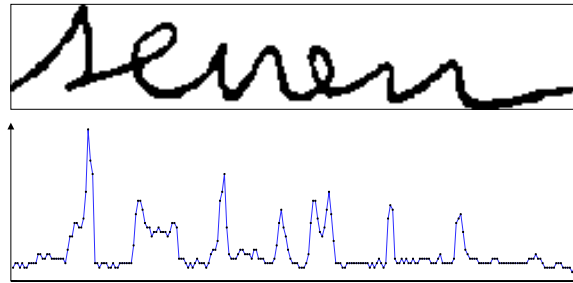


FIG. 2.3 – Projection verticale.

- **Analyse de composantes connexes.** Les techniques précédentes sont généralement inapplicables dans le cas de polices proportionnelles ou de caractères manuscrits. L'analyse d'espacements est peu efficace lorsque la largeur des caractères est variable tandis que l'analyse de projections donne des résultats médiocres lorsque les caractères sont inclinés. La segmentation de l'écriture manuscrite implique une analyse bidimensionnelle de l'image car il est souvent difficile de séparer deux caractères par de simples droites (verticales ou inclinées). L'analyse de composantes connexes consiste à déterminer les régions noires ou "composantes connexes" (connected components, blobs). Ces composantes connexes sont ensuite fusionnées ou divisées pour former des caractères en analysant leurs boîtes englobantes (Fig. 2.4) [Cesa90]. L'analyse de composantes connexes est en général plus effective et plus rapide que l'analyse de projections.
2. **Dissection avec post-traitement contextuel.** La segmentation obtenue par les techniques précédentes peut être vérifiée à l'aide d'une technique de post-traitement contextuel (modèles de Markov). L'image est alors divisée en sous-images (graphèmes, caractères, pseudo-caractères) qui sont ensuite reconnues individuellement [Sayr73]. Dans ce cas, l'étape de dissection est appelée "pré-segmentation" [Leco91]. En pratique, un caractère se décompose au maximum en deux ou trois graphèmes, et un

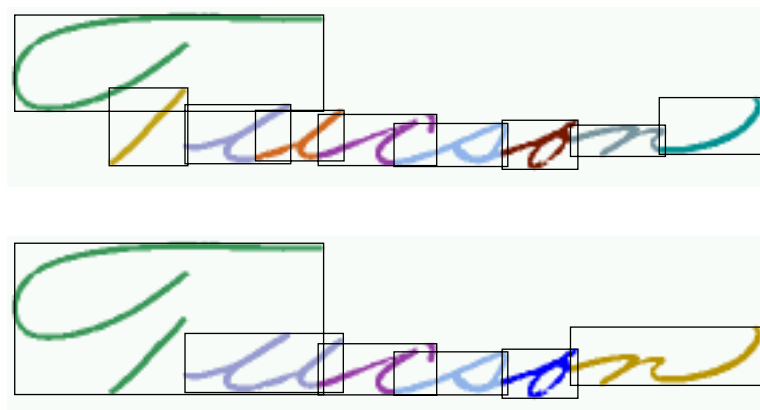


FIG. 2.4 – Analyse de composantes connexes. En haut, segmentation en graphèmes [Leco91]. En bas, segmentation optimale en lettres. Les lettres 'T', 'u', 'n' ont été obtenues chacune par la fusion de deux graphèmes.

graphème représente au plus un à deux caractères. L'inconvénient principal de cette technique est que la segmentation reste fortement dépendante de l'efficacité de la méthode de dissection.

2.2.2 Approches analytiques implicites

Dans ces approches, l'image est divisée systématiquement en parties (avec ou sans superposition) sans tenir compte de leur contenu. Ces parties sont ensuite utilisées pour trouver un résultat de segmentation cohérent avec la reconnaissance. La segmentation en lettres est donc un sous-produit de la reconnaissance de mots pouvant être guidée par une analyse contextuelle.

L'intérêt principal de ces méthodes est qu'elles évitent le délicat problème de la segmentation. Aucun algorithme complexe de dissection n'est nécessaire et les erreurs de reconnaissance sont dues principalement à des erreurs de classification. Pour cette raison, cette approche a été appelée aussi reconnaissance sans segmentation (**segmentation-free**).

On distingue deux types d'approches : recherche dans l'image et segmentation d'une représentation de caractéristiques.

1. **Recherche dans l'image.** Cette technique consiste, comme son nom l'indique, à chercher dans l'image des formes reconnaissables qui permettront de donner le

meilleur score de reconnaissance. Par exemple, un réseau de neurones peut être utilisé pour trouver un ensemble de points de segmentation possibles (Shortest Path Segmentation) [Burg92]. Puis, la recherche de la segmentation optimale peut être obtenue en minimisant une fonction de coût à l'aide de techniques basées sur la programmation dynamique (Fig. 2.5).

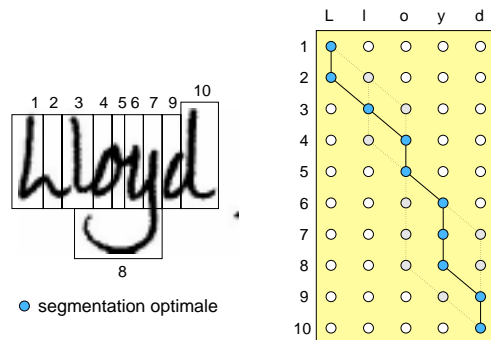


FIG. 2.5 – Segmentation à l'aide de la programmation dynamique.

2. **Segmentation d'une représentation de caractéristiques.** Ces approches se divisent en : approches Markoviennes et approches non-Markoviennes.

- **Approches Markoviennes.** Ces approches modélisent les variations de l'écriture par des processus stochastiques (modèles de Markov cachés, HMM). Les observations constituent donc une représentation séquentielle des caractéristiques de l'image d'entrée. Ces caractéristiques sont usuellement mesurées de gauche à droite. Ceci facilite la représentation du mot par les modèles de caractère qui le constituent (Fig. 2.6). La segmentation est donc réalisée (implicitement) par la recherche de la suite d'états qui explique au mieux la suite d'observations. Les modèles de Markov exploitent le fait que les lettres n'ont pas toujours de frontières de segmentation distinctes. Ce qui fait qu'une dissection "parfaite" n'est pas possible dans le cas général. Plusieurs modèles de lettres peuvent ainsi être combinés pour former soit plusieurs modèles discriminants, soit un seul modèle à chemin discriminant [Gilli92, ChenMY93, Seni93, Berc93]. Les HMMs à chemin discriminant sont capables de manipuler de grands vocabulaires, mais ils sont généralement moins précis

que les HMMs à modèle discriminant.

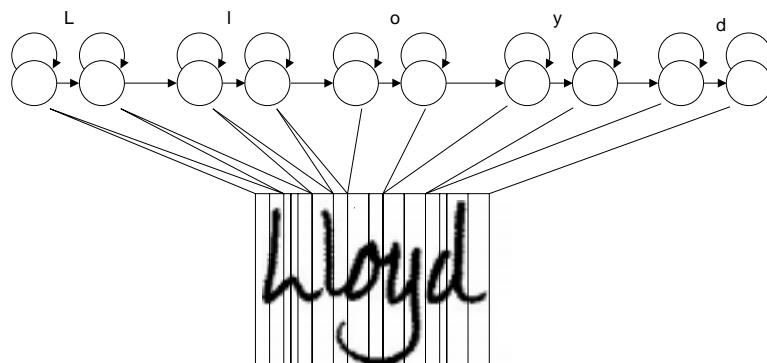


FIG. 2.6 – Modèles de Markov cachés (HMM).

- **Approches non-Markoviennes.** Dans ces approches, chaque primitive détectée dans l'image et sa position sont codées dans un graphe de primitives. La segmentation est alors effectuée par la mise en correspondance avec des graphes prototypes à l'aide d'une technique de programmation dynamique [Tapp82, Simo92] ou d'une technique de relaxation [Haye80]. La mise en correspondance peut être effectuée soit au niveau des lettres, soit au niveau des caractéristiques primaires ou secondaires à l'aide d'une étape descendante de vérification (backward matching) [Leco93].

2.2.3 Approches holistiques

Les approches holistiques modélisent le mot comme un tout [Earn62]. Elles ne cherchent donc pas à identifier des lettres séparées. Les approches holistiques comprennent généralement deux étapes : extraction de caractéristiques (globales et/ou locales) et reconnaissance holistique du mot inconnu.

L'utilisation des approches holistiques est habituellement limitée aux cas où le lexique est prédéfini et fixé. En effet, la reconnaissance est forcément limitée à un lexique spécifique de mots puisque ces méthodes ne prennent pas les lettres en considération. Ce point est particulièrement critique lorsque l'apprentissage s'effectue sur un grand nombre d'échantillons. Une étape d'apprentissage est ainsi obligatoire pour augmenter ou modifier le lexique. Cette propriété rend ces méthodes plus appropriées pour des applications à lexique statique, telles

que la reconnaissance de chèques.

Ces approches peuvent être utilisées pour la reconnaissance en-ligne sur un ordinateur personnel (ou un bloc-notes électronique), l'algorithme de reconnaissance étant alors adapté à l'écriture d'un utilisateur spécifique ainsi qu'au vocabulaire particulier concerné.

2.2.4 Approches hybrides

Ces méthodes combinent plusieurs stratégies de reconnaissance afin d'exploiter les avantages d'approches complémentaires. En général, un algorithme de dissection est appliqué à l'image, mais le but est de "sur-segmenter". Autrement dit, le but est de découper l'image en un nombre suffisant d'endroits tel que les frontières correctes de segmentation soient incluses dans l'ensemble des découpages faits. La classification est ensuite utilisée pour évaluer les différentes hypothèses et pour choisir la meilleure segmentation en considérant toutes les combinaisons possibles de segments (graphèmes) à l'aide d'une technique de programmation dynamique ou de modèles de Markov cachés [Fuji92, Kimu92, Breu94, Case92a]. Certaines études ont montré que ce type d'approches est capable de donner d'assez bonnes performances pour la reconnaissance de mots cursifs [Kner98b, Yaco99, Kim00].

2.2.5 Exemples de systèmes

Vu le grand nombre de systèmes qui ont été présentés dans la littérature, nous limitons notre discussion à deux études récentes sur la reconnaissance de mots cursifs isolés. Pour une discussion extensive des différents systèmes, on pourra se reporter à [Tapp90, Leco93, Gillo94, Case96, Weso96, Liu00].

- Reconnaissance de montants littéraux par l'utilisation de modèles de Markov cachés et de réseaux de neurones [Kner98b]. D'abord, chaque image est traitée afin de corriger l'angle d'inclinaison. Puis, une méthode de segmentation explicite est utilisée afin de diviser l'image en graphèmes. Ensuite, chaque graphème obtenu est traité afin d'obtenir un ensemble de 74 caractéristiques géométriques. La segmentation opti-

male est obtenue en utilisant un modèle de Markov caché ergodique. Les probabilités d'observation sont estimées à l'aide d'un réseau de neurones statique à trois couches. Ce système a été testé sur une base de données de mots cursifs isolés de montants littéraux de chèques français contenant 170.000 images dont la base d'apprentissage contient 130.000 images et la base de test contient environ 40.000 images binaires. La performance estimée du système pour la reconnaissance de mots est de l'ordre de 92,9%. Un détail intéressant de ce système est qu'il ne semble pas reconnaître les segments obtenus en tant que graphèmes car sa performance pour la reconnaissance de graphèmes est très mauvaise (48,4%). Par ailleurs, comme le système n'effectue pas une segmentation en lettres, il est adapté pour des lexiques de petite taille. Par comparaison, la reconnaissance du système de base (HMM [Kner98a]) est de l'ordre de 73,3% pour un ensemble d'apprentissage contenant 11.121 mots.

- Reconnaissance de mots cursifs par la combinaison de deux classifieurs [Kim00]. Le premier classifieur utilise une méthode de segmentation implicite à l'aide d'un réseau de neurones multi-couche. Le deuxième classifieur utilise une méthode de segmentation explicite (en graphèmes) et une méthode de reconnaissance à l'aide de modèles de Markov cachés. Chaque image est alors segmentée en graphèmes et traitée afin d'obtenir un ensemble de caractéristiques. Chaque suite de graphèmes est transformée en une suite de symboles selon la composition de chaque graphème en primitives (ascendants, descendants et boucles). Les résultats obtenus par ces deux méthodes sont combinés par une technique de combinaison de classifieurs afin d'obtenir une réponse unique. Les méthodes ont été testées sur une base de données de mots cursifs isolés correspondant à des montants littéraux de chèques anglais. La base d'apprentissage contient 5.223 images et la base de test contient 2.482 images. La performance de la méthode de segmentation implicite à l'aide de réseaux de neurones est de l'ordre de 86.0% tandis que la performance de la méthode de segmentation explicite et de reconnaissance à l'aide de modèles de Markov cachés est de l'ordre de 82.0%. La performance combinée est de l'ordre de 92,2%.

2.3 Reconnaissance de caractères

La reconnaissance du manuscrit fait partie, avec la reconnaissance de l'imprimé, de ce qu'il est convenu d'appeler l'OCR (Optical Character Recognition), qui traite donc de la reconnaissance optique de caractères.

La reconnaissance de caractères se décompose en deux sous-problèmes liés mais distincts : l'extraction de caractéristiques et la classification.

Avant de les reconnaître, les images sont encore sous une forme brute (différentes hauteurs, largeurs, contrastes, inclinaisons, ...) inexploitable pour la plupart des systèmes de classification [Lamy95]. Le codage vise donc à normaliser cette entrée (l'image) afin d'obtenir une représentation adéquate pour la classification (un vecteur de caractéristiques de taille fixe).

Même s'il n'existe pas de méthode d'extraction de caractéristiques "idéale", nous pouvons énumérer quelques unes des qualités qui font une bonne méthode :

1. le pouvoir discriminant. Autrement dit, les caractéristiques doivent comporter suffisamment d'information afin de pouvoir distinguer les différentes classes,
2. la résistance au bruit (de numérisation). Les caractéristiques d'un caractère avec ou sans bruit doivent être sensiblement les mêmes,
3. la résistance aux déformations. C'est à dire, les caractéristiques doivent résister aux différentes variations naturelles des caractères ainsi qu'aux variations provoquées par les différents scripteurs (variations de style),
4. la compacité. L'extraction de caractéristiques est aussi une compression de l'information. En général, la performance des méthodes de classification est influencée par la dimension du vecteur de caractéristiques.

2.3.1 Les diverses approches

De nombreux critères de classification des systèmes de reconnaissance de caractères ont été proposés. Selon le type de technique utilisée [Gail79, Mant86, Leco90, ?], on distingue

les catégories suivantes :

1. **Mise en coïncidence** (Template matching). Ces méthodes réalisent des comparaisons point à point des pixels du caractère considéré avec des modèles de référence (templates).
2. **Transformations globales** (Fourier, Hadamard, Karhunen-Loève, moments, etc.). Ces méthodes effectuent un changement de l'espace de représentation afin de faciliter la détection par une réduction de sa dimensionnalité [Shri84b].
3. **Recherche d'intersections avec des droites**. Ces méthodes cherchent à capturer la topologie du caractère en calculant le nombre d'intersections du tracé avec des droites horizontales ou verticales fixées à l'avance ou mobiles sur tout le caractère suivant des techniques heuristiques. Par exemple l'approche "characteristic-loci" cherche, à partir du même principe, à étiqueter les zones blanches du caractère [Gluc67].
4. **Extraction de caractéristiques locales**, (des extrémités de segments, des points anguleux, des jonctions, des croisements, etc.) au moyen de masques fixes (gabarits) ou variables selon plusieurs méthodes. L'extraction de ce type de primitives peut être facilitée par une squelettisation préalable.
5. **Calcul de pentes et de rayons de courbure**, par balayage horizontal, utilisation de masques ou suivi de contour.
6. **Analyse de concavités et de boucles**, par suivi de contour, ou après une approximation polygonale ou détermination de l'enveloppe convexe ou d'autres constructions géométriques [Kaha87]. Les concavités peuvent aussi être caractérisées par un étiquetage de zones blanches [Komo77].
7. **Méthodes structurelles**, qui insistent sur l'aspect de description globale du caractère à partir de primitives élémentaires, et en particulier des méthodes basées sur des représentations à base de graphes, ou qui utilisent des grammaires [Ali77].

Ces approches se divisent en deux grandes catégories : statistiques et structurelles.

1. **Approches statistiques.** Les approches statistiques effectuent un changement de l'espace de représentation : un ensemble de mesures permet d'obtenir à partir d'une forme donnée un vecteur correspondant dans l'espace de représentation. Le problème consiste alors à trouver la classe de l'objet inconnu à partir de ce vecteur. On crédite généralement les approches statistiques d'une certaine robustesse et d'une grande souplesse d'adaptation à de nouvelles données grâce à la notion d'apprentissage et à un traitement global du problème. L'adaptativité est une propriété essentielle dans le domaine du manuscrit libre, mais la globalité de ces méthodes a un inconvénient : elles ne peuvent que difficilement prendre en compte des caractères très particuliers ou qui sont en nombre très limité dans la base pour justifier la création d'une classe spécifique dans l'apprentissage.
2. **Approches structurelles.** Les approches structurelles décrivent une forme complexe à partir de l'ensemble de formes élémentaires qui la composent. Elles s'attachent donc à décrire l'organisation géométrique ou, si l'on préfère, la structure de la forme, plutôt que ses caractéristiques métriques. Les approches structurelles sont plus proches de nos concepts naturels et conviennent donc mieux pour distinguer des cas particuliers. Par contre, la prise en compte a posteriori de nouveaux types de caractères, non prévus initialement, peut s'avérer lourde à gérer en absence d'apprentissage (en particulier si la méthode s'appuie sur une structure arborescente de règles de décision déterminées manuellement).

2.3.2 Extraction de caractéristiques

Les techniques précédentes ont pour but de capturer les caractéristiques distinctives des caractères. Ces caractéristiques se divisent en : caractéristiques globales, caractéristiques structurelles, caractéristiques morphologiques, caractéristiques de contour, caractéristiques statiques, caractéristiques métriques et caractéristiques adaptatives.

2.3.2.1 Caractéristiques globales

On parle de caractéristiques globales lorsque le codage ne fait pas intervenir la position spécifique d'éléments particuliers de l'image. L'image est considérée globalement sans chercher à distinguer les différentes zones. Parmi les méthodes faisant intervenir une transformation globale de l'image on trouve : l'analyse en composantes principales ou la transformée de Karhunen-Loève [Grot93], la transformée de Gabor [Garr91], la transformée de Fourier, etc.

2.3.2.2 Caractéristiques structurelles

Les caractéristiques structurelles ont pour but de capturer la structure ou la forme du caractère. On cherche donc à détecter dans l'image des éléments tels que : des boucles, des points de rebroussement [Case92b] ou les directions principales du tracé [Chha93].

La difficulté principale de ces approches est, le plus souvent, de rendre ces caractéristiques insensibles aux différentes variations que doivent subir les caractères. On est en fait ramené à un sous problème de reconnaissance : identifier les "sous-formes" de l'image.

Au contraire des caractéristiques globales, les caractéristiques structurelles ont la propriété de localité. C'est à dire, les caractéristiques s'attachent très spécifiquement aux différentes zones de l'image.

2.3.2.3 Caractéristiques morphologiques

L'extraction des caractéristiques morphologiques s'appuie sur une étude des positions relatives des différentes composantes noires et blanches de l'image. On décrit alors le caractère en termes de composantes blanches et noires, de cavités (parties blanches partiellement entourées de noir) et de boucles (parties blanches entièrement entourées de noir) (Fig. 2.7).

La définition de cavité est assez simple : une cavité Ouest est une région de points blancs, entourée par le tracé dans les trois autres directions (Nord, Sud et Est) [Mitc89]. On peut définir de la même manière d'autres types de cavités, mais aussi les cavités centrales (qui sont entourées de tous côtés par le tracé). La détection des boucles (c.f. [Chha93]) requiert

usuellement l'utilisation d'un algorithme de recherche de composantes connexes (souvent plus coûteux à mettre en œuvre).

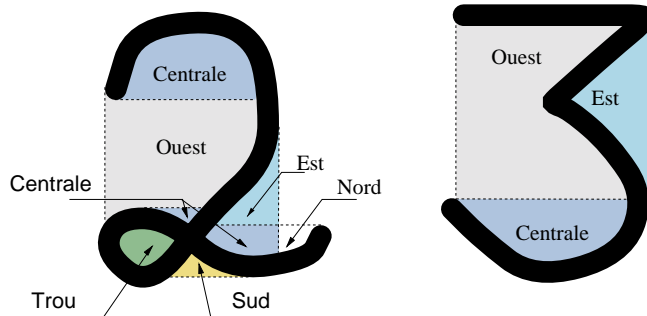


FIG. 2.7 – Caractéristiques morphologiques [Mitc89].

La détection des caractéristiques morphologiques peut être effectuée par des opérateurs morphologiques de dilatation selon les quatre directions, et d'intersection d'images. Une fois que toutes les régions ont été détectées, on doit coder leur présence et leur taille dans les différentes régions de l'image [Gade94].

2.3.2.4 Caractéristiques de contour

Le contour du caractère est défini par la séquence (ou les séquences dans le cas d'un caractère composé de plusieurs tracés) des pixels entourant le tracé. Une fois le contour repéré, on peut soit garder sa description explicite (le codage de Freeman ou des représentations équivalentes [Thép94]), soit l'utiliser pour en déduire des composantes particulières (les points de rebroussement, ou encore certaines des caractéristiques morphologiques présentées plus haut), soit encore l'utiliser sous forme de représentation codée (par des moments de Fourier, ou de Zernike) [Milg90].

La description par contour est proche d'une description en squelette. La squelettisation consiste à transformer l'image en supprimant l'épaisseur du tracé, en réduisant ainsi le caractère au tracé ayant généré l'image.

L'utilisation de ce type de caractéristiques présente cependant certains problèmes. D'une part, le codage du contour dépend fortement du point de départ de l'algorithme de suivi : le contour est en fait une chaîne bouclée de points. D'autre part, il faut savoir s'affranchir du bruit de l'image, par exemple en lissant le contour extrait. Enfin, pour certains caractères, le suivi peut amener à la détection de plusieurs contours (comme dans la lettre 'e'). Il faut alors distinguer les contours intérieurs et extérieurs du caractère. Ce qui nous amène à la détection des boucles fermées.

2.3.2.5 Caractéristiques statistiques

On cherche ici à représenter le caractère par des mesures statistiques de l'image. On peut par exemple utiliser la distribution des pixels dans différentes régions de l'image [Bure92], ou bien des histogrammes (nombre de points noirs par colonne, par ligne, ou dans d'autres directions). Des méthodes plus complexes comme l'analyse en composantes principales peuvent également produire ce type de caractéristiques. De même, on peut calculer le nombre de pixels présentant une caractéristique particulière dans différentes régions de l'image (Fig. 2.8).

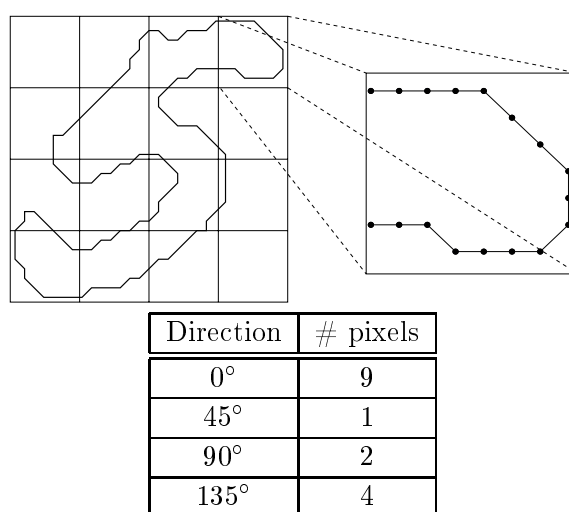


FIG. 2.8 – Caractéristiques statistiques [Itoh90].

2.3.2.6 Caractéristiques métriques

Cette catégorie comprend des caractéristiques basées sur des mesures physiques de l'image. Outre des caractéristiques assez simples, comme hauteur, largeur et rapport de ces deux grandeurs, on peut utiliser des caractéristiques plus complexes comme le codage des profils (Fig. 2.9). Le profil gauche d'une image est exprimé par les distances au bord gauche du carré englobant le caractère. On peut définir les profils par rapport aux quatre axes naturels (gauche, droit, haut et bas), mais aussi par rapport à d'autres directions (des profils à 0° , 45° , 90° et 135°) [Bure92].

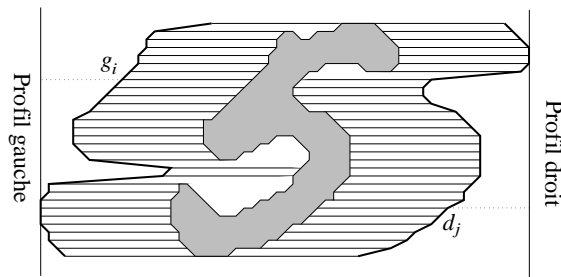


FIG. 2.9 – Caractéristiques métriques.

2.3.2.7 Caractéristiques adaptatives

L'inconvénient des approches précédentes réside sur le choix adéquat des caractéristiques à utiliser. En effet, il est possible que les caractéristiques utilisées ne soient pas suffisantes pour différencier certaines classes. Par exemple, si l'utilisation des profils gauche et droit permet de différencier des chiffres, en revanche, elle ne permet pas de différencier certaines lettres comme "M" et "H".

Les caractéristiques adaptatives sont obtenues directement de l'image et requièrent une phase d'apprentissage. Autrement dit, le système opère sur une représentation proche de l'image d'origine [Ridd96] et doit lui-même construire et optimiser l'extracteur de caractéristiques (Fig. 2.10).

Cette stratégie est utilisée notamment dans les systèmes TDNN [LeCu90], dans le neocognitron [Fuku82] ou encore dans des systèmes à base de réseaux de Kohonen, le système

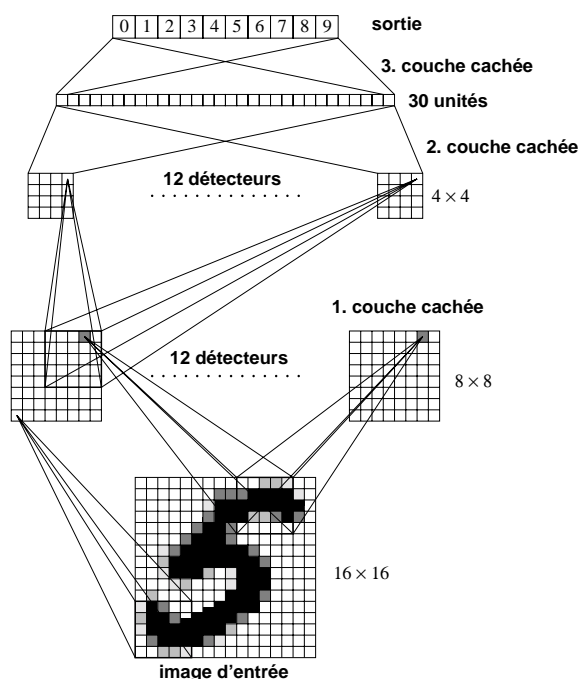


FIG. 2.10 – Caractéristiques adaptatives.

LASSO [Idan92]. Dans la plupart de ces systèmes, on représente donc les caractères normalisés dans une image de taille fixe (rétine).

2.3.3 Techniques de classification

Les techniques de classification se divisent en : statistiques et neuronales, stochastiques, structurelles et hybrides.

2.3.3.1 Statistiques et neuronales

Dans ces techniques, la reconnaissance consiste à trouver la classe ayant la plus forte probabilité d'appartenance. Les méthodes utilisées peuvent être : estimation Gaussienne, séparation linéaire, k-plus proches voisins, etc.

Les réseaux de neurones statiques utilisant un codage 1 parmi n sont capables d'approcher des probabilités a posteriori et sont usuellement plus performants que les méthodes statistiques [Ney95].

La classification statistique comprend : calcul d'un vecteur de caractéristiques représen-

tatif des primitives analysées et classification de ce vecteur.

La classification par réseaux de neurones peut être appliquée directement sur l'image ou après extraction de caractéristiques.

2.3.3.2 Stochastiques

Dans ces techniques, le caractère est représenté par une suite d'observations exprimant les différentes progressions du stylo. Dans le cas hors-ligne, cette modélisation est plus difficile car l'information dynamique n'existe pas. Cependant, il est possible de la simuler en fixant un sens particulier de description. Par exemple, il est possible d'explorer l'image à différents angles d'inclinaison en utilisant des techniques telles que la transformée de Hough [Elli92]. Une approche différente consiste à traiter l'image à l'aide de filtres de détection de bords afin d'obtenir une liste de points décrivant l'orientation du tracé (points de croisement, extrémités de segments) [Lall97]. Cette liste de points est ensuite filtrée (filtrage de Kalman) afin d'obtenir une représentation non-bruitée de la trajectoire du tracé.

Chaque classe de caractère est donc modélisée par un modèle de Markov caché comprenant : une séquence d'observations (primitives ou symboles) et une séquence d'états. La reconnaissance consiste à trouver le modèle (HMM) ayant la plus forte probabilité d'appartenance.

2.3.3.3 Structurelles

Dans ces techniques, la structure du caractère est représentée en termes de primitives structurelles (primitives de rebroussement, de changement d'orientation, de croisement, d'accroissement ou de décroissance de pente).

Les techniques structurelles sont les suivantes [Bela92] :

1. **Méthodes syntaxiques.** Chaque caractère est représenté par une phrase dans un langage spécial de primitives. Les caractères d'une même famille étant représentés par une même grammaire, la reconnaissance consiste à déterminer si la phrase a été

générée par la grammaire. Par exemple, la distance d'une phrase à une grammaire est définie comme la distance minimale de modifications que doit subir la phrase pour qu'elle soit proche de la grammaire (distance d'édition) [Wagn74].

2. **Appariement de graphes ou d'arbres** : calcul d'une distance relationnelle et étiquetage consistant. Les caractères sont représentés par des graphes de primitives. La reconnaissance est donc effectuée par la comparaison du caractère à reconnaître avec des modèles de référence à l'aide d'une technique de programmation dynamique ou d'une méthode de relaxation probabiliste.
3. **Arbres de décision** (méthodes de test). Ces méthodes consistent à appliquer sur l'image des tests de plus en plus fins sur la présence ou l'absence de certaines primitives afin de séparer les caractères en plusieurs catégories. Donc, la reconnaissance consiste à diviser, à chaque test, l'ensemble de choix possibles jusqu'à n'obtenir qu'un seul choix correspondant à l'identité du caractère reconnu.

2.3.4 Exemples de systèmes

A cause du grand nombre de systèmes qui ont été décrits dans la littérature, une discussion exhaustive est au delà de nos objectifs. Pour cette raison, nous nous contentons ici de citer quelques exemples.

- Reconnaissance de caractères cursifs isolés [Mohi94]. Dans ce cas, chaque caractère est représenté par une image binaire de taille 16×24 . Le système extrait deux types de caractéristiques : des caractéristiques locales dérivées de la direction du contour et des caractéristiques locales dérivées des extremums de courbure et des extrémités de segments. Dans le premier cas, l'image est divisée en tranches rectangulaires parallèles à plusieurs orientations afin d'obtenir 88 caractéristiques. Dans le deuxième cas, l'image est divisée en zones rectangulaires afin d'obtenir 96 caractéristiques. Ce système a été appris sur une base de données contenant 24.388 caractères segmentés et testé sur une base de données contenant 15.248 caractères segmentés (NIST especial database 3). L'ensemble de classes comprend 26 lettres minuscules. La performance

estimée du système est de 94,43% en utilisant un classifieur hybride.

- Reconnaissance de caractères cursifs segmentés [Kimu97]. Dans ce cas, le système extrait deux types de caractéristiques : des caractéristiques de contour et des caractéristiques dérivées des gradients de l'image. Le système utilise deux types de classifieurs : un classifieur à base de réseaux de neurones et un classifieur statistique. Ce système a été testé sur une base de données contenant 43.678 images binaires de caractères cursifs segmentés. Cette base a été obtenue par la segmentation semi-automatique d'une base de données contenant des images de mots cursifs d'adresses postales (USPS). La performance estimée du système est de 83,87% en utilisant un classifieur statistique avec des vecteurs de caractéristiques de dimension 400. Par comparaison, la performance des lecteurs humains sur la même base est de l'ordre de 85,85%. La bonne performance de ce système est due probablement au grand nombre d'échantillons d'apprentissage utilisés.
- Le système LeNet [LeCu90]. Ce système utilise une technique adaptative d'extraction de caractéristiques. Il comprend plusieurs niveaux consécutifs d'extracteurs. Son application à la reconnaissance de chiffres manuscrits s'avère très efficace. En revanche, ce système est assez complexe (au moins 98.448 connexions, seulement pour la reconnaissance de chiffres manuscrits!!) et requiert une phase d'apprentissage assez prolongée (plusieurs mois). Chaque chiffre manuscrit est représenté par une image en niveaux de gris de taille fixe (16×16). La base de données d'apprentissage contient 7.291 images de chiffres manuscrits et 2.549 images de chiffres imprimés tandis que la base de test contient 2.007 images de chiffres manuscrits et 700 images de chiffres imprimés. La performance du système est de l'ordre de 96,6%.

2.4 Conclusions

Dans ce chapitre nous avons présenté les différentes approches de reconnaissance de l'écriture. La segmentation constitue l'un des problèmes principaux auxquels doivent faire

face les systèmes de reconnaissance. D'une part, les approches mixtes sont capables d'atteindre des performances très intéressantes et sont particulièrement adaptées pour des applications à vocabulaire de grande taille. D'autre part, les approches holistiques présentent des performances très honorables et sont particulièrement adaptées pour des applications à vocabulaire fixe de petite taille.

Le domaine de la reconnaissance de l'écriture est actuellement un domaine largement développé. Nonobstant les nombreuses études, aucune solution générale à ce problème n'est encore connue. Dans certaines applications, il est pourtant possible d'atteindre des performances comparables à la performance humaine, à condition que l'on dispose de suffisamment de données d'apprentissage.

Chapitre 3

Psychologie de la reconnaissance de l'écriture

3.1 Introduction

Le processus de lecture humaine a suscité depuis longtemps un intérêt particulier de la part des chercheurs en sciences cognitives et en reconnaissances des formes. Cela est dû au fait que les lecteurs humains sont capables de lire pratiquement n'importe quel texte, même dans des conditions difficiles.

Un des résultats importants issus des recherches en psychologie de la lecture est que la lecture de chaînes de caractères est aussi rapide que la lecture d'un caractère isolé [Reic69]. Ceci met en évidence l'importance de l'information contextuelle pour la reconnaissance humaine.

Bien que le processus de lecture ne soit pas encore bien connu, on considère actuellement qu'il comporte les étapes suivantes [Tayl83] :

1. Un processus global. Il s'agit d'un processus rapide qui dure entre 50 et 100 ms et qui traite le mot de façon globale.
2. Un processus au niveau des lettres (**letter-based**). Les identités des lettres commencent à être disponibles après 50 ms d'exposition. Les lettres extérieures sont les

premières à être identifiées et peuvent être utilisées pour ajuster les premières hypothèses ou pour en générer d'autres.

3. Un processus **scan-parse**. C'est un processus plus lent qui utilise les identités des lettres afin de produire une description phonétique du mot écrit. Cette description peut être utilisée comme évidence additionnelle pour l'identification du mot.

3.2 Modèles d'accès lexical

Le processus par lequel les lecteurs humains associent l'image d'un mot avec son identité est connu sous le nom de "**accès lexical**".

On considère que le processus d'accès lexical présente les caractéristiques suivantes :

1. Automatisation. L'accès lexical est un processus qui a lieu de façon automatique sans demander de l'attention de la part du lecteur (étape pré-attentive).
2. des effets de fréquence : les mots fréquents sont identifiés plus rapidement que les mots peu fréquents.
3. des effets de contexte : les mots sont identifiés plus rapidement lorsqu'ils sont présentés dans un contexte de phrase.

Il existe principalement deux types de modèles d'accès lexical [Taft93] : les modèles de recherche et les modèles d'activation.

3.2.1 Modèles de recherche (Search models)

Dans ces modèles, le lexique est exploré de façon séquentielle, par ordre de fréquence, afin de détecter une entrée du lexique coïncidant avec le stimulus visuel [Fors76]. Comme il est peu probable que la liste complète des mots soit examinée chaque fois que l'image d'un mot doit être traitée, la recherche est limitée à un sous-ensemble du lexique appelé "bin". Les effets de fréquence sont alors attribués au fait que la recherche d'un mot fréquent est effectuée sur un ensemble réduit de mots.

Les bins sont définis sur la base de caractéristiques sensorielles (p. e., tous les mots commençant par 's' et se terminant par 'e') et sont choisis avant le déclenchement du processus de recherche.

Les modèles de recherche utilisent deux ensembles de représentations qui sont respectivement compatibles avec l'information orthographique (c.-à-d. la forme visuelle du mot) et l'information phonologique (c.-à-d. la forme auditive du mot). Le fichier d'accès orthographique est exploré de façon séquentielle chaque fois qu'une chaîne de lettres est présentée visuellement. Le mot vérifié par rapport à ce stimulus est, en cas de concordance avec la chaîne de lettres, pris comme le mot recherché. En cas d'échec, la recherche continue dans le fichier d'accès orthographique afin de trouver un meilleur candidat.

Un exemple de ce type de modèle est décrit dans [Fors81]. Ce modèle comporte un moteur de résolution de problèmes (GPS) qui est associé à des modules lexicaux, syntaxiques, et sémantiques. Le traitement linguistique est strictement ascendant. Les effets de fréquence sont dus à l'ordre de recherche et les effets de contexte sont dus à la prise de décision "post-lexicale" effectuée par le GPS.

3.2.2 Modèles d'activation (threshold models)

Dans ces modèles, le mot est identifié une fois que son niveau d'activation a atteint un certain seuil. Les effets de fréquence sont alors attribués au fait que les mots fréquents ont des niveaux d'activation plus élevés que les mots peu fréquents. Les effets de contexte sont attribués au fait que des mots similaires atteignent des valeurs proches (après saturation).

Un modèle de ce type est décrit dans [Seid89]. Ce modèle reçoit en entrée la description orthographique et produit en sortie la description phonologique. Le modèle est appris par rétro-propagation. Les auteurs affirment que ce modèle est capable d'expliquer un certain nombre d'effets (effets de contexte, de fréquence, de régularité orthographique-phonologique, le temps de prononciation et la dyslexie).

3.3 Les différents modèles

Les premiers modèles de lecture proposés par les psychologues étaient basés sur des processus de recherche. Les modèles les plus importants sont : le modèle pandémonium, le modèle de recherche (search model), le modèle d'activation (threshold model), le modèle logogen, le modèle de vérification, le modèle d'activation interactif, le modèle à deux voies et le modèle triangulaire.

3.3.1 Le modèle pandémonium

Le modèle pandémonium consiste en une hiérarchie de détecteurs appelés “démons” [Self59] opérant en parallèle (Fig. 3.1).

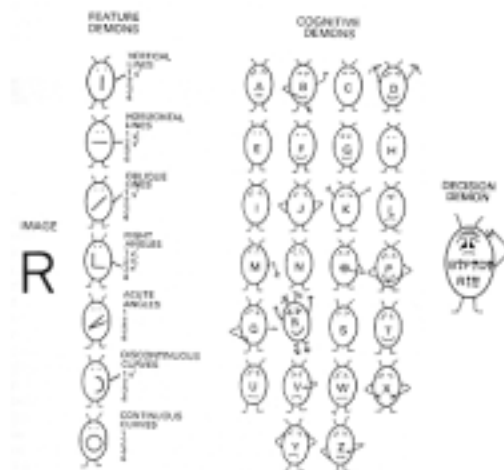


FIG. 3.1 – Le modèle pandémonium [Self59].

Ce modèle comprend généralement les niveaux suivants :

1. **niveau de démons de caractéristiques.** Dans ce niveau, des démons de caractéristiques cherchent à détecter un certain type de segments : angles, contours, etc.
2. **niveau de démons cognitifs.** Dans ce niveau, des démons cognitifs cherchent à détecter certaines combinaisons de caractéristiques,

3. **niveau de démons de décision.** Dans ce niveau, des démons de décision cherchent à détecter les démons cognitifs les plus actifs.

Après présentation du stimulus visuel (une lettre), certains démons de caractéristiques sont activés. Ceux-ci activent à leur tour certains démons cognitifs. Par exemple, le démon cognitif représentant la lettre 'A' est activé par les démons correspondant aux segments diagonaux et au segment horizontal. Enfin, le démon de décision examine les activations des démons cognitifs et choisit le démon cognitif le plus actif afin de reconnaître la lettre. Ce modèle a été utilisé principalement pour modéliser la reconnaissance humaine d'objets.

3.3.2 Le modèle Logogen

C'est le premier modèle qui incorpore la notion d'activation. Son architecture est très similaire à celle du modèle de recherche, sauf pour le mécanisme d'accès au lexique (Fig. 3.2).

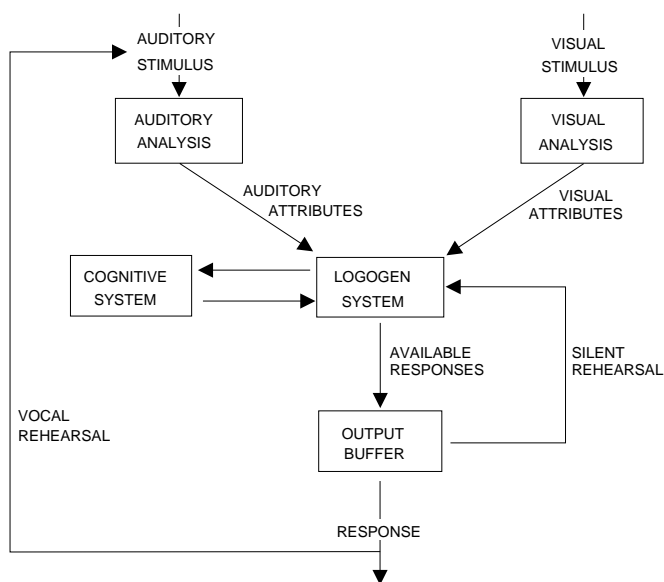


FIG. 3.2 – Le modèle Logogen [Mort69].

Chaque entrée du lexique est représentée par un dispositif collecteur d'évidence ou **logogen** [Mort69]. A chaque logogen est associé un seuil indiquant le niveau d'activation nécessaire pour reconnaître le mot associé. Chaque seuil dépend de la fréquence du mot de

telle sorte que les mots fréquents aient des seuils plus grands que les mots peu fréquents. Ceci simule le fait que les lecteurs identifient plus rapidement les mots fréquents (Forster et Chambers, 1973). Les logogens sont activés de façon croissante chaque fois que les caractéristiques du stimulus visuel sont proches de celles des mots qu'ils représentent. Une fois qu'un logogen a atteint un seuil d'activation prédéterminé, le mot est disponible comme réponse.

Par exemple, si on présente le mot HOUSE comme stimulus d'entrée, tous les logogens qui représentent les mots commençant par H recevront de l'activation. Il en va de même pour les mots finissant par E, les mots de 5 lettres, les mots contenant 'OU' et ainsi de suite. Ceci a pour effet d'activer les logogens correspondant aux mots HOUSE et HORSE. Mais le logogen correspondant au mot HOUSE est activé plus fortement car il est également activé par toutes les caractéristiques d'entrée.

3.3.3 Le modèle de vérification

Ce modèle comprend les étapes suivantes : génération d'un ensemble sémantique, recherche dans l'ensemble sémantique, génération d'un ensemble sensoriel, et recherche dans l'ensemble sensoriel [Beck79].

Dans la première étape, le modèle Logogen est utilisé pour obtenir deux ensembles initiaux de candidats. Puis, chaque ensemble est vérifié de façon séquentielle contre chaque stimulus visuel afin de trouver le meilleur candidat.

Les effets de fréquence sont expliqués par l'ordre de recherche et les effets de contexte sont expliqués par le fait que la recherche est effectuée d'abord sur l'ensemble sémantique.

3.3.4 Le modèle d'activation interactif

Ce modèle utilise une architecture connexionniste à trois couches (primitives, lettres et mots), organisées hiérarchiquement [McCl81] (Fig. 3.3).

Il a été conçu originalement pour expliquer l'effet de supériorité de mot dans le cadre

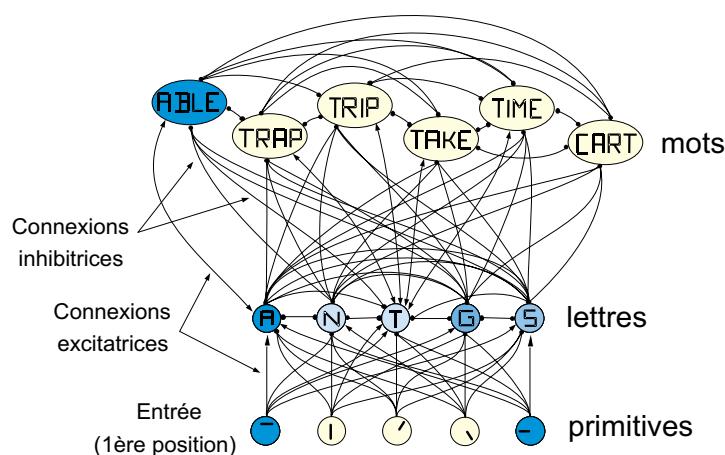


FIG. 3.3 – Le modèle d'activation interactif [McCl81].

d'une tâche simplifiée de reconnaissance de mots imprimés de 4 lettres. Les primitives, les lettres et les mots sont tous représentés par des neurones.

Chaque lettre est représentée par un ensemble de 16 primitives, chacune représentant un segment de droite avec une orientation particulière (Fig. 3.4).



FIG. 3.4 – Codage des lettres.

Son fonctionnement repose sur les faits suivants :

1. la perception comporte plusieurs niveaux de traitement, chaque niveau correspondant à un niveau d'abstraction différent,
2. la perception comporte des traitements parallèles de l'information visuelle, et

3. la perception comporte des processus “interactifs”, c’est-à-dire des processus ascendants et descendants.

Les connexions entre couches sont excitatrices ou inhibitrices et s’organisent comme suit :

1. aucune connexion entre neurones-primitives,
2. connexions inhibitrices ou excitatrices entre neurones-primitives et neurones-lettres,
3. connexions inhibitrices entre neurones-lettres d’une même position,
4. connexions inhibitrices ou excitatrices bidirectionnelles entre neurones-lettres et neurones-mots (le neurone-mot “WORK” est excité par le neurone-lettre “K” de la quatrième position mais est inhibé par le neurone-lettre “R” de la quatrième position),
5. connexions inhibitrices entre neurones-mots.

Ce modèle a été repris ultérieurement pour la modélisation de la reconnaissance auditive de mots (Fig. 3.5) et pour la reconnaissance de mots cursifs [Côté97].

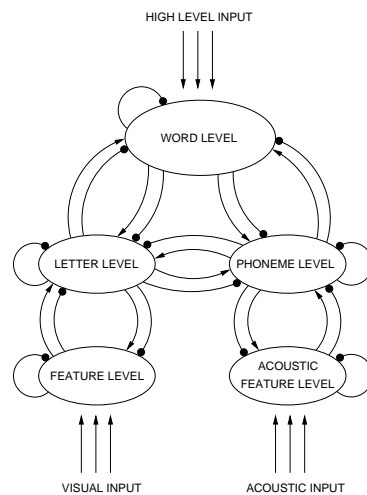


FIG. 3.5 – Reconnaissance visuelle et auditive de mots [McC181].

3.3.5 Le modèle à deux voies en cascade (DRC model)

Ce modèle comporte deux voies d’accès au lexique [Colt94] : une voie visuelle directe et une voie phonologique indirecte (Fig. 3.6).

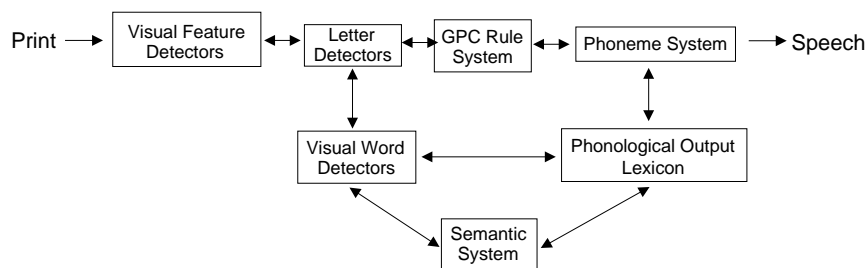


FIG. 3.6 – Le modèle à deux voies en cascade (DRC model) [Colt94].

Ces deux voies sont accédées au moment de la lecture d'un mot, mais la voie directe est généralement plus rapide. La voie phonologique n'influe donc sur le traitement que lorsque la voie directe est très lente ou aboutit à un échec.

Ce modèle utilise le même type de modélisation des lettres que le modèle d'activation interactif (IAM) [Rume74] (Fig. 3.4). Par exemple, le mot "CART" est représenté comme `CART`.

3.3.6 Le modèle triangulaire

Ce modèle utilise une architecture connexionniste de traitement distribué (PDP models) [Seid89] (Fig. 3.7). Il a principalement été conçu pour expliquer certains troubles d'ordre neurologique tels que la dyslexie.

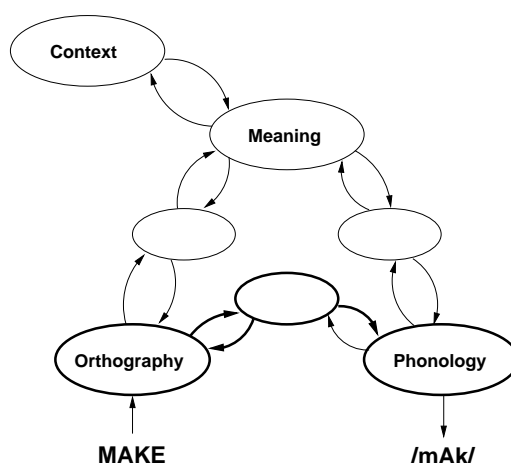


FIG. 3.7 – Le modèle triangulaire [Seid89].

Les mots sont codés en trois voies séparées, mais interactives : la voie orthographique,

la voie phonologique, et la voie sémantique. Chaque mot est ainsi représenté par trois composantes.

A la différence du modèle à deux voies, ce modèle ne fait aucune supposition quant à la nature des processus impliqués dans chaque voie. Cependant, celles-ci sont toutefois simulées par des connexions entre les différentes représentations. Les neurones de la voie sémantique sont activés pendant la prononciation car le but habituel est de transmettre des informations sémantiques. Même pendant la dictée, qui peut être effectuée principalement à base d'information phonologique, la signification du mot est automatiquement activée.

Ce modèle a été repris ultérieurement pour la modélisation de la reconnaissance auditive de mots [Harm98].

3.4 Systèmes de reconnaissance de l'écriture

Le succès des recherches en psychologie de la lecture a motivé le développement de plusieurs systèmes de reconnaissance de l'écriture cursive, notamment le système Nuscript, le système BLIRTNET et le système PERCEPTO.

1. Le système BLIRTNET est basé sur des modèles de traitement distribué (PDP) [Moze87]. Il partage donc des points communs avec le modèle CID (Connection Information Distributor). A la différence du modèle d'activation interactif, le traitement est strictement ascendant (bottom-up). Par ailleurs, ce modèle est capable de reconnaître plusieurs mots à la fois et explique certains effets psychologiques (transposition de lettres, incertitude dans la position de lettres et de mots, etc.). Les primitives utilisées sont : des segments verticaux, horizontaux et diagonaux et des extrémités de segments.
2. Le système décrit dans [Hull86] est basé sur le modèle de vérification. Ce système comprend les étapes suivantes : génération d'hypothèses par l'analyse holistique de l'image et vérification d'hypothèses qui utilise les hypothèses générées afin d'effectuer une analyse détaillée de l'image. Dans l'étape de génération d'hypothèses, le système

utilise un nombre réduit de caractéristiques afin d'effectuer une réduction de lexique. Dans l'étape de vérification d'hypothèses, le système utilise ce sous-ensemble de mots afin de construire un arbre dont les nœuds représentent les hypothèses à vérifier en termes de primitives. Les primitives utilisées comprennent : ascendants, descendants, points de 'i' ou de 'j', barres verticales courtes, etc. Ce système a été testé dans une tâche de reconnaissance de caractères imprimés.

3. Le système Nuscript est basé sur le modèle de vérification [Higg91]. Il utilise des ascendants et des descendants comme primitives de base afin d'effectuer une réduction de lexique. A chaque mot du lexique est associé un score représentant sa valeur d'activation. A la différence du modèle Logogen, chaque unité lexicale ne possède pas son propre mécanisme d'agrégation d'évidence. Le système simule cependant un effet similaire par l'utilisation d'une étape d'extraction de caractéristiques et d'une étape spéciale d'ajustement de scores. Dans l'étape d'extraction de caractéristiques, le stimulus d'entrée et des caractéristiques déjà détectées sont utilisés afin de déduire de nouvelles caractéristiques. Ceux-ci sont alors utilisés pour ajuster le score de chaque unité lexicale. Ainsi, des mots similaires au mot correct accumuleront des scores plus élevés que les autres mots. En particulier, le mot correct doit finalement présenter le meilleur score. Les caractéristiques utilisées sont : longueur, rotondité de l'extrémité gauche du mot et nombre de points de 'i'. Ce système a été testé dans une tâche de reconnaissance hors-ligne de mots manuscrits.
4. Le système décrit dans [Leco94b] est basé sur un processus descendant de vérification (rétro-appariement). A la différence des systèmes précédents, il considère trois niveaux de traitement : le niveau primitives, le niveau lettres et le niveau mots. Dans un premier temps, le système détecte les primitives contenues dans l'image (traits, boucles fermées et boucles ouvertes) afin d'obtenir une représentation symbolique du mot. Pendant la phase de reconnaissance, pour un ensemble de mots hypothèses, le système cherche à mettre en correspondance chacune de ces hypothèses avec le mot contenu dans l'image. Ces mots hypothèses sont décrites sous forme d'une liste de

lettres elles mêmes décrites en termes de primitives. Puis, la mise en correspondance s'effectue de l'extérieur vers l'intérieur du mot (comme dans la reconnaissance humaine) et suivant l'ordre d'importance des lettres. Ce système a été testé dans une tâche de reconnaissance hors-ligne de mots cursifs isolés.

5. Le système PERCEPTO est basé sur le modèle d'activation interactif et sur le modèle de vérification [Côté97]. A la différence du modèle d'activation interactif, ce système est adapté pour la reconnaissance de mots cursifs de longueur quelconque. La variabilité dans la largeur des lettres est prise en compte par l'utilisation d'un score flou de position qui dépend de la longueur de chaque mot. Après présentation d'une image à l'entrée du système, les primitives primaires (ascendants, descendants et boucles fermées) sont d'abord détectées. Les primitives extraites sont alors présentées à l'entrée du module de reconnaissance de mots afin de générer un ensemble initial de mots candidats. Cet ensemble permet d'établir des hypothèses qui sont ensuite vérifiées dans l'image initiale. Le système établit après plusieurs cycles une liste de mots candidats triés par ordre décroissant d'activation. Le mot ayant obtenu la plus forte activation correspond ainsi au résultat de la reconnaissance. Ce système a été testé sur une base de données contenant 2.929 images de mots cursifs isolés. La base de données d'apprentissage contient 184 images de mots cursifs isolés correspondant à des montants littéraux de chèques. La performance estimée du système est de l'ordre de 73,6%, sans rejet.

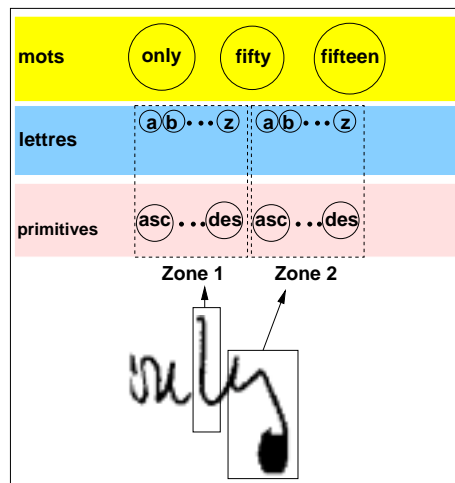


FIG. 3.8 – Le système PERCEPTO [Côté97].

Chapitre 4

Extraction de caractéristiques

Les lettres sans extension (ascendants et descendants) sont usuellement les plus difficiles à reconnaître [Tay183]. D'ailleurs, la détection d'ascendants et de descendants est relativement facile comparée à la détection d'autres types de primitives (boucles, cavités, extrémités de segments, points caractéristiques, courbes). D'autre part, l'inconvénient des approches adaptatives comme celles basées sur des réseaux de type LeNET est dû au fait qu'elles requièrent l'utilisation de très grandes bases de données. Nous devons donc développer nos propres méthodes d'extraction de caractéristiques. Comme notre but principal est de simuler certaines caractéristiques du système visuel humain, nous croyons qu'il est plus convenable de développer une méthode capable d'opérer sur des images binaires ou en niveaux de gris. Même si nous ne disposons que d'images binaires pour nos expériences.

4.1 Introduction

L'utilisation d'une représentation à base de contours permet de faciliter la tâche d'extraction de l'information pertinente tout en conservant suffisamment d'information de la scène visuelle. A cet égard, certaines théories postulent que le système visuel construit une représentation bidimensionnelle de la scène visuelle (primal sketch) en termes de caractéristiques primaires (bords, segments de droite, régions noires) [Marr80]. Un grand nombre

de techniques de détection de bords et de droites ont été donc proposées afin d'obtenir cette représentation. D'ailleurs, la détection automatique de droites est une tâche très importante pour un grand nombre d'applications. Par exemple, la détection de routes et de chemins de fer dans des images aériennes, de lignes de texte et de tracés dans des images textuelles, etc.

Certaines études expérimentales ont montré que la perception de l'orientation d'un segment de droite pourrait avoir lieu de façon globale [Pizlo97, Popple00]. C'est à dire que la perception de l'orientation d'un segment de droite est perturbée par la présence d'autres formes dans le champ visuel. Ce phénomène semble être à l'origine de nombreuses illusions optiques (Fig. 4.1) [Boum68, Avery69, Masin83]. D'autres études ont montré que les droites sont en général mieux perçues que d'autres formes et que le temps de réponse est indépendant de la longueur [Uttal75, Pizlo97]. Nous allons donc étudier dans ce chapitre une méthode globale pour modéliser la perception humaine de droites. Nous allons voir comment cette méthode peut être généralisée pour la détection d'autres caractéristiques plus complexes.

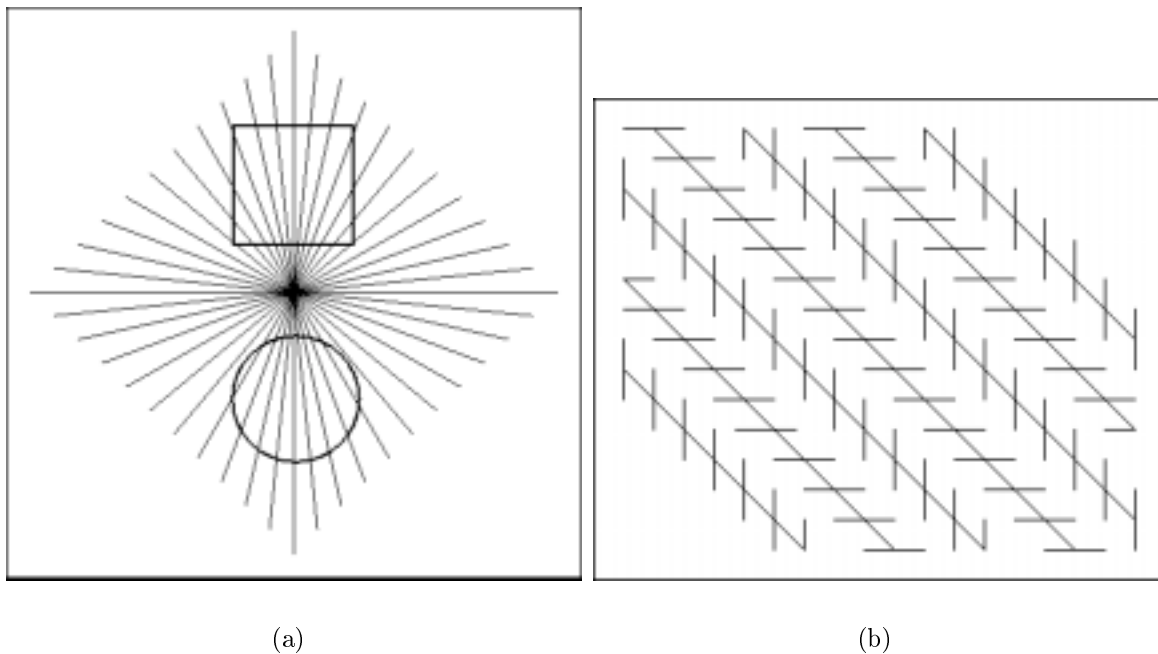


FIG. 4.1 – Illusions optiques. a) Cette figure présente un carré et un cercle parfaits qui semblent être déformés. b) Les droites diagonales sont en réalité parallèles.

Le bruit contenu dans l'image peut réduire dramatiquement les performances des méthodes classiques de traitement d'images et de reconnaissance des formes. Les techniques de détection de droites ne font pas exception à la règle. Comme les méthodes locales sont souvent sensibles au bruit et aux points singuliers (outliers), il peut être utile de recourir à l'utilisation de méthodes globales. La transformée de Hough constitue une technique globale très efficace qui a été utilisée avec succès dans de nombreuses applications.

4.1.1 Transformée de Hough

La transformée de Hough classique (TH) est une technique de détection de droites dans des images binaires [Ball81]. Par cette technique, une droite ℓ est transformée en un point de coordonnées (ρ, θ) représentant ses paramètres (distance et orientation), voir Fig. 4.1.

Soit ℓ une droite, soit θ son angle par rapport à la partie positive de l'axe X , soit ρ sa distance à l'origine. La forme normale de la droite ℓ est donnée par :

$$\rho = x \cos \theta + y \sin \theta \quad (4.1)$$

Pour un seul pixel de coordonnées (x, y) , cette équation décrit une sinusoïde dans l'espace de paramètres $\rho - \theta$. Donc, dans le cas idéal, les trajectoires correspondant à des alignements de pixels doivent s'intersecter au point commun (ρ_c, θ_c) . La détection de droites consiste ainsi à rechercher ces points d'intersection.

Afin de calculer la transformée de Hough, il est en général nécessaire de discrétiser l'espace de paramètres. Donc, la trajectoire sinusoïdale correspondant à chaque pixel noir de coordonnées (x, y) est discrétisée et ses pixels accumulés dans une matrice $h(\rho, \theta)$. Comme cette matrice présente des pics aux points d'intersection des trajectoires sinusoïdales, la détection de droites devient une opération de détection de pics dans la matrice $h(\rho, \theta)$.

Soit $f(x, y)$ une image en niveaux de gris, soient N_ρ et N_θ les nombres de niveaux de discrétisation des axes ρ et θ respectivement. La transformée de Hough $h(\rho, \theta)$ peut être

représentée par :

$$h(\rho_j, \theta_k) = \sum_{x,y} f(x, y) \delta(\rho_j - x \cos \theta_k - y \sin \theta_k) \quad (4.2)$$

avec

$$\theta_k = \frac{\pi k}{N_\theta}$$

$$\rho_j = \frac{j \rho_m}{N_\rho}$$

et

$$\delta(t) = \begin{cases} 1 & \text{si } t = 0 \\ 0 & \text{si } t \neq 0 \end{cases}$$

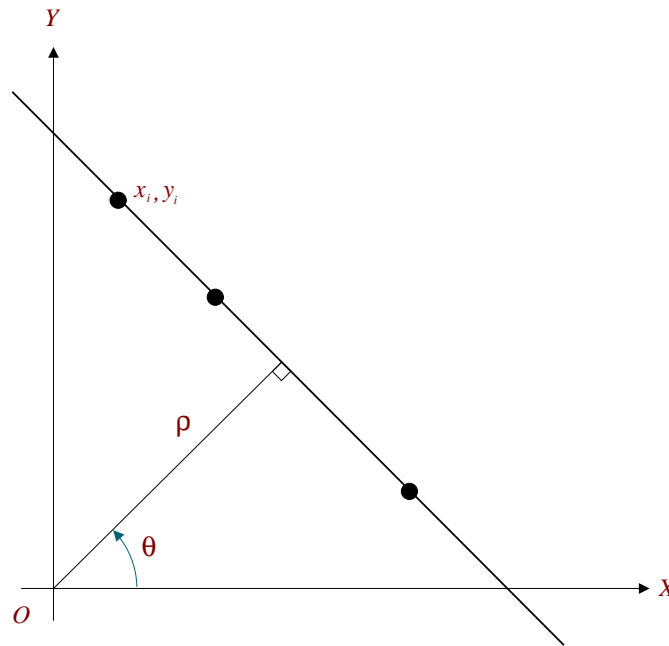


FIG. 4.2 – Paramétrisation d'une droite.

L'algorithme de calcul de la transformée de Hough classique d'une image binaire procède comme suit :

1. pour chaque pixel noir de coordonnées (x_i, y_i) ,
2. pour chaque $k \in \{0, 1, \dots, N_\theta - 1\}$,

3. calculer ρ_k par :

$$\rho_k = x_i \cos \theta_k + y_i \sin \theta_k$$

avec

$$\theta_k = \frac{\pi k}{N_\theta}$$

4. actualiser l'accumulateur $h(\rho_k, \theta_k)$ par :

$$h(\rho_k, \theta_k) \leftarrow h(\rho_k, \theta_k) + 1 \quad (4.3)$$

Certaines méthodes d'extraction de caractéristiques, par exemple celles basées sur le codage de Freeman, requièrent que l'image soit préalablement binarisée. Malheureusement, la binarisation peut provoquer des pertes importantes d'information. Il est donc préférable de recourir à l'utilisation d'images en niveaux de gris. Dans ce cas, la formule d'accumulation (4.3) doit être remplacée par celle-ci :

$$h(\rho_k, \theta_k) \leftarrow h(\rho_k, \theta_k) + f(x_i, y_i) \quad (4.4)$$

A cause de la discrétisation, deux segments contenant le même nombre de points peuvent avoir des longueurs différentes (Fig. 4.3) [Lo95]. Dans ce cas, la formule d'accumulation (4.4) peut être remplacée par celle-ci :

$$h(\rho_k, \theta_k) = h(\rho_k, \theta_k) + f(x, y) \begin{cases} \frac{1}{|\sin \theta_k|} & \text{si } \frac{\pi}{4} \leq \theta_k \leq \frac{3\pi}{4} \\ \frac{1}{|\cos \theta_k|} & \text{sinon} \end{cases}$$

Un autre aspect important qui doit être pris en compte concerne la localisation de l'origine des coordonnées [Imme98]. En effet, la transformée de Hough est plus compacte lorsque l'origine est placée au centre de l'image. Par exemple, lorsque l'origine est placée dans un coin de l'image, ρ prend des valeurs dans l'intervalle $[-\frac{D}{\sqrt{2}}, D]$ (où D est la taille

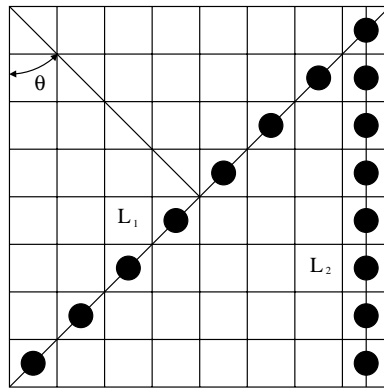


FIG. 4.3 – Problème de la longueur des segments. ($L_1 = 8\sqrt{2}$, $L_2 = 8$)

de la diagonale). Au contraire, lorsque l'origine est placée au centre de l'image, ρ prend des valeurs dans l'intervalle $[-\frac{D}{2}, \frac{D}{2}]$. Ceci représente une réduction de la taille de l'intervalle de ρ et permet une réduction de l'espace mémoire requis pour stocker l'accumulateur $h(\cdot)$. Il est donc possible, pour une taille constante de l'accumulateur d'augmenter la résolution de ρ lorsque l'origine est placée au centre de l'image.

4.1.2 Variabilité

Plusieurs études ont été réalisées afin d'appliquer la transformée de Hough à la reconnaissance de l'écriture, notamment à la reconnaissance de caractères cursifs. Cependant, son application reste limitée à certains types d'écriture composés principalement de tracés linéaires comme l'Hébreu et le Chinois [Cast96, Chan92, Mand92, Cheng89, Mats85].

L'écriture cursive est fondamentalement sujette à des déformations. Ces déformations sont à l'origine de variations complexes dans l'enveloppe de la transformée de Hough, elles sont très difficiles à éliminer. Outre le fait que l'écriture cursive est composée principalement de tracés courbes, les variations de l'épaisseur du tracé provoquent des variations dans la forme des pics de la transformée de Hough (Fig. 4.4). Ce qui se traduit donc par une détérioration de la précision des algorithmes de détection de pics.

Un autre inconvénient est que les caractéristiques capturées par la transformée de Hough sont essentiellement de nature globale. Ce qui veut dire que l'information locale (tracés courbes, tracés courts) est moins bien capturée que l'information globale et l'information

de connexité est perdue.

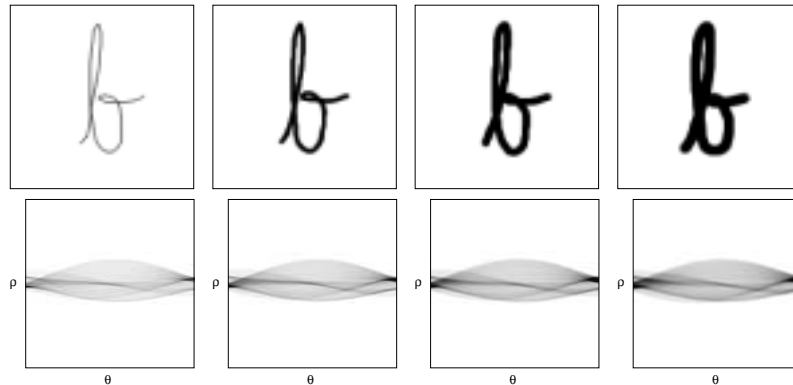


FIG. 4.4 – Variation de l'épaisseur. En haut, des images de test. En bas, les transformées de Hough respectives.

D'autre part, les translations se traduisent par des modulations de l'enveloppe de la transformée de Hough. Nous pouvons constater dans les figures 4.5a), b), e) et f) que la forme de l'enveloppe change : les zones concaves peuvent devenir convexes et les convexes peuvent devenir concaves et la valeur θ ne change pas (les pics se déplacent dans la direction ρ). Heureusement, ce type de variations peut être éliminé en partie grâce au centrage de l'image par rapport à son barycentre. Finalement, les rotations ne modifient pas la forme de l'enveloppe, mais se traduisent par des translations dans la direction θ (les figures 4.5a), c), e) et g) montrent cet effet). Nous pouvons voir dans les figures 4.5a), d), e) et h) que l'omission de certains tracés a peu d'effet sur la forme de l'enveloppe de la transformée de Hough (on a l'impression que le caractère a subi une légère rotation).

4.1.3 Préservation de l'information

Les moments géométriques de l'image peuvent être utilisés pour extraire des caractéristiques invariantes à différents types de transformations. Une propriété importante de cette technique est qu'elle capture l'information géométrique de l'image. Cette propriété peut donc nous permettre d'évaluer la capacité de préservation de l'information de la transformée de Hough.

Soit $f(x, y)$ une image, soit $h(\rho, \theta)$ sa transformée de Hough. Les moments géométriques

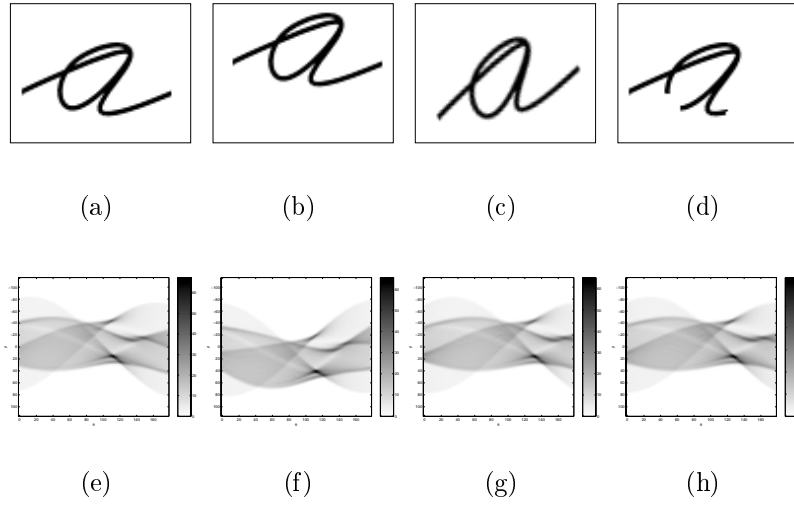


FIG. 4.5 – Effets dus à des transformations de l'image. En haut, des images de test. En bas, les transformées de Hough respectives. Les valeurs de l'accumulateur sont indiquées par des niveaux de gris (à droite de chaque image).

de la transformée de Hough $m_r(\theta)$ par rapport à ρ peuvent être exprimés par :

$$m_1(\theta) = \sum_{\rho} \rho h(\rho, \theta) = m_{10} \cos \theta + m_{01} \sin \theta$$

$$m_2(\theta) = \sum_{\rho} \rho^2 h(\rho, \theta) = \frac{1}{2}(m_{20} - m_{02}) \cos 2\theta + m_{11} \sin 2\theta + \frac{1}{2}(m_{20} + m_{02})$$

$$m_3(\theta) = \sum_{\rho} \rho^3 h(\rho, \theta) = \frac{1}{4}(m_{30} - m_{03}) \cos 3\theta + \frac{1}{4}(m_{21} - m_{03}) \sin 3\theta + \frac{3}{4}(m_{30} + m_{12}) \cos \theta + \frac{3}{4}(m_{03} + m_{21}) \sin \theta$$

avec

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y)$$

A partir de ces formules, nous pouvons obtenir facilement des expressions pour calculer les moments géométriques $m_{pq}(x, y)$ de l'image en fonction des moments géométriques

$m_j(\theta)$. Par exemple,

$$m_{10} = m_1(0)$$

$$m_{01} = m_1\left(\frac{\pi}{2}\right)$$

$$m_{20} = m_2(0)$$

$$m_{02} = m_2\left(\frac{\pi}{2}\right)$$

$$m_{11} = m_2\left(\frac{\pi}{4}\right) - m_2(0) - m_2\left(\frac{\pi}{2}\right)$$

De ce fait, nous pouvons affirmer que l'information géométrique capturée par les moments géométriques de l'image est aussi capturée par la transformée de Hough. D'ailleurs, nous pouvons vérifier facilement que le centrage de l'image par rapport à son centre de masse équivaut à centrer l'enveloppe de la transformée de Hough de telle sorte que $\sum_{\rho} \rho h(\rho, \theta) = 0$.

La transformée de Hough inverse permet de reconstituer l'image à partir de sa transformée de Hough (Fig. 4.6). Soit $f(x, y)$ une image, soit $h(\rho, \theta)$ sa transformée de Hough, soit N_{θ} le nombre d'orientations. La transformée de Hough inverse $\tilde{f}(x, y)$ peut être calculée par :

$$\tilde{f}(x, y) = \sum_{k=0}^{N_{\theta}-1} h(x \cos \theta_k + y \sin \theta_k, \theta_k)$$

Dans la Fig. 4.6 nous pouvons voir que pour une valeur relativement réduite de N_{θ} les principales caractéristiques géométriques du caractère sont reconstituées. D'ailleurs, les tracés droits sont généralement mieux reconstitués que les tracés courbes. Notons que la reconstitution n'est pas parfaite, même si on augmente la résolution de la transformée de Hough.

4.1.4 Applications

La transformée de Hough a de nombreuses applications dans des domaines très divers : imagerie médicale, analyse d'empreintes digitales, analyse de photos aériennes, suivi de cibles, analyse de minéraux, détection d'objets, reconnaissance de caractères, etc.

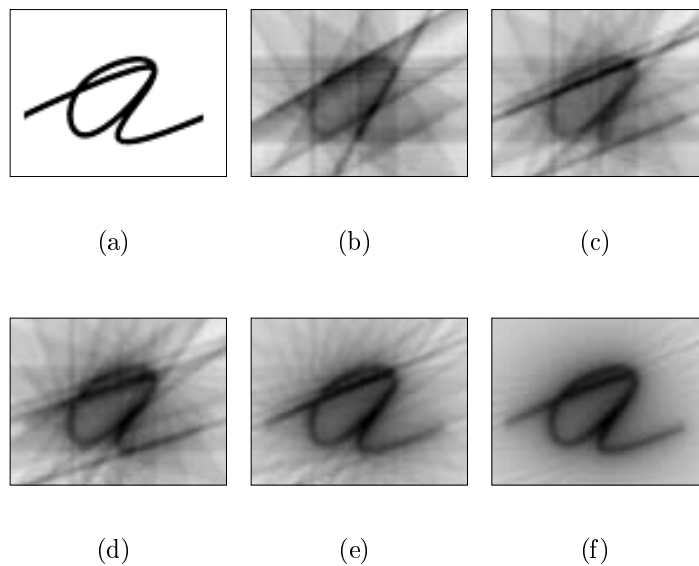


FIG. 4.6 – Reconstitution de l'image. a) Image de test, b) à e) sa reconstitution pour $N_\theta = 6, 8, 10, 16$ et 32 respectivement.

Dans le cadre de la reconnaissance de l'écriture, les principales applications sont :

1. la reconnaissance de caractères hébreux [Kush85]. Une caractéristique particulière de l'alphabet hébreu est que la plupart des caractères sont composés principalement de traits droits. Tout d'abord, chaque caractère est représenté par une image binaire de taille 32×32 . L'étape de pré-traitement comprend : squelettisation, et centrage par rapport au centre de masse. Dans l'étape suivante, des caractéristiques comprenant des extrémités de segments sont extraites de l'image du squelette. Ces caractéristiques sont alors utilisées pour effectuer une classification grossière (coarse classification) en 9 sous-classes. Deux étapes distinctes de reconnaissance sont donc utilisées selon chaque sous-classe : une étape utilisant des caractéristiques extraites de la TH et une étape utilisant des caractéristiques structurales. La TH de l'image du squelette est calculée en utilisant 45 niveaux de discrétisation pour la variable ρ et 18 pour la variable θ . Une opération de seuillage est alors effectuée afin de ne laisser que les cellules correspondant aux pics. La matrice résultante est utilisée directement comme vecteur de caractéristiques. Pour la classification finale, plusieurs vecteurs prototypes sont générés pendant la phase d'apprentissage.

2. la reconnaissance de caractères chinois [Cheng89]. De même que pour les caractères hébreux, les caractères chinois sont principalement composés de traits droits. Donc, chacun de ces tracés peut être détecté à l'aide de la TH. Tout d'abord, chaque caractère est représenté par une image binaire de taille 64×64 . L'étape de pré-traitement comprend : seuillage, squelettisation et normalisation. Dans l'étape suivante, l'image résultante est divisée en deux sous-images de taille 64×32 . Puis, une version modifiée de la TH est calculée pour chaque sous-image. Une opération de seuillage est alors utilisée afin de détecter des pics dans chaque sous-image. Chaque tracé détecté est ensuite codé par un vecteur comprenant sa longueur, sa position, et son orientation. Afin d'effectuer la classification finale, les vecteurs résultants sont utilisés pour reconnaître le caractère au moyen de la programmation dynamique. Le système a été testé sur une base de données contenant 881 classes de caractères (avec 160 exemples par classe). La performance estimée du système est de l'ordre de 94,5%.
3. la reconnaissance de caractères manuscrits [Mand92]. Dans cette application, l'ensemble des classes comprend des chiffres, des lettres majuscules et des lettres minuscules. L'étape de pré-traitement comprend : changement d'échelle et squelettisation. Chaque droite et courbe sont détectées par le seuillage des accumulateurs de la TH. Puis, la classification est effectuée en utilisant deux réseaux de neurones : l'un pour des lettres majuscules et l'autre pour des lettres minuscules.
4. la reconnaissance de chiffres manuscrits [Cast96]. Dans cette application, chaque caractère est représenté par une image binaire de taille 128×128 . Dans la première étape, la TH de l'image est calculée une première fois afin de calculer les paramètres de translation du caractère par l'approximation de son enveloppe convexe. Puis, ces paramètres sont utilisés pour calculer la TH normalisée de l'image qui est ensuite échantillonnée afin d'extraire un vecteur de caractéristiques. Finalement, la classification est effectuée en utilisant un réseau de Kohonen. La performance estimée du système est de l'ordre de 80%.

Par ailleurs, d'autres applications utilisent des techniques similaires. Le système de recon-

naissance de chiffres manuscrits décrit dans [Taka91] constitue un bon exemple. Dans ce cas, le système effectue une détection grossière de l'orientation du contour à l'aide de masques binaires (Fig. 4.7). Les cartes de caractéristiques ainsi obtenues sont ensuite échantillonnées en utilisant des tranches parallèles à plusieurs orientations. Comme chaque tranche peut être représentée par les coordonnées (ρ, θ) de sa droite centrale, cette opération équivaut donc à calculer la TH de chaque carte de caractéristiques. En conclusion, cette méthode effectue une estimation grossière de la direction du contour suivie d'une estimation plus robuste à l'aide de la transformée de Hough.

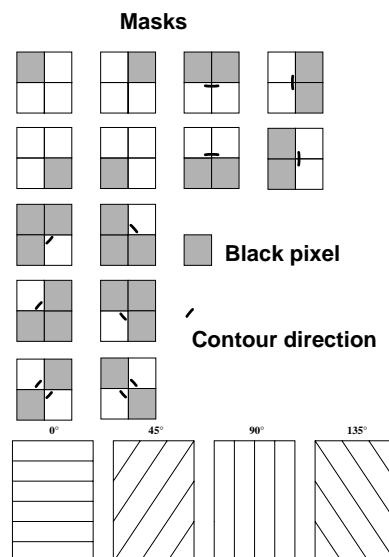


FIG. 4.7 – Exemple d'extraction de caractéristiques de contour [Taka91].

La détection de l'angle d'inclinaison de mots cursifs constitue une autre application potentielle de la transformée de Hough [Côté97]. Dans ce cas, on commence par calculer la TH $h(\rho, \theta)$ de l'image réduite aux contours (ou bords). Dans le cas d'une image en niveaux de gris, il est possible d'utiliser les gradients de l'image. Puis, pour chaque orientation θ_t on doit calculer l'entropie :

$$H(\theta_t) = - \sum_k h'(\rho_k, \theta_t) \ln h'(\rho_k, \theta_t)$$

avec

$$h'(\rho_k, \theta_t) = \frac{h(\rho_k, \theta_t)}{\sum_j h(\rho_j, \theta_t)}$$

Finalement, on calcule l'angle d'inclinaison α par :

$$\alpha = \arg \min_{\forall t} H(\theta_t) - \frac{\pi}{2}$$

4.1.5 La transformée de Radon

La transformée de Hough constitue une variante de la transformée de Radon [Dean81, Murp86, Illi88]. La transformée de Radon, pour la détection de caractéristiques linéaires, d'une image $f(x, y)$ peut être définie comme

$$R(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(\rho - x \cos \theta - y \sin \theta) dx dy$$

où δ représente la fonction delta de Dirac. Cette fonction force l'intégration de la fonction $f(x, y)$ le long d'une droite représentée par

$$\rho = x \cos \theta + y \sin \theta$$

Cette transformée donne les projections $R(\rho, \theta)$ de la fonction d'intensité $f(x, y)$ à différents angles d'inclinaison θ . La transformée de Hough classique est alors équivalente à la transformée de Radon discrète ($R(\rho, \theta) = h(\rho, \theta)$). Ces deux techniques impliquent donc une intégration de la fonction d'intensité $f(x, y)$ le long d'une droite. Pour cette raison, elles sont très appropriées pour des images bruitées. Cela est dû au fait que le bruit tend à être annulé par le processus d'intégration. Ce qui veut dire que les rapports signal/bruit de ces deux transformées sont supérieurs à celui de l'image elle-même.

La transformée de Radon peut être calculée rapidement à l'aide de la transformée de Fourier [Murp86]. Soit $F(u, v)$ la transformée de Fourier de la fonction d'intensité $f(x, y)$, soit $G_\theta(w)$ la transformée de Fourier de la fonction de projection $R(\rho, \theta)$. Par application

du théorème de projection de Fourier (slice theorem), la fonction $G_\theta(w)$ peut être exprimée par

$$G_\theta(w) = F(w \cos \theta, w \sin \theta) \quad (4.5)$$

Cela veut dire que la transformée de Radon peut être calculée par l'échantillonnage de la transformée de Fourier de l'image en coordonnées polaires, puis par le calcul de la transformée inverse de Fourier pour chaque angle θ . Ainsi, le calcul de la transformée de Radon procède comme suit :

1. Calculer la transformée de Fourier $F(u, v)$ de l'image $f(x, y)$.
2. Calculer chacune des fonctions $\{G_\theta(w) | \theta = \frac{\pi k}{N_\theta}; k = 0, 1, \dots, N_\theta - 1\}$ par la formule (4.5).
3. Calculer la transformée de Fourier inverse pour chacune des fonctions $\{G_\theta(w)\}$ afin d'obtenir les fonctions de projection $R(\rho, \theta)$ qui constituent la transformée de Radon.

L'efficacité de cet algorithme vient du fait que la transformation de l'image a lieu dans le domaine fréquentiel. C'est à dire que le calcul comprend essentiellement des appels d'une routine efficace comme la transformée rapide de Fourier (FFT).

La complexité de cet algorithme est de l'ordre de $O(N_\theta N \log N)$ contre $O(N_\theta N^2)$ qui correspond à la transformée de Hough. Il faut cependant insister que, pour des images non-bruitées de petite taille, le calcul de la transformée de Hough classique peut être effectué très rapidement en ne considérant que les pixels noirs. A titre d'exemple, le nombre de pixels noirs d'une image contenant un caractère cursif est généralement très inférieur au nombre de pixels blancs. Donc, le temps de calcul de la transformée de Hough classique peut être réduit au moins de moitié.

4.2 Méthodes développées

Comme nous avons vu dans les sections précédentes, la transformée de Hough constitue l'une des meilleures techniques de détection de droites. C'est d'ailleurs pour cette raison

que cette technique avait été appliquée à la reconnaissance de caractères principalement composés de traits droits. Dans le cas de caractères cursifs, la détection des tracés est plus difficile car l'image contient généralement des tracés droits et courbes. Comme la transformée de Hough d'une image contenant un caractère cursif ne présente pas de pics bien définis, les algorithmes de détection ne sont pas capables de détecter tous les tracés et peuvent même détecter des tracés inexistantes. Une solution à ce problème consiste donc à coder pour chaque position de l'image (ou bien pour chaque pixel noir) la présence d'une droite avec une orientation particulière. En vision, une représentation de ce type est connue sous le nom de "carte de caractéristiques". Une carte de caractéristiques est donc une matrice bidimensionnelle dont les positions représentent la présence d'une caractéristique particulière. Le système visuel est censé construire des représentations de ce type afin de coder la présence de certaines primitives à chaque position de l'image [Trei85]. Afin d'obtenir cette représentation, nous nous sommes inspiré de la méthode de calcul de la transformée de Hough que nous avons présentée dans la section précédente.

4.2.1 Extraction de caractéristiques directionnelles

Nous avons vu dans la section 4.1.5 que la transformée de Hough peut être calculée par l'échantillonnage du domaine fréquentiel à l'aide de l'équation 4.5. Nous pouvons vérifier facilement que cette équation correspond en réalité à la corrélation de l'image avec une droite centrée dans l'origine. D'ailleurs, la transformée de Fourier d'une droite d'angle θ est donnée par

$$T_{\theta}(u, v) = \delta(-u \sin \theta + v \cos \theta)$$

avec

$$\delta(t) = \begin{cases} 1 & \text{si } t = 0 \\ 0 & \text{sinon} \end{cases}$$

C'est à dire,

$$T_\theta(w) = 1$$

Donc, au lieu de calculer la transformée de Hough de l'image, nous pouvons extraire l'information directionnelle capturée par cette transformée en calculant plutôt la transformée inverse de Fourier de chacune des fonctions $\{G_\theta(w)\}$.

Soit $f(x, y)$ une image en niveaux de gris, soit $F(u, v)$ sa transformée de Fourier, soit $T_\theta(u, v)$ la transformée de Fourier d'une droite d'orientation θ . L'information directionnelle correspondant à une orientation θ peut être extraite par

$$m_\theta(x, y) = \left| \mathfrak{S}^{-1}\{F(u, v)T_\theta(u, v)\} \right| \quad (4.6)$$

avec

$$T_\theta(u, v) = \delta(-u \sin \theta + v \cos \theta)$$

et

$$\delta(t) = \begin{cases} 1 & \text{si } t = 0 \\ 0 & \text{sinon} \end{cases}$$

Les cartes de caractéristiques $\{m_\theta(x, y)\}$ ainsi obtenues représentent la direction du tracé dans chaque pixel de l'image $f(x, y)$. Donc, pour chaque pixel de l'image $f(x, y)$, $m_\theta(x, y)$ représente la contribution des pixels alignés dans la direction θ .

Donc, dans la première étape, nous calculons la FFT $F(u, v)$ de l'image $f(x, y)$. Puis, nous calculons le produit $F(u, v)T_\theta(u, v)$. Ensuite, nous calculons chaque carte de caractéristiques $m_\theta(x, y)$ par la FFT inverse.

La figure 4.8 montre quelques exemples des cartes de caractéristiques obtenues par cette méthode. Comme nous le verrons dans le chapitre suivant, les caractéristiques obtenues par cette méthode sont très similaires à celles qui peuvent être obtenues par l'échantillonnage de la transformée de Hough (voir Fig. 5.2). Cependant, les résultats ne sont pas identiques

parce que la FFT considère l'image comme périodique. Un avantage de cette méthode est qu'elle n'est pas affectée par des erreurs de quantification de la variable ρ . Les caractéristiques obtenues par cette méthode sont donc de nature globale et présentent une bonne résistance au bruit et aux déconnexions.

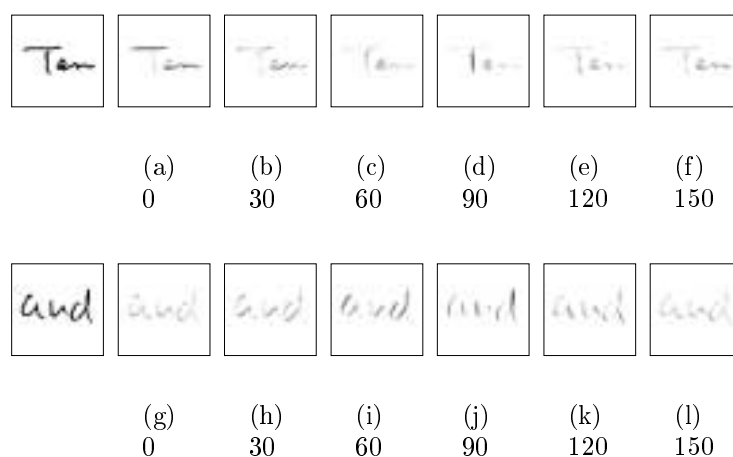


FIG. 4.8 – Cartes de caractéristiques directionnelles obtenues par notre méthode globale basée sur la FFT pour θ allant de 0 à 150°. A gauche, des images de test. A droite, les cartes de caractéristiques correspondantes.

4.2.2 Information locale

Certaines études ont montré que la transformée de Hough peut présenter des pics ne correspondant à aucune ligne dans l'image [Kris87]. Ces pics sont dus à des alignements de pixels qui ne forment réellement aucun segment de droite dans l'image (à cause de la perte de l'information de connexité). Ce problème est très commun dans des images de mots cursifs car la zone centrale contient généralement un grand nombre de pixels. D'ailleurs, les tracés cursifs ne sont pas généralement des tracés droits.

Ce problème peut être attribué à la nature globale de la transformée de Hough et au fait qu'elle ne prend pas en compte la connexité du tracé. Une solution à ce problème consiste donc à prendre en compte la connexité du tracé ou l'information locale comme dans certaines techniques telles que la transformée de Hough aléatoire (RHT) et la transformée de Hough combinatoire (CHT : Combinatorial Hough Transform) [Kalv95]. L'idée principale

de cette approche est de placer des fenêtres centrées sur chaque pixel noir afin de calculer la transformée de Hough de façon locale. Donc, nous avons adapté la méthode que nous avons présentée dans la section précédente afin d'extraire l'information directionnelle de façon locale. C'est à dire, au lieu d'utiliser des droites d'orientation variable, nous devons plutôt utiliser des segments de droite. D'ailleurs, la corrélation de l'image avec des segments de droite est équivalente à calculer la corrélation de l'image avec une droite dans une fenêtre (circulaire). La longueur du segment est alors égale au diamètre de la fenêtre utilisée.

Soit $f(x, y)$ une image en niveaux de gris, soit $F(u, v)$ sa transformée de Fourier, soit $T_{\lambda\theta}(u, v)$ la transformée de Fourier d'un segment de droite de longueur 2λ (diamètre de la fenêtre). L'information directionnelle correspondant à une orientation θ peut être calculée par

$$m_{\theta}(x, y) = \left| \mathfrak{S}^{-1}\{F(u, v)T_{\theta}(u, v)\} \right| \quad (4.7)$$

avec

$$T_{\theta}(u, v) = \frac{\sin(\pi\lambda(u \cos \theta + v \sin \theta))}{\pi\lambda(u \cos \theta + v \sin \theta)}$$

Dans la première étape, nous calculons la FFT $F(u, v)$ de l'image $f(x, y)$. Puis, nous calculons le produit $F(u, v)T_{\theta}(u, v)$. Ensuite, nous calculons chaque carte de caractéristiques $m_{\theta}(x, y)$ à l'aide de la FFT inverse. A la différence de la méthode globale, cette méthode est capable de capturer l'information directionnelle contenue dans la zone centrale du mot. L'efficacité de la méthode reste cependant dépendante de la valeur de λ . Usuellement, λ peut être déterminé en fonction de la hauteur de la ligne de base.

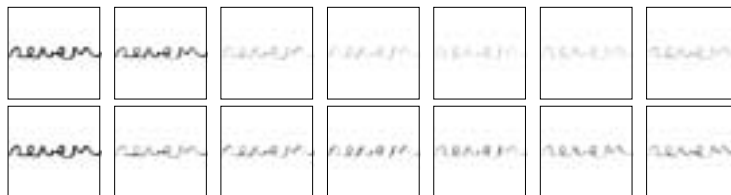


FIG. 4.9 – Utilisation de l'information locale. A gauche, des images de test. A droite, les cartes de caractéristiques correspondantes. En haut, la méthode basée sur la transformée de Hough. En bas, la méthode locale basée sur la FFT pour $\lambda = 18$.

La figure 4.9 montre quelques exemples des cartes de caractéristiques obtenues par cette méthode. La première ligne montre les cartes de caractéristiques obtenues par notre méthode globale. La deuxième ligne montre les cartes de caractéristiques obtenues par notre méthode locale. Nous pouvons voir que, à la différence de notre méthode globale, la méthode locale est capable de calculer les orientations des tracés contenus dans la zone centrale.

4.2.3 Autre méthode considérée

Dans la section précédente nous avons vu que l'extraction des caractéristiques directionnelles consiste à échantillonner le domaine fréquentiel à l'aide de secteurs symétriques d'angle $\Delta\theta \approx 0$ (Fig. 4.10). Cependant, nous avons remarqué que cette opération provoque des pertes d'information à cause du sous-échantillonnage. Nous avons donc tenté d'éviter ces pertes d'information en augmentant la valeur de l'angle $\Delta\theta$ des secteurs symétriques de façon à couvrir tout le spectre fréquentiel (Fig. 4.10). C'est à dire que pour un nombre d'orientations N_θ , nous avons

$$\Delta\theta = \frac{\pi}{N_\theta}$$

et

$$\theta_k = \frac{\pi k}{N_\theta}$$

Soit $f(x, y)$ une image contenant un caractère cursif, soit $F(u, v)$ sa transformée de Fourier, soit $G_\theta(u, v)$ un secteur symétrique orienté à un angle θ . Chaque carte de caractéristiques $m_k(x, y)$ correspondant à une orientation θ peut être calculée par :

$$m_k(x, y) = \left| \mathfrak{S}^{-1}\{F(u, v)G_{\theta_k}(u, v)\} \right| \quad (4.8)$$

La figure 4.11 montre quelques exemples des cartes de caractéristiques obtenues par cette formule. Nous pouvons voir que cette méthode est aussi capable de capturer l'information directionnelle des tracés contenus dans la zone centrale. Un avantage de cette méthode par

rapport à la précédente est qu'elle ne dépend pas du paramètre λ .

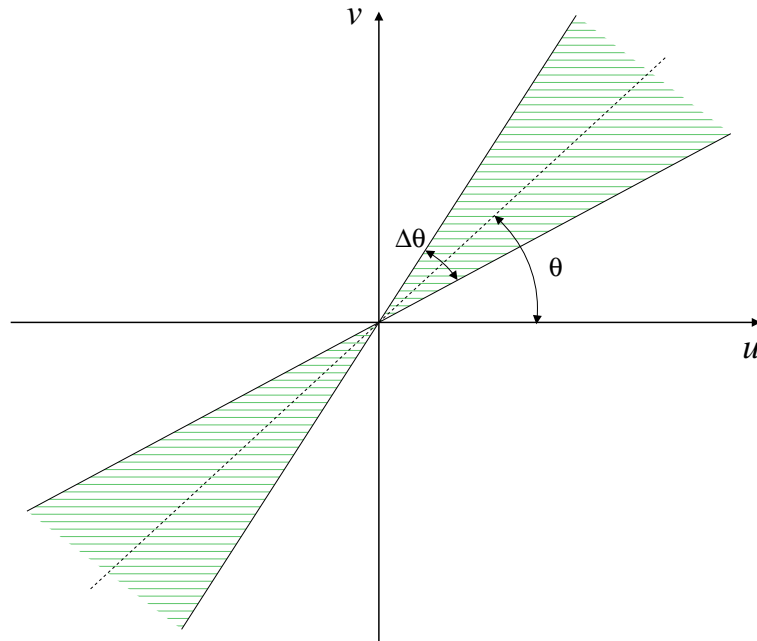


FIG. 4.10 – Le noyau $G_\theta(u, v)$. ($\Delta\theta = \frac{\pi}{N_\theta}$, N_θ = nombre d'orientations)



FIG. 4.11 – Extraction de caractéristiques directionnelles par l'utilisation de secteurs symétriques. A gauche, des images de test. A droite, les cartes de caractéristiques correspondantes.

4.3 Caractéristiques de courbure

Certaines études ont montré que le système visuel humain possède des cellules répondant sélectivement à l'information courbure de la scène visuelle [Whit98]. Cette information serait donc utile pour la reconnaissance de visages ou d'objets comportant des formes courbes.

La transformée de Hough est une technique de détection de droites très efficace qui permet d'extraire des caractéristiques directionnelles présentant une bonne tolérance aux

déconnexions et au bruit. Cette technique peut être utilisée également pour la détection d'autres formes paramétriques (cercles, paraboles, ellipses, arcs, rectangles, etc). Nous avons donc considéré cette technique pour l'extraction de caractéristiques de courbure dans des images en niveaux de gris. Comme le temps de calcul est usuellement très important, nous avons opté pour une implémentation à l'aide de la transformée rapide de Fourier.

Soit $f(x, y)$ une image en niveaux de gris, soit $F(u, v)$ sa transformée de Fourier, soit $H_r(x, y)$ la transformée de Fourier d'un cercle de rayon r . Les cartes de caractéristiques de courbure $c_r(x, y)$ correspondant au rayon de courbure r sont calculées par :

$$c_r(x, y) = \left| \mathfrak{F}^{-1}\{F(u, v)H_r(u, v)\} \right| \quad (4.9)$$

La figure 4.12 montre quelques exemples des cartes de caractéristiques obtenues. Nous pouvons voir que cette méthode produit des pics aux centres de boucles et de tracés courbes.

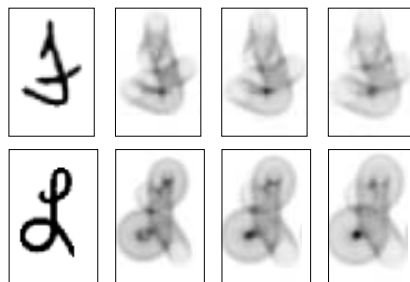


FIG. 4.12 – Extraction de caractéristiques de courbure. A gauche, des images de test. A droite, les cartes de caractéristiques correspondantes.

4.4 Généralisation

A cause de la nature de l'écriture cursive, les caractéristiques à utiliser doivent être résistantes aux déformations, au bruit et aux variations de style. D'ailleurs, les caractéristiques topologiques sont très résistantes aux déformations [Davi86, Gail79, Govi90, Mant86]. Par exemple, le nombre de trous est invariant aux changements d'échelle et aux rotations. Cependant, l'extraction de ce type de caractéristiques s'avère complexe. D'autre part, certaines études ont montré que le système visuel est capable d'extraire ce type d'information

de la scène visuelle. Par exemple, il a été montré que la reconnaissance humaine d'objets est une fonction des propriétés topologiques de la scène visuelle [Hetc98]. C'est à dire que, plus les objets diffèrent dans leurs propriétés topologiques, plus ils sont faciles à reconnaître.

La transformée de Hough constitue en réalité une implémentation efficace de la méthode de template matching [Illi88]. Par ailleurs, la corrélation de l'image avec des templates peut être utilisée pour détecter des droites. Cependant, même si la valeur des pics de la transformée de Hough donne une indication sur la longueur des segments de droite, en revanche elle ne permet pas de calculer la position d'un pixel donné par rapport au segment de droite dont il fait partie. Une solution à ce problème consiste donc à utiliser une approche similaire à celle utilisée par certaines méthodes d'intersection avec des droites. Par exemple, au lieu d'utiliser une seule droite par direction, nous pouvons utiliser deux demi-droites de directions opposées. Dans ce cadre, notre méthode d'extraction de caractéristiques consiste donc à calculer la corrélation de l'image avec des demi-droites à plusieurs orientations $\theta \in [0, 2\pi]$ et positions (x, y) . D'ailleurs, cette opération peut être effectuée à l'aide de techniques efficaces comme la FFT.

Soit $f(x, y)$ une image en niveaux de gris, soit $F(u, v)$ sa transformée de Fourier, soit $H_\theta(x, y)$ la transformée de Fourier d'un segment de droite ℓ_θ , soit $2N_\theta$ le nombre d'orientations. Chaque carte de caractéristiques $b_k(x, y)$ correspondant à une orientation $\theta_k = \frac{\pi k}{N_\theta}$ peut être calculée par :

$$b_k(x, y) = \left| \mathfrak{S}^{-1}\{F(u, v)H_{\theta_k}(u, v)\} \right| \quad (4.10)$$

avec

$$H_{\theta_k}(u, v) = \frac{\sin(\pi\lambda(u \cos \theta_k + v \sin \theta_k))}{\pi\lambda(u \cos \theta_k + v \sin \theta_k)} e^{-j\pi\lambda(u \cos \theta_k + v \sin \theta_k)}$$

Comme nous le verrons ensuite, les caractéristiques $b_k(x, y)$ sont plus élémentaires que celles qui peuvent être obtenues à l'aide de la transformée de Hough. Elles présentent aussi une bonne tolérance au bruit et aux déconnexions du tracé, peuvent être utilisées pour

calculer des caractéristiques plus complexes et sont donc très informantes.

Les boucles peuvent être détectées par la détection des cavités centrales (régions de points blancs entourés de tous cotés par le tracé). C'est à dire que pour un point blanc de coordonnées (x, y) appartenant à une cavité centrale, nous avons que $\forall k \in \{0, 1, \dots, 2N_\theta - 1\} b_k(x, y) = 0$. Donc, afin de détecter les boucles de l'image, nous calculons une carte de caractéristiques $h(x, y)$ par :

$$h(x, y) = (1 - f(x, y)) \prod_{k=0}^{2N_\theta-1} g(b_k(x, y))$$

avec

$$g(z) = \begin{cases} 1 & \text{si } z \geq \frac{1}{\lambda} \\ \lambda z & \text{sinon} \end{cases}$$

Nous pouvons vérifier facilement que $h(x, y)$ est nul pour des pixels noirs ($f(x, y) = 1$). La figure 4.12 montre quelques exemples des cartes de caractéristiques obtenues. Un avantage de cette méthode par rapport à d'autres méthodes de détection de boucles est qu'elle présente une bonne résistance aux déconnexions du tracé.



FIG. 4.13 – Détection de boucles et de cavités centrales.

Par une méthode similaire, nous pouvons également détecter différents types de cavités. Par exemple, les cavités Est peuvent être détectées en utilisant la formule :

$$(1 - f(x, y))(1 - g(b_{N_\theta}(x, y))) \prod_{k \neq N_\theta} g(b_k(x, y))$$

Les points caractéristiques constituent une description complète de la forme contenue dans l'image [Attn54]. Afin de capturer une partie de l'information contenue dans les points

caractéristiques, nous pouvons calculer des cartes de caractéristiques $s_k(x, y)$ par :

$$s_k(x, y) = f(x, y)b_k(x, y)(1 - b_{k+N_\theta}(x, y))$$

La figure 4.14 montre des exemples des caractéristiques obtenues par cette méthode. Nous pouvons voir que cette méthode est capable de détecter des points de changement de direction et des extrémités de segments.

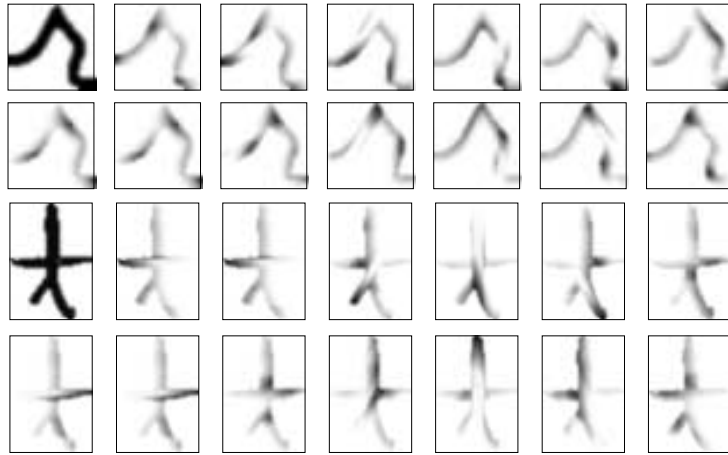


FIG. 4.14 – Des caractéristiques capturant une partie de l'information contenue dans les points caractéristiques.

Maintenant, afin de calculer des caractéristiques directionnelles (orientation du tracé), nous utilisons la formule suivante :

$$m_k(x, y) = b_k(x, y) + b_{k+N_\theta}(x, y)$$

Les cartes de caractéristiques $b_k(x, y)$ peuvent également être utilisées pour estimer l'épaisseur moyenne du tracé. Pour cela nous calculons :

$$b'_k(x, y) = (1 - f(x, y))b_k(x, y)$$

puis nous éliminons les pixels satisfaisant à la condition :

$$b'_k(x, y) < \frac{1}{\lambda}$$

Ensuite, nous calculons l'épaisseur moyenne du tracé en déterminant la position du plus grand pic de l'histogramme d'intensité des cartes $b'_k(x, y)$. La figure 4.15 montre des exemples de l'utilisation de cette méthode.

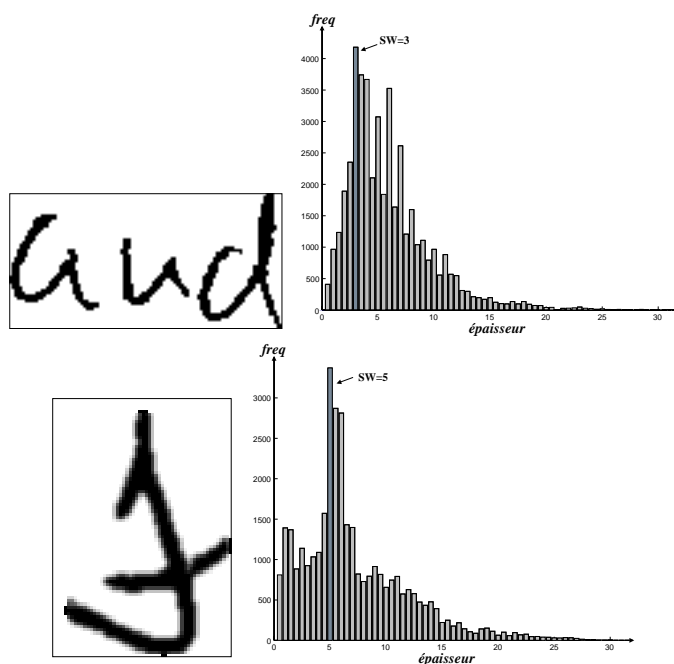


FIG. 4.15 – Estimation de l'épaisseur moyenne du tracé. À gauche, des images de test. À droite, les histogrammes correspondants (sans lissage).

4.5 Conclusions

Nous avons présenté une méthode globale et une méthode locale pour l'extraction de caractéristiques directionnelles présentant une bonne robustesse au bruit et aux déconnexions du tracé. Également, nous avons présenté une méthode d'extraction de caractéristiques de courbure. Enfin, nous avons présenté une méthode générale pour l'extraction de différents types de caractéristiques présentant une bonne résistance au bruit et aux déconnexions.

Dans la première étape de notre méthode, nous obtenons des caractéristiques élémentaires que nous utilisons pour le calcul de caractéristiques plus complexes. Par exemple, une caractéristique directionnelle est calculée comme la somme de deux caractéristiques élémentaires. L'extraction de caractéristiques plus complexes et plus résistantes aux déformations (boucles, cavités, extrémités de segments, points d'inflexion) doit comporter des fonctions non-linéaires. D'ailleurs, les caractéristiques élémentaires semblent être biologiquement plausibles parce qu'elles peuvent être obtenues en utilisant une carte bidimensionnelle de neurones ayant des connexions latérales d'orientation variable.

Des recherches ultérieures porteraient éventuellement sur l'évaluation des caractéristiques obtenues. De même, il serait intéressant de réaliser l'adaptation d'un réseau de neurones de type LeNet pour l'utilisation de ces caractéristiques et de vérifier sa performance.

Chapitre 5

Reconnaissance de caractères cursifs

5.1 Introduction

Comme nous avons indiqué dans le chapitre 1, dans une image, l'information pertinente est principalement contenue dans les contours et les bords. C'est pour cette raison, que certains systèmes utilisent des méthodes de détection de bords et de suivi de contours. Des caractéristiques directionnelles ont été utilisées avec succès pour la reconnaissance de chiffres manuscrits et pour la reconnaissance de caractères cursifs segmentés [Srik96, Taka91, Kimu97]. Ces caractéristiques sont, en général, extraites des contours en utilisant différentes techniques (codage de Freeman, opérateurs de différentiation, filtres directionnels, filtres de Gabor). Cependant, dans le cas du cursif, les contours et les bords sont usuellement sujets à des déformations et au bruit qui peuvent affecter la performance de ces méthodes. Au lieu de mesurer la direction du contour ou la direction du gradient de l'image, on peut plutôt mesurer l'orientation du tracé qui constitue une caractéristique plus stable. Pour cette tâche, il existe des techniques très efficaces de détection de droites telles que la transformée de Hough. Comme nous l'avons vu dans le chapitre précédent, cette technique présente la particularité d'être très résistante aux déconnexions du tracé et au bruit. Nous avons donc développé une méthode de reconnaissance de caractères cursifs basée sur cette technique.

5.2 Notre système

Notre système (Fig. 5.1) reçoit en entrée une image binaire ou en niveaux de gris de taille variable contenant un caractère cursif. Dans la première étape, nous normalisons cette image par rapport à la ligne de base afin d'obtenir une image en niveaux de taille fixe (32×48). Dans l'étape suivante, nous calculons la transformée de Hough de cette image. Puis, nous calculons par balayage de l'espace de Hough un ensemble de cartes de caractéristiques directionnelles. Ensuite, nous appliquons une méthode d'extraction de caractéristiques locales sur chaque carte. Finalement, nous utilisons les caractéristiques obtenues pour reconnaître l'image du caractère cursif à l'aide d'un réseau de neurones.

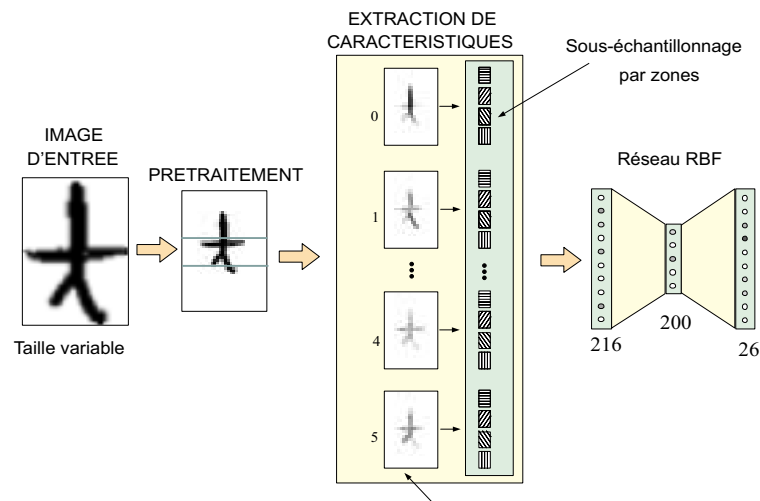


FIG. 5.1 – Architecture du système.

5.3 Extraction de caractéristiques

Dans certains systèmes à base de réseaux de neurones, les pixels sont utilisés directement pour la classification de l'image. Cette méthode pose certains problèmes car même pour des images de petite taille (32×32), la dimensionnalité de l'espace de caractéristiques est très élevée (1024). Or, la performance des systèmes est en général influencée par la dimensionnalité de l'espace de caractéristiques (curse of dimensionality). Pour cette raison,

il est usuel que ces systèmes utilisent des images de petite taille (16×16). L'extraction de caractéristiques vise donc à éviter ce problème par l'extraction d'informations pertinentes permettant de distinguer plus facilement les différentes classes.

Chaque caractère cursif peut être représenté soit par des segments de droite et des courbes ayant des orientations et positions relatives données, soit par des points caractéristiques de l'image (extrémités, extremums de courbure), soit par des ascendants, des descendants, des boucles et des concavités. Nous pouvons donc reconnaître chaque caractère à partir de ces informations telles que l'orientation et la position relative de chaque tracé. D'ailleurs, l'orientation de chaque tracé peut être calculée à l'aide de techniques robustes de détection de droites telles que la transformée de Hough.

Dans la première étape de notre méthode d'extraction de caractéristiques, nous utilisons la transformée de Hough pour capturer l'information directionnelle du tracé. La détection de droites à l'aide de la transformée de Hough implique généralement le seuillage des accumulateurs de la transformée. Cette technique est effective lorsque l'image contient principalement des tracés quasi-linéaires. Afin de conserver toute l'information capturée par la transformée de Hough, nous devons plutôt représenter l'orientation du tracé de façon implicite (sous forme de cartes de caractéristiques). Cette représentation peut être ensuite utilisée pour l'extraction de caractéristiques locales (caractéristiques statistiques).

Soit $f(x, y)$ une image en niveaux de gris, soit $h(\rho, \theta)$ sa transformée de Hough, soient N_ρ et N_θ les nombres de niveaux de discrétisation des variables ρ et θ respectivement. Nous calculons une carte de caractéristiques $m_k(x, y)$ pour chaque orientation θ_k par :

$$m_k(x, y) = \frac{h(x \cos \theta_k + y \sin \theta_k, \theta_k)}{\sum_{\theta} h(x \cos \theta + y \sin \theta, \theta)} f(x, y) \quad (5.1)$$

avec $\theta_k \in \{\frac{\pi k}{N_\theta} | k = 0, 1, \dots, N_\theta - 1\}$.

Ces cartes de caractéristiques conservent toute l'information contenue dans l'image $f(x, y)$. Donc, nous pouvons vérifier facilement que $f(x, y) = \sum_k m_k(x, y)$.

Comme chaque carte de caractéristiques est riche en information, nous appliquons sur

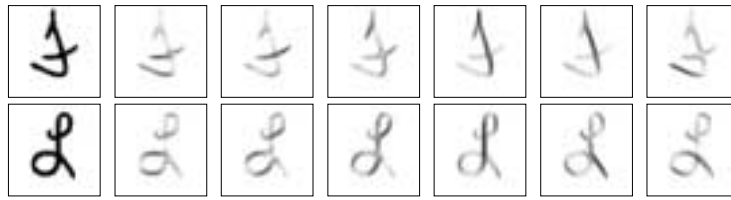


FIG. 5.2 – Extraction de caractéristiques directionnelles à l’aide de la transformée de Hough. A gauche, des images de test. A droite, les cartes de caractéristiques correspondantes.

chaque carte une méthode d’extraction de caractéristiques locales. Les caractéristiques ainsi obtenues représentent l’orientation du tracé dans une zone donnée de l’image. Pour cela, nous divisons chaque carte de caractéristiques en un ensemble de tranches parallèles $\{R_{ij} | i = 0, 1, \dots, N_s - 1; j = 0, 1, \dots, N_d - 1\}$ d’orientation variable (Fig. 5.3).

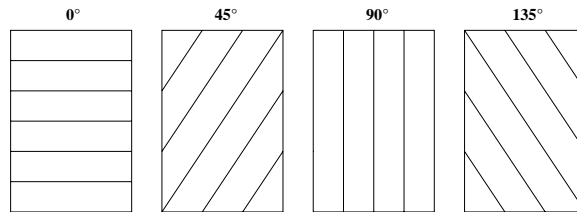


FIG. 5.3 – Division de l’image en zones [Taka91].

Puis, pour chaque zone R_{ij} , nous calculons des caractéristiques locales par :

$$F_{ijk} = \frac{\sum_{x, y \in R_{i,j}} m_k(x, y)}{\sum_{p=0}^{N_\theta-1} \sum_{x, y \in R_{i,j}} m_p(x, y)} \quad (5.2)$$

Cette méthode d’échantillonnage présente l’avantage que, dans une zone donnée, les caractéristiques extraites sont invariantes à des translations. La figure 5.4 montre un exemple des caractéristiques obtenues.



FIG. 5.4 – Caractéristiques obtenues après échantillonnage. A gauche, une image de test. A droite, les caractéristiques correspondantes.

5.4 Reconnaissance

Plusieurs études ont démontré l'efficacité des réseaux de neurones pour des tâches de classification. Parmi leurs applications nous pouvons citer la reconnaissance de caractères manuscrits. Des réseaux de neurones de type RBF ont été appliqués avec succès dans certaines études [Lema93, Park91]. L'avantage de ce type de réseaux de neurones par rapport à d'autres réseaux de neurones est qu'il existe de bonnes méthodes d'initialisation. Pour cette raison, le temps d'apprentissage peut être réduit considérablement. D'ailleurs, ce type de réseau présente une bonne performance et une bonne capacité de rejet [Lamy95]. Un autre avantage des réseaux de neurones est qu'ils sont capables d'approcher des probabilités a posteriori [Ney95].

Dans cette partie, nous utilisons un réseau de neurones RBF. Ce réseau reçoit en entrée chaque vecteur de caractéristiques obtenu dans l'étape précédente et donne en sortie une liste de scores de classification.

Le réseau de neurones comprend trois couches : couche d'entrée, couche cachée et couche de sortie. La couche d'entrée est totalement connectée à la couche cachée. La couche cachée comprend plusieurs unités de type Gaussien. La couche cachée est totalement connectée à la couche de sortie. La couche de sortie comprend plusieurs unités de type soft-max. Le nombre d'unités d'entrée est égal à la dimension des vecteurs de caractéristiques. Le nombre d'unités de sortie est égal au nombre de classes (26).

Le réseau est optimisé afin d'approcher les probabilités a posteriori $P(class = i | F)$ en minimisant l'erreur d'entropie croisée :

$$J = - \sum_{i=1}^N \sum_{j=1}^{26} \ln(1 - \|t_{ij} - o_{ij}\|)$$

où t_{ij} représente la valeur désirée et o_{ij} représente la valeur de sortie de l'unité j .

Une propriété notable de cette fonction est qu'elle permet de maximiser l'entropie em-

pirique conditionnelle [Ney95] :

$$H_{emp}(C|F) = \frac{1}{N} \sum_{i=1}^N \ln P(class = i|F_i)$$

où F_i représente le i -ème vecteur de caractéristiques et N représente le nombre de vecteurs.

Cela revient donc à minimiser l'information mutuelle empirique :

$$I_{emp}(C; F) = H_{emp}(C) - H_{emp}(C|F)$$

Ce qui veut dire que nous pouvons tester en même temps la performance et le pouvoir discriminant des caractéristiques utilisées. D'ailleurs, nous pouvons exploiter cette propriété pour effectuer une étape de sélection de caractéristiques. Dans ce cas, l'estimation est relativement rapide car la dimension du vecteur d'entrée est égale à 1.

Dans la première étape de l'apprentissage, nous initialisons les centres de chaque unité cachée par la méthode LVQ1. Puis, nous fixons les variances des unités cachées et nous calculons les poids des unités de sortie par la méthode de Widrow-Hopf.

Après initialisation, nous optimisons le réseau par la méthode de rétro-propagation du gradient.

5.5 Résultats

5.5.1 La base de données

Nous avons testé ce système sur une base de données contenant 2114 images binaires de caractères cursifs segmentés. Nous avons obtenu cette base de données par la segmentation semi-automatique d'une autre base de données contenant 416 images binaires de noms de villes (sous-ensemble de la base 'bincities', USPS). La figure 5.5 montre que les caractères contiennent différents types de déformations. Le tableau 5.1 montre la distribution des

classes dans la base de données. Nous pouvons voir que certaines classes sont très peu nombreuses ('b', 'f', 'p', 'x', 'z').

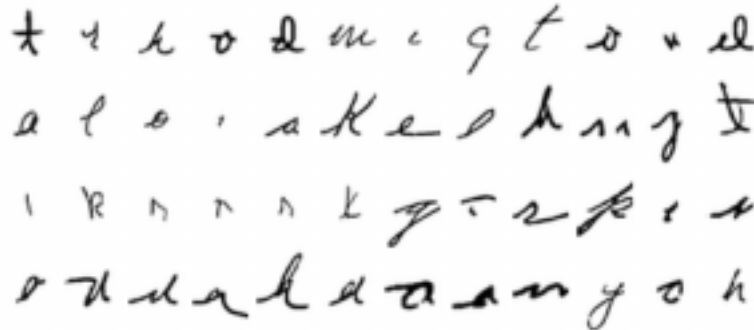


FIG. 5.5 – Quelques exemples de la base de données.

classe	#	classe	#
<i>a</i>	254	<i>n</i>	213
<i>b</i>	18	<i>o</i>	196
<i>c</i>	44	<i>p</i>	16
<i>d</i>	62	<i>r</i>	163
<i>e</i>	220	<i>s</i>	92
<i>f</i>	13	<i>t</i>	128
<i>g</i>	56	<i>u</i>	45
<i>h</i>	39	<i>v</i>	29
<i>i</i>	177	<i>w</i>	36
<i>k</i>	45	<i>x</i>	2
<i>l</i>	177	<i>y</i>	33
<i>m</i>	48	<i>z</i>	8

TAB. 5.1 – Distribution des classes dans la base de données.

5.5.2 Conditions de test

Étant donné que la taille de notre base des données est relativement petite et afin d'obtenir une estimation non-biaisée de la performance du classifieur, nous avons utilisé une technique de validation croisée. D'autre part, comme le temps requis par cette technique peut être très important, nous avons utilisé $k = 10$. Ainsi, nous avons obtenu dix ensembles de tests et dix ensembles d'apprentissage.

Puis, pour le calcul de la performance du système, nous avons utilisé $N_s = 6$, $N_d = 6$,

$N_\theta = 6$ et $N_\rho = 48$. Après avoir éliminé les composantes nulles, la dimension de chaque vecteur de caractéristiques était de 210.

Afin de déterminer la complexité du réseau de neurones pour laquelle la performance du système est la meilleure, nous avons fait varier le nombre d'unités cachées. Nous avons donc conclu que, pour obtenir la meilleure performance, le nombre d'unités cachées devait être au moins de 100. Dans nos expériences, le réseau RBF a donné de meilleurs résultats que d'autres classifieurs tels que k -plus proches voisins (43,89%) et des Perceptrons multi-couche. D'ailleurs, nous avons vu que la performance d'un réseau à poids partagés de type LeNET est de l'ordre de 42,65%. Cela est dû probablement au nombre insuffisant de données d'apprentissage disponibles.

5.5.3 Evaluation des caractéristiques

Comme le calcul de l'information mutuelle requiert un temps de calcul qui peut être très important, nous avons utilisé une méthode de partitionnement en régions (k -moyennes). Cette mesure de l'information mutuelle peut être utilisée pour maximiser le pouvoir discriminant des caractéristiques sur l'ensemble d'apprentissage. Cependant, nous ne l'avons utilisée que pour évaluer l'ensemble de caractéristiques. D'ailleurs, une estimation plus précise requiert l'utilisation de techniques dont le temps de calcul est en général très important (des mélanges de Gaussiennes, des réseaux de neurones). Ce qui entraîne également l'utilisation de techniques de validation. Par exemple, il est possible d'utiliser un réseau de neurones pour estimer l'information mutuelle empirique et une technique de validation croisée pour évaluer sa fiabilité.

Afin d'évaluer notre méthode d'extraction de caractéristiques par rapport à des méthodes opérant sur les contours, nous avons produit un autre ensemble contenant 210 caractéristiques directionnelles par l'utilisation d'une méthode basée sur les gradients de l'image [Srik96]. L'information mutuelle des caractéristiques résultantes était $H(C; F) = 1,78$. Nous avons optimisé un réseau de neurones en utilisant ces caractéristiques et sa performance était de 51,64%. Le temps et le nombre d'opérations requis pour calculer chaque

vecteur de caractéristiques étaient $0,49s$ et $8,3 \times 10^4$ respectivement.

L'information mutuelle des caractéristiques obtenues par notre méthode était $H(C; F) = 1,89$. Le temps et le nombre d'opérations requis pour calculer chaque vecteur de caractéristiques étaient $0,89s$ et $1,93 \times 10^5$ respectivement sous Matlab dans un poste de travail SUN. Donc, comparée avec la méthode basée sur les gradients de l'image, notre méthode est seulement 1,8 fois plus lente mais présente une meilleure capacité de généralisation.

Le tableau 5.2 montre l'information mutuelle en fonction de la taille du vecteur de caractéristiques. Nous pouvons voir que l'information mutuelle s'améliore lorsque nous augmentons le nombre de niveaux de discrétisation N_ρ . Au contraire, nous pouvons voir qu'augmenter la dimension du vecteur de caractéristiques ne se traduit pas généralement par une amélioration de performance.

N_ρ	$4 \times 4 \times 4$	$4 \times 4 \times 6$	$4 \times 4 \times 8$	$4 \times 6 \times 6$	$4 \times 6 \times 8$	$6 \times 6 \times 6$
32	1.909	1.925	1.919	1.896	1.895	1.862
48	1.924	1.948	1.957	1.918	1.921	1.892

TAB. 5.2 – L'information mutuelle $H(C; F)$ en fonction de la taille du vecteur de caractéristiques ($N_s \times N_d \times N_\theta$). (Le nombre de partitions est de 200.)

5.5.4 Résultats obtenus

Le tableau 5.3 montre un résumé des résultats obtenus pour la reconnaissance de caractères cursifs segmentés. La première ligne montre le taux de reconnaissance de notre système (vecteurs de caractéristiques de taille 210). La performance obtenue par validation croisée était de 60,71% avec une variance de 3,72%. La meilleure performance obtenue dans les tests était de 66,35%. La troisième ligne montre le taux de reconnaissance d'un autre système basé sur des caractéristiques structurales, mais testé sur une grande base de données (9166 échantillons d'apprentissage et 615 échantillons de test) [Srih93]. Donc, au moins dans nos expériences, notre système présente une bonne performance.

Le tableau 5.4 montre les résultats obtenus par validation croisée ($k = 10$) pour chaque classe. Les classes 'a', 'l', 'n', et 'o' présentent les meilleurs scores de classification et aussi

Méthode	top 1	top 2	top 3	top 5	top 6
Notre méthode	66.35	79.62	85.31	93.84	95.26
Gradients	51.64	69.95	79.81	85.45	88.73
Structurelle [Srih93]	63	73	78	-	-

TAB. 5.3 – Taux de reconnaissance.

classe	moyenne	variance	classe	moyenne	variance
<i>a</i>	65,09	9,06	<i>n</i>	71,19	8,03
<i>b</i>	20,00	42,16	<i>o</i>	66,02	5,89
<i>c</i>	24,14	18,43	<i>p</i>	58,33	46,65
<i>d</i>	60,91	17,96	<i>r</i>	53,90	17,36
<i>e</i>	65,56	15,91	<i>s</i>	35,07	19,19
<i>f</i>	68,33	36,39	<i>t</i>	76,40	14,23
<i>g</i>	82,07	15,62	<i>u</i>	41,89	31,30
<i>h</i>	40,33	36,32	<i>v</i>	7,50	16,87
<i>i</i>	57,06	10,97	<i>w</i>	51,00	32,31
<i>k</i>	33,79	27,38	<i>x</i>	80,00	42,16
<i>l</i>	84,49	8,75	<i>y</i>	50,67	31,84
<i>m</i>	40,17	19,29	<i>z</i>	50,00	52,70

TAB. 5.4 – Résultats obtenus par la validation croisée ($k = 10$).

les plus petites variances. D'ailleurs, les classes présentant les plus grandes variances correspondent aux classes dont les nombres d'échantillons sont les plus petits. De plus, parmi les classes dont le nombre d'échantillons est petit, celles possédant des extensions ('f', 'g') sont mieux reconnues que les autres. Le tableau 5.5 montre la matrice de confusion pour 72 unités cachées de type gaussien. Nous pouvons voir que la lettre 'e' est parfois reconnue comme 'a' mais pas à l'inverse. De même, la lettre 'i' est plus souvent reconnue comme 'e' que l'inverse.

La figure 5.6 montre la courbe de rejet. Nous pouvons voir que, pour obtenir une performance de 80%, le système doit rejeter presque 65% des échantillons. D'autre part, pour un rejet de 50%, la performance est de l'ordre de 75%.

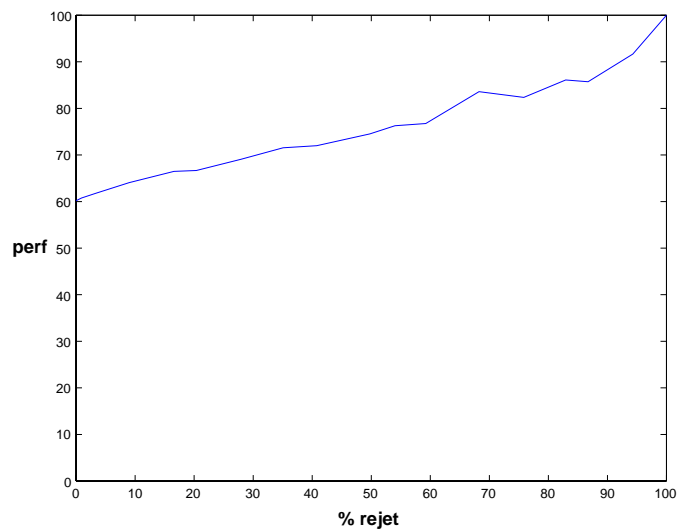


FIG. 5.6 – Courbe de rejet pour la tâche de reconnaissance de caractères cursifs segmentés.

5.6 Conclusions

Nous avons présenté un système pour la reconnaissance de caractères cursifs utilisant la transformée de Hough et un réseau de neurones.

La transformée de Hough est une technique robuste de détection de droites dans des images binaires. Cette transformée est capable de capturer l'information directionnelle de l'image et permet d'extraire des caractéristiques directionnelles présentant une bonne tolérance au bruit et aux déconnexions. Nous avons donc utilisé cette technique pour extraire l'information directionnelle de l'image. Puis nous avons appliqué une méthode d'extraction de caractéristiques locales. Le but de notre méthode est d'inférer l'orientation du tracé dans chaque zone de l'image. Nos résultats ont montré que les caractéristiques obtenues sont moins vulnérables aux déformations que celles qui peuvent être extraites des contours.

Nous avons comparé les caractéristiques obtenues par notre méthode avec celles obtenues par la méthode basée sur les directions du gradient. Bien que le temps requis pour calculer chaque vecteur de caractéristiques par notre méthode soit 1,8 fois plus grand, la performance de notre méthode était meilleure.

Nous avons observé qu'une partie des cas de mauvaise reconnaissance avec notre système était due à des caractères ayant subi des translations (à cause de mots inclinés). D'ailleurs,

comme nous considérons que la position verticale constitue une donnée importante, nous conservons la position du caractère par rapport à la ligne de base du mot cursif. Afin de résoudre ce problème, nous envisageons deux possibilités : l'utilisation d'une option de rejet et le centrage du caractère avec codification de sa position verticale originale. Dans le premier cas, lorsque la distance aux classes est trop grande, le caractère doit être rejeté et remplacé par une version centrée du même caractère. Dans le dernier cas, la position verticale originale peut être utilisée soit comme caractéristique additionnelle, soit pour résoudre éventuellement des problèmes d'ambiguïté dans la classification résultante.

Nous avons aussi envisagé plusieurs améliorations de notre méthode afin d'améliorer la performance de notre système. Par exemple, nous pouvons utiliser la méthode d'extraction de caractéristiques élémentaires, que nous avons présentée dans le chapitre 4, pour l'extraction de caractéristiques plus complexes présentant une bonne résistance au bruit et aux déconnexions du tracé (des caractéristiques directionnelles, des points caractéristiques, des boucles, des cavités, l'épaisseur moyenne du tracé).

La sortie de notre système consiste en une liste de scores qui peut être utilisée pour la validation des hypothèses de segmentation dans un système analytique de reconnaissance de mots. Nous présentons dans le chapitre suivant l'application de notre méthode à la reconnaissance de mots cursifs isolés.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>k</i>	<i>l</i>	<i>m</i>	<i>n</i>	<i>o</i>	<i>p</i>	<i>r</i>	<i>s</i>	<i>t</i>	<i>u</i>	<i>v</i>	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>
<i>a</i>	17	0	2	0	0	0	0	1	0	0	0	0	0	3	0	0	3	0	1	0	1	0	0	0
<i>b</i>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<i>c</i>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<i>d</i>	0	1	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<i>e</i>	5	0	0	1	13	0	0	0	2	0	2	0	1	1	0	0	1	0	0	0	0	0	0	0
<i>f</i>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<i>g</i>	0	0	0	0	0	0	6	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
<i>h</i>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<i>i</i>	2	0	1	1	4	0	0	0	11	1	1	0	0	0	0	2	1	0	0	0	0	0	0	0
<i>k</i>	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<i>l</i>	0	2	0	2	1	0	0	2	1	1	18	0	0	0	0	0	0	2	0	0	0	0	0	0
<i>m</i>	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
<i>n</i>	1	1	0	0	1	0	0	0	0	0	0	2	21	0	0	0	0	0	1	2	0	1	0	0
<i>o</i>	3	0	1	0	3	0	0	0	1	0	0	0	1	8	0	0	2	0	0	0	0	0	0	0
<i>p</i>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
<i>r</i>	0	0	1	0	0	0	0	1	4	0	0	0	1	0	0	9	0	0	0	0	0	0	0	0
<i>s</i>	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
<i>t</i>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	10	0	0	0	0	0	0
<i>u</i>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
<i>v</i>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<i>w</i>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
<i>x</i>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<i>y</i>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1
<i>z</i>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TAB. 5.5 – La matrice de confusion.

Chapitre 6

Reconnaissance holistique de mots cursifs

6.1 Introduction

La reconnaissance de mots cursifs est une tâche très importante avec des applications industrielles telles que la lecture d'adresses postales et la reconnaissance de montants littéraires. Actuellement, aucune solution générale au problème de la reconnaissance de l'écriture cursive n'est encore connue. Bien que, des progrès importants dans ce domaine aient permis le développement de quelques systèmes pratiques, des taux de reconnaissance comparables à la performance humaine n'ont pas été atteints. Pour cette raison, ce problème constitue encore un sujet de recherche intéressant.

Comme nous l'avons mentionné auparavant, la plupart de l'information discriminante est contenue dans la partie singulière du mot cursif (ascendants, descendants, boucles). La méthode d'extraction de caractéristiques directionnelles que nous avons utilisée dans le chapitre précédent est capable d'extraire une partie de cette information (ascendants et descendants) mais elle ne peut pas extraire celle qui est contenue dans la zone centrale du mot (boucles, cavités). Donc, nous avons utilisé pour cette partie la méthode de détection de boucles que nous avons présentée dans le chapitre 4. Nous avons testé cette approche

dans une tâche de reconnaissance de mots cursifs isolés. Nos résultats sont intéressants et même meilleurs que ceux obtenus par d'autres chercheurs sur la même base.

6.2 Architecture

Notre système (Fig. 6.1) comprend trois étapes : pré-traitement, extraction de caractéristiques et reconnaissance.

- **Pré-traitement.** Cette étape reçoit en entrée une image binaire ou en niveaux de gris et donne en sortie une image en niveaux de gris de taille fixe.
- **Extraction de caractéristiques.** Cette étape reçoit en entrée une image en niveaux de gris de taille fixe et donne en sortie un vecteur de caractéristiques directionnelles locales F . Chaque caractéristique directionnelle est calculée à l'aide d'une méthode basée sur la transformée de Hough. Des caractéristiques supplémentaires telles que le nombre moyen de transitions horizontales et des boucles fermées peuvent éventuellement être utilisées pour la reconnaissance de mots cursifs.
- **Reconnaissance.** Cette étape est implémentée au moyen d'un réseau de neurones à trois couches. Ce réseau reçoit en entrée un vecteur de caractéristiques et donne en sortie une liste de scores de classification. La couche d'entrée est complètement connectée à la couche cachée. La couche cachée est complètement connectée à la couche de sortie. Le nombre d'unités d'entrée est égal au nombre de composantes du vecteur de caractéristiques. Le nombre d'unités de sortie est égal au nombre de classes. Le réseau est optimisé afin d'approcher les probabilités a posteriori $P(class = i | F)$ en minimisant l'erreur d'entropie croisée.

6.3 Extraction de caractéristiques

Dans la première étape, nous calculons la transformée de Hough de l'image d'entrée $f(x, y)$. Puis, nous balayons l'espace de paramètres afin d'extraire l'information direc-

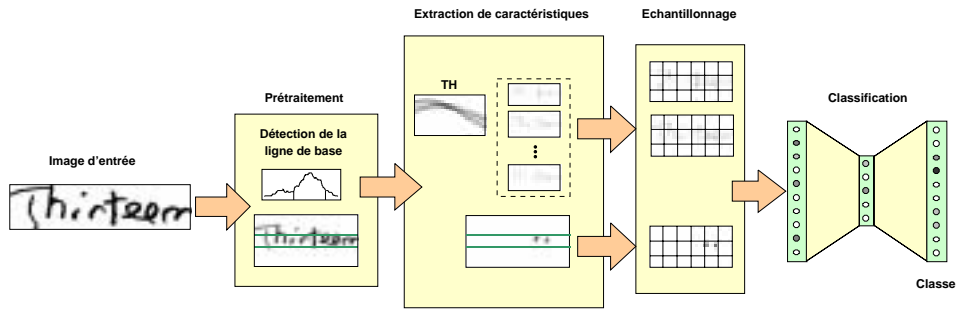


FIG. 6.1 – Architecture du système.

tionnelle globale de l'image. Cette information est représentée sous forme de cartes de caractéristiques de même taille que l'image (Fig. 6.2).

Soit $f(x, y)$ une image en niveaux de gris, soit $h(\rho, \theta)$ sa transformée de Hough, soient N_ρ et N_θ les nombres de niveaux de discrétisation des variables ρ et θ respectivement. Chaque carte de caractéristiques $m_k(x, y)$ correspondant à une orientation θ_k est calculée par :

$$m_k(x, y) = \frac{h(x \cos \theta_k + y \sin \theta_k, \theta_k)}{\sum_{\theta} h(x \cos \theta + y \sin \theta, \theta)} f(x, y)$$

avec $\theta_k \in \{\frac{\pi k}{N_\theta} | k = 0, 1, \dots, N_\theta - 1\}$.

Ces caractéristiques conservent toute l'information contenue dans l'image $f(x, y)$. Nous pouvons vérifier facilement que $f(x, y) = \sum_k m_k(x, y)$.

Comme la transformée de Hough est une transformation globale, l'information directionnelle extraite par cette technique ne reflète pas l'orientation locale des tracés. Nous divisons donc l'image en un ensemble de zones rectangulaires R_{it} , puis nous calculons des caractéristiques locales par :

$$F_{i,t,k} = \frac{\sum_{\forall x, y \in R_{i,t}} m_k(x, y)}{\sum_{j=0}^{N_\theta-1} \sum_{\forall x, y \in R_{i,t}} m_j(x, y)}$$

De par la nature globale de la transformée de Hough et du fait qu'elle ne prend pas en compte la connexité du tracé, l'information contenue dans la zone centrale du mot cursif est moins bien capturée que l'information contenue dans la partie singulière comprenant

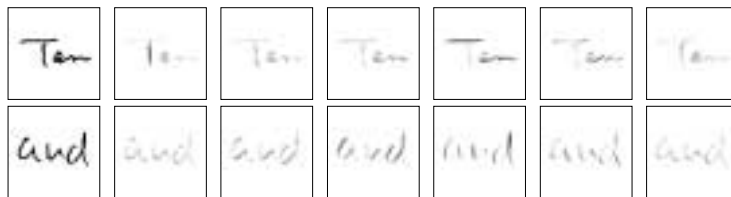


FIG. 6.2 – Extraction de caractéristiques directionnelles. À gauche, des images de test. À droite, les cartes de caractéristiques correspondantes.

des ascendants et des descendants. Nous avons vu dans le chapitre 4 que ce problème peut être résolu en tenant compte de la connexité du tracé. Cependant, cette modification présente l'inconvénient de capturer davantage d'information peu utile pour la reconnaissance (correspondant à la partie régulière). Par ailleurs, parmi les primitives contenues dans la zone centrale, les boucles sont censées être les plus informantes. Nous pouvons donc utiliser la méthode de détection de boucles et de cavités, que nous avons présentée dans le chapitre 4, afin de capturer l'information contenue dans la zone centrale. Notre approche tente ainsi d'extraire l'information contenue dans la partie singulière du mot cursif (ascendants, descendants et boucles) [Simo92] à l'aide de techniques présentant une bonne robustesse au bruit et aux déconnexions.

Dans l'étape suivante, nous détectons les boucles de l'image $f(x, y)$. Comme la position des boucles est en général plus importante que leur taille, nous les codons sous forme d'une carte de caractéristiques $s(x, y)$. Nous divisons donc l'image en un ensemble de zones rectangulaires $R_{i,t}$, puis nous calculons des caractéristiques locales par :

$$B_{i,t} = \sum_{\forall x,y \in R_{i,t}} s(x, y)$$

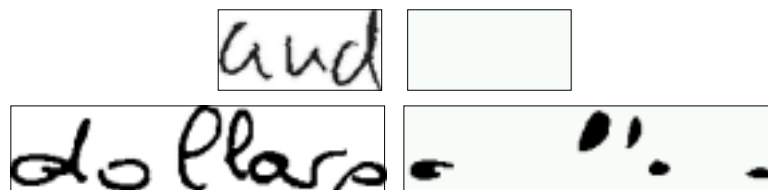


FIG. 6.3 – Détection de boucles.

La figure 6.3 montre des exemples des cartes de caractéristiques obtenues.

6.4 Résultats

Nous avons mis en œuvre différentes versions du système pour effectuer les tâches suivantes : reconnaissance de la première lettre et reconnaissance du mot cursif.

6.4.1 Base de données

Nous avons utilisé une base de données contenant 3.140 images binaires de mots cursifs isolés correspondant à des montants littéraux de chèques anglais (sous-ensemble de la base CENERFIA). La figure 6.4 montre que les mots contiennent différents types de déformations. Le tableau 6.1 montre la distribution des différentes classes dans la base de données. Nous pouvons voir que certaines classes sont très peu nombreuses.

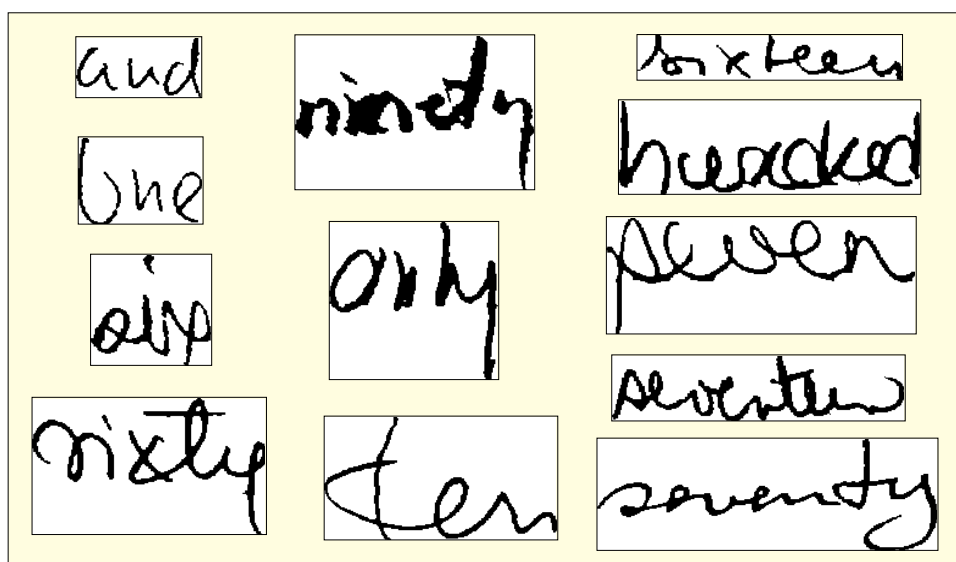


FIG. 6.4 – Quelques exemples de la base de données.

6.4.2 Conditions de test

Nous avons normalisé chaque image en utilisant une estimation de la position de la ligne de base. Puis, nous avons centré le mot dans une image de taille fixe (Fig. 6.5). Comme la plupart des mots de la base de données ne sont pas généralement inclinés, nous n'avons utilisé aucune méthode de correction de l'angle d'inclinaison. La méthode utilisée pour

<i>i</i>	classe	<i>n</i>	<i>i</i>	classe	<i>n</i>	<i>i</i>	classe	<i>n</i>	<i>i</i>	classe	<i>n</i>
1	and	452	9	only	34	17	eleven	44	25	seventy	49
2	one	138	10	eight	99	18	ninety	36	26	sixteen	29
3	six	100	11	fifty	57	19	thirty	71	27	eighteen	36
4	ten	63	12	forty	47	20	twelve	43	28	fourteen	38
5	two	157	13	seven	105	21	twenty	86	29	nineteen	31
6	five	110	14	sixty	48	22	dollars	177	30	thirteen	45
7	four	123	15	three	176	23	fifteen	34	31	thousand	248
8	nine	102	16	eighty	56	24	hundred	539	32	seventeen	37

TAB. 6.1 – Distribution des classes dans la base de données.

détecer la ligne de base est basée sur l'histogramme des transitions horizontales. Après avoir lissé l'histogramme au moyen d'un filtre, nous avons déterminé la position de la ligne de base par une mise en correspondance entre l'histogramme des transitions horizontales et une fonction rectangulaire.

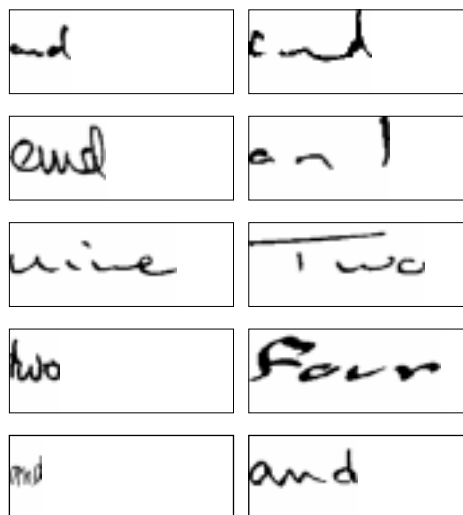


FIG. 6.5 – Exemple des images utilisées pour la reconnaissance de mots cursifs (normalisées par rapport à la zone centrale).

Afin d'obtenir des échantillons pour la reconnaissance de la première lettre du mot, nous avons coupé la portion gauche de chaque image. La taille des images résultantes était donc suffisamment grande afin de couvrir toute première lettre (Fig. 6.6).

Nous avons considéré deux types différents de zones : des zones rectangulaires et des tranches parallèles à orientation variable. Cependant, nos expériences ont montré que ce choix n'a aucun impact appréciable sur la performance du système, sauf que le temps requis

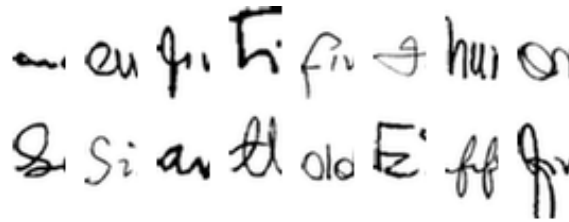


FIG. 6.6 – Exemple des images utilisées pour la reconnaissance de la première lettre.

pour calculer un vecteur de caractéristiques est plus petit pour des zones rectangulaires.

Chaque base de données a été divisée en deux ensembles : un ensemble de test et un ensemble d'apprentissage.

Après initialisation, le réseau de neurones a été optimisé pendant plusieurs cycles par la méthode du gradient conjugué. Nous avons fait varier les valeurs des différents paramètres. Afin de déterminer la complexité du réseau de neurones, nous avons fait varier le nombre d'unités cachées. Après plusieurs essais, nous avons conclu que 72 unités cachées étaient suffisantes. Bien que nous ayons exigé que les sorties du réseau de neurones soient des probabilités, nous avons obtenu des résultats similaires en relâchant cette condition. En ce qui concerne l'architecture du réseau, l'utilisation de réseaux de neurones RBF n'a pas apporté d'améliorations notables par rapport à l'utilisation de Perceptrons multi-couche. D'ailleurs, nous avons observé que la taille des réseaux RBF résultants était beaucoup plus grande que celle des Perceptrons multi-couche.

Pour toutes les expérimentations, les paramètres sont $N_\theta = 6$ et $N_\rho = 64$. Pour la tâche de reconnaissance de mots, la taille de la fenêtre est de 96×48 , le nombre de classes est de 32 et le nombre de zones est de 48. Pour la reconnaissance de la première lettre, la taille de la fenêtre est 32×48 , le nombre de classes est de 8 et le nombre de zones est de 18.

6.4.3 Résultats obtenus

6.4.3.1 Reconnaissance de la première lettre

Le tableau 6.2 montre la performance du système pour la reconnaissance de la première lettre. La performance est de 76, 10% et 96, 77% pour le premier et cinquième choix respec-

tivement. La deuxième ligne du même tableau montre la performance d'un autre système basé sur une analyse de composantes connexes [Guil95a]. Bien que nous n'ayons pas eu recours à l'utilisation d'une méthode de segmentation afin d'isoler la première lettre, la performance de notre système est bonne. Notons aussi que l'autre système utilise une base de données beaucoup plus grande.

choix	1	2	3	4	5	8
Notre méthode	76,10	88,12	92,82	95,16	96,77	97,51
composantes connexes [Guil95a]	79,3	90,0	-	-	96,6	-

TAB. 6.2 – Résumé de résultats pour la reconnaissance de la première lettre de mots cursifs.

La matrice de confusion est montrée dans le tableau 6.3. Nous pouvons voir que la lettre 'o' est souvent reconnue comme 'a' mais pas l'inverse. Ces lettres correspondent d'ailleurs à des mots de longueurs similaires ('and', 'one' et 'only'). D'autre part, les lettres qui sont mieux reconnues correspondent aux lettres présentant le plus grand nombre d'échantillons et aux lettres ayant des extensions ('d', 'f', 'h', 't'). Ce qui est cohérent par rapport à ce que nous pouvions supposer.

classe	<i>a</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>h</i>	<i>n</i>	<i>o</i>	<i>s</i>	<i>t</i>	#	%
<i>a</i>	82	0	0	0	1	3	3	2	5	96	85,42
<i>d</i>	1	25	1	0	1	0	0	2	2	32	78,13
<i>e</i>	1	2	26	3	0	2	0	2	4	40	65,00
<i>f</i>	0	1	0	63	1	0	0	1	7	73	86,30
<i>h</i>	0	1	1	1	98	0	0	2	13	116	84,48
<i>n</i>	4	0	0	0	1	22	0	3	4	34	64,71
<i>o</i>	10	0	0	0	0	2	15	3	1	31	48,39
<i>s</i>	7	8	1	1	4	3	0	43	20	87	49,43
<i>t</i>	3	2	0	1	13	2	0	7	145	173	83,24

TAB. 6.3 – Matrice de confusion pour la reconnaissance de la première lettre.

6.4.3.2 Reconnaissance de mots cursifs

Le tableau 6.4 montre la performance du système pour la reconnaissance de mots cursifs. La performance est de 77,13% et 94,72% pour le premier et cinquième choix respectivement. La deuxième ligne montre la performance du système par l'utilisation d'une technique

d'élimination de poids (pruning). Dans ce cas, la performance est de 78,00% pour le premier choix. La troisième ligne du même tableau montre la performance d'un système plus complexe basé sur des caractéristiques globales et la première lettre [Guil95a] (utilisant une base de données dont la nôtre est un sous-ensemble). La quatrième ligne montre les résultats obtenus par un autre système basé sur des ascendants, des descendants, des boucles et le modèle de l'activation interactif [Côté98] (utilisant la même base de données que nous). Pendant nos expériences, nous avons observé que 68,91% des mots de l'ensemble de test étaient bien reconnus en utilisant 85 caractéristiques qui avaient été calculées en utilisant 12 zones horizontales.

choix	1	2	3	5	10
Notre système	77,13	85,92	90,62	94,72	97,65
Notre système (pruning)	78,00	-	-	-	-
Globale +1ère lettre [Guil95a]	73,5	84,6	-	94,8	98,4
PERCEPTO [Côté98]	73,6	81	-	89,4	92,7

TAB. 6.4 – Résumé de résultats pour la reconnaissance de mots cursifs.

Le tableau 6.5 montre la matrice de confusion. Nous pouvons voir que les mots qui sont mieux reconnus correspondent en général aux mots présentant le plus grand nombre d'échantillons.

La figure 6.7 montre la courbe de rejet. Nous pouvons voir que pour obtenir une performance de 100%, le système doit rejeter 70% des échantillons. D'autre part, pour obtenir une performance de 90%, le système doit rejeter 35% des échantillons.

6.5 Conclusions

Nous avons présenté et testé un système pour la reconnaissance de la première lettre et pour la reconnaissance de mots cursifs utilisant la transformée de Hough et un réseau de neurones. La transformée de Hough est une technique robuste de détection de droites dans des images discrètes. La capacité de cette transformée pour la détection de droites nous permet d'extraire des caractéristiques directionnelles globales présentant une bonne tolérance

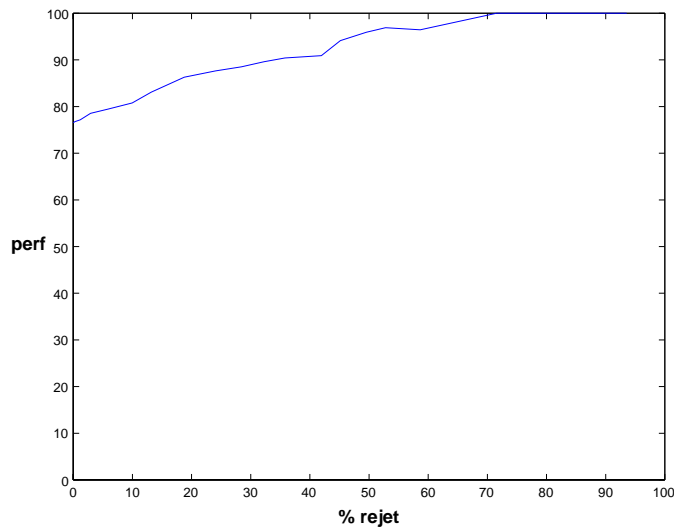


FIG. 6.7 – Courbe de rejet pour la tâche de reconnaissance de mots cursifs.

au bruit et aux déconnexions. Les résultats ont montré que les caractéristiques obtenues présentent une bonne résistance aux déconnexions du tracé et une certaine résistance aux déformations.

La performance du système pour la reconnaissance de la première lettre de mots cursifs est de 76,10% et 96,77% pour les premier et cinquième choix respectivement. La performance du système pour la reconnaissance de mots cursifs est de 77,13% et 94,72% pour les premier et cinquième choix respectivement.

Nous avons testé et comparé notre système avec d'autres systèmes. La performance de notre système est en général bonne. Pour la tâche de reconnaissance de mots cursifs, les résultats sont intéressants parce que nous avons obtenu des résultats meilleurs que ceux obtenus par d'autres auteurs sur la même base. Notons que nous avons utilisé une base de données de taille plus réduite et que nous n'avons pas corrigé l'angle d'inclinaison. D'ailleurs, à l'exception de l'étape de normalisation de taille, nous n'avons utilisé aucun algorithme particulier de pré-traitement.

Le résultat principal de cette étude est que notre méthode d'extraction de caractéristiques est capable d'extraire l'information contenue dans la partie singulière du mot cursif (ascendants, descendants et boucles). Les caractéristiques obtenues par cette méthode présentent une bonne résistance au bruit et aux déconnexions. Cependant, notre méthode

d'extraction de caractéristiques directionnelles n'est pas capable de distinguer les orientations des tracés contenus dans la zone centrale. Ceci constitue en réalité un avantage parce que nous évitons d'extraire information de la partie non informante du mot cursif (la zone régulière). Comme nous avons indiqué dans le chapitre 4, notre méthode peut être utilisée pour extraire d'autres types de caractéristiques présentant une bonne résistance au bruit et aux déconnexions (cavités, points caractéristiques). Donc, il serait intéressant de vérifier la performance de notre système en utilisant ces caractéristiques. Par ailleurs, il serait également intéressant de combiner les résultats donnés par le module de reconnaissance de la première lettre et le module de reconnaissance holistique de mots.

classe	i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
and	1	42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
one	2	0	17	1	1	0	1	0	0	0	0	0	0	0	0	0	0
six	3	0	3	8	0	1	1	0	0	0	0	0	0	0	0	0	0
ten	4	0	0	0	2	2	0	1	0	0	0	0	0	0	0	0	0
two	5	1	0	0	0	10	1	0	1	0	0	0	0	0	0	3	0
five	6	0	0	0	0	1	3	4	0	0	0	0	0	0	0	0	0
four	7	0	0	0	0	0	2	10	0	0	0	0	0	0	0	0	0
nine	8	0	0	0	0	0	1	0	7	0	0	0	0	1	0	0	0
only	9	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
eight	10	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0
fifty	11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
forty	12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
seven	13	0	0	0	0	0	0	0	0	0	0	0	0	6	0	1	0
sixty	14	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0
three	15	0	0	0	0	2	0	0	0	0	0	0	1	0	0	16	0
eighty	16	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	3
eleven	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ninety	18	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
thirty	19	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
twelve	20	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
twenty	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
dollars	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
fifteen	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
hundred	24	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
seventy	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sixteen	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
eighteen	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
fourteen	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nineteen	29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
thirteen	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
thousand	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
seventeen	32	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

TAB. 6.5 – Matrice de confusion pour la tâche de reconnaissance de mots cursifs.

classe	i	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	#	corr
and	1	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	44	95,45
one	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	85,00
six	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	61,54
ten	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	40,00
two	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	62,50
five	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	37,50
four	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	83,33
nine	8	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	11	63,64
only	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	100,00
eight	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	100,00
fifty	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	100,00
forty	12	0	0	2	0	1	0	0	0	0	0	0	0	0	0	0	0	4	25,00
seven	13	0	0	0	1	0	4	0	0	0	0	0	0	0	0	0	0	12	50,00
sixty	14	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	5	80,00
three	15	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	21	76,19
eighty	16	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	7	42,86
eleven	17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	100,00
ninety	18	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	5	20,00
thirty	19	1	0	4	0	1	0	0	0	0	0	0	0	0	1	0	0	8	50,00
twelve	20	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	3	0,00
twenty	21	0	0	0	0	2	0	0	1	0	0	0	0	0	0	0	0	3	66,67
dollars	22	1	0	0	0	0	14	0	0	0	0	1	0	0	0	0	0	16	87,50
fifteen	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0,00
hundred	24	0	0	0	0	0	0	0	61	0	0	0	0	0	0	4	0	66	92,42
seventy	25	0	0	0	0	0	0	0	1	4	0	0	0	0	0	0	0	5	80,00
sixteen	26	0	0	0	0	0	1	0	0	1	2	0	0	0	0	0	0	4	50,00
eighteen	27	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	3	100,00
fourteen	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-
nineteen	29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0,00
thirteen	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-
thousand	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28	0	28	100,00
seventeen	32	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	4	0,00

TAB. 6.6 – Matrice de confusion pour la tâche de reconnaissance de mots cursifs. (Continuation)

Chapitre 7

Reconnaissance perceptive de mots cursifs

7.1 Introduction

Des études expérimentales avec des sujets humains ont montré que le système visuel procède par saccades suivies par des fixations [Tay183]. C'est principalement pendant la fixation qu'un lecteur acquiert l'information afin de reconnaître le mot fixé. Ce fait a également motivé le développement de systèmes de reconnaissance de l'écriture, par exemple la reconnaissance de champs numériques manuscrits [Mart96]. Dans ce cas, le champ numérique est exploré de gauche à droite en une série de saccades. À chaque position, le système vérifie si la fenêtre est centrée sur un caractère. Donc, il produit pour chaque position une liste de scores de classification, donnant ainsi une représentation bidimensionnelle (image ou carte de caractéristiques). Cette représentation est alors utilisée pour reconnaître le champ numérique à l'aide d'une technique de programmation dynamique.

Les lecteurs humains ont tendance à fixer la partie gauche du mot (position préférée de fixation) et reconnaissent plusieurs caractères par fixation. Ceci pourrait permettre aux lecteurs de minimiser le temps de fixation ou le nombre de saccades [Attn54, O'Re90, Legge97]. Ce fait a aussi motivé le développement d'autres systèmes, par exemple pour la

reconnaissance de chaînes de lettres [Mart96, Rayn96] ou pour expliquer la lecture déficiente [Legge97].

Des études expérimentales suggèrent que les lecteurs humains utilisent la forme du mot. La forme du mot peut comprendre : longueur, lettres initiale et finale, caractéristiques internes et contour externe. Certains chercheurs ont exploré l'utilisation de ce type d'information pour la reconnaissance de l'écriture cursive [Guil95a, Higg91].

Dans ce chapitre, nous proposons un nouveau système pour la reconnaissance de mots cursifs basé sur un modèle perceptif et des réseaux de neurones. Dans la section suivante, nous passons en revue les idées principales justifiant notre approche.

7.2 Fondements psychologiques

Les lecteurs humains ont tendance à ne pas apercevoir des transpositions de lettres et des erreurs d'orthographe. D'après certaines études [Tayl83], les lettres et les primitives ne sont pas bien localisées pendant le traitement initial et, dans une étape ultérieure, elles semblent être combinées pour former des objets. Ainsi, pendant le processus de lecture, celles-ci peuvent être multipliées, combinées ou mal localisées. Les détecteurs de caractéristiques (et détecteurs de lettres) peuvent collecter des informations dans une région relativement grande. Deux effets peuvent ainsi avoir lieu : les détecteurs ne savent pas précisément où, dans le champ visuel, la forme a été détectée et ils trouvent difficile de distinguer des occurrences multiples d'une même forme.

Ces idées ont été reprises dans certains systèmes de reconnaissance. Par exemple, le système BLIRTNET présente de l'incertitude dans la prise en compte de la position des lettres [Moze87]. Ceci est dû au fait que les positions relatives des caractéristiques (et lettres) dans leurs champs réceptifs ne sont pas prises en compte par ce système. Certains systèmes ont utilisé la longueur du mot pour coder les positions relatives des lettres [Guil95a]. Par exemple, le système PERCEPTO utilise la logique floue et la longueur du mot pour calculer les positions relatives des lettres afin de prendre en compte l'incertitude

de la position des primitives et lettres détectées [Côté98].

L'incertitude de la position des lettres est donc un facteur important pour la reconnaissance de l'écriture cursive parce qu'il est souvent nécessaire d'avoir une bonne tolérance aux déplacements des caractéristiques. Il est ainsi préférable que les positions de chaque primitive et/ou lettre détectées dans le champ visuel ne soient pas codées dans leurs champs réceptifs respectifs.

Les erreurs commises par des lecteurs humains ont tendance à ne pas affecter les première et dernière lettres du mot et correspondent plutôt à des mots ayant des longueurs proches. Ceci signifie que la longueur et les identités des première et dernière lettres jouent un rôle important dans le processus de lecture humaine [Higg91]. L'extraction de l'information positionnelle est une tâche très difficile à cause de la déformation présentée par l'écriture cursive. Cependant, ce problème peut être compensé par l'utilisation de la longueur du mot comme information a priori [Côté97].

Les lecteurs humains ont tendance à sous-estimer la longueur de chaînes de lettres lorsque l'espacement entre lettres est petit. Dans ce cas, les erreurs sont dues principalement à l'omission de lettres internes et à la fusion de lettres adjacentes [Liu00]. D'ailleurs, le processus de lecture humaine semble avoir lieu de l'extérieur vers l'intérieur du mot et les lettres externes sont mieux reconnues que les lettres internes [Tayl83]. Ceci suggère que l'identité et la position relative des première et dernière lettres puissent constituer des indices importants pour la reconnaissance. D'ailleurs, lorsque la reconnaissance est contrainte par les extrémités du mot, comme dans certains systèmes de reconnaissance de chaînes de caractères, les lettres externes sont mieux reconnues que les lettres internes [LeeS95, Mart96]. Dans ce cadre, les extrémités du mot pourraient constituer des points d'ancrage pour le processus de reconnaissance. Cette contrainte aurait pour effet de faciliter la détection de la première et la dernière lettre (Fig. 7.1). Pour simuler cet effet, la reconnaissance peut ainsi être contrainte par les extrémités du mot.

Certaines études en imagerie cérébrale ont montré que l'accès lexical pourrait avoir lieu dans l'hémisphère gauche du cerveau [Kuri98]. En fait, les parties gauche et droite du mot

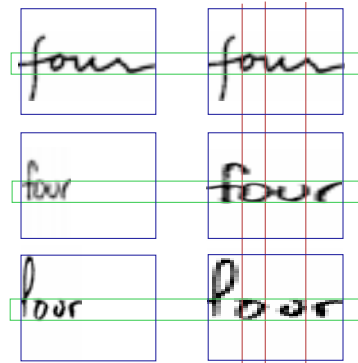


FIG. 7.1 – Incertitude de la position des lettres. A gauche, normalisation par rapport à la zone centrale. A droite, normalisation en largeur.

doivent être traitées indépendamment car l'hémisphère droit reçoit la partie gauche du champ visuel et l'hémisphère gauche reçoit la partie droite [Hube79]. Puis, l'information extraite dans chaque hémisphère est combinée afin d'effectuer l'accès lexical dans un des hémisphères. En principe, chacun des hémisphères est capable de reconnaître des mots, mais l'un d'entre eux est typiquement plus rapide que l'autre. D'ailleurs, le temps de réponse de chaque hémisphère est influencé par la longueur du mot. Donc, le mot doit être fixé à un endroit précis (optimal viewing position) afin de minimiser le temps de fixation et/ou le nombre de saccades [O'Re90]. Ce qui veut dire que l'hémisphère le plus rapide doit reconnaître plus de lettres que l'autre. Il est possible de modéliser cet effet en utilisant une architecture similaire à celle montrée dans la figure 7.2.

Les lecteurs humains reconnaissent usuellement des chaînes de lettres au lieu de lettres individuelles. D'ailleurs, les lettres faisant partie de chaînes régulières de lettres sont mieux perçues que celles faisant partie de chaînes irrégulières de lettres (orthographiquement irrégulières) [Baro73]. Ceci signifie que l'effet de supériorité de mot peut être attribué en partie à la régularité orthographique. Donc, il est possible de prendre en compte cet effet par la reconnaissance de chaînes de lettres (pré-reconnaissance de lettres). Pour cette tâche, une approche basée sur des réseaux de neurones à connexions locales et à poids partagés semble être appropriée (des réseaux de neurones de type TDNN). Afin de prendre en compte l'incertitude dans la position des lettres, nous devons donc normaliser l'image en tenant

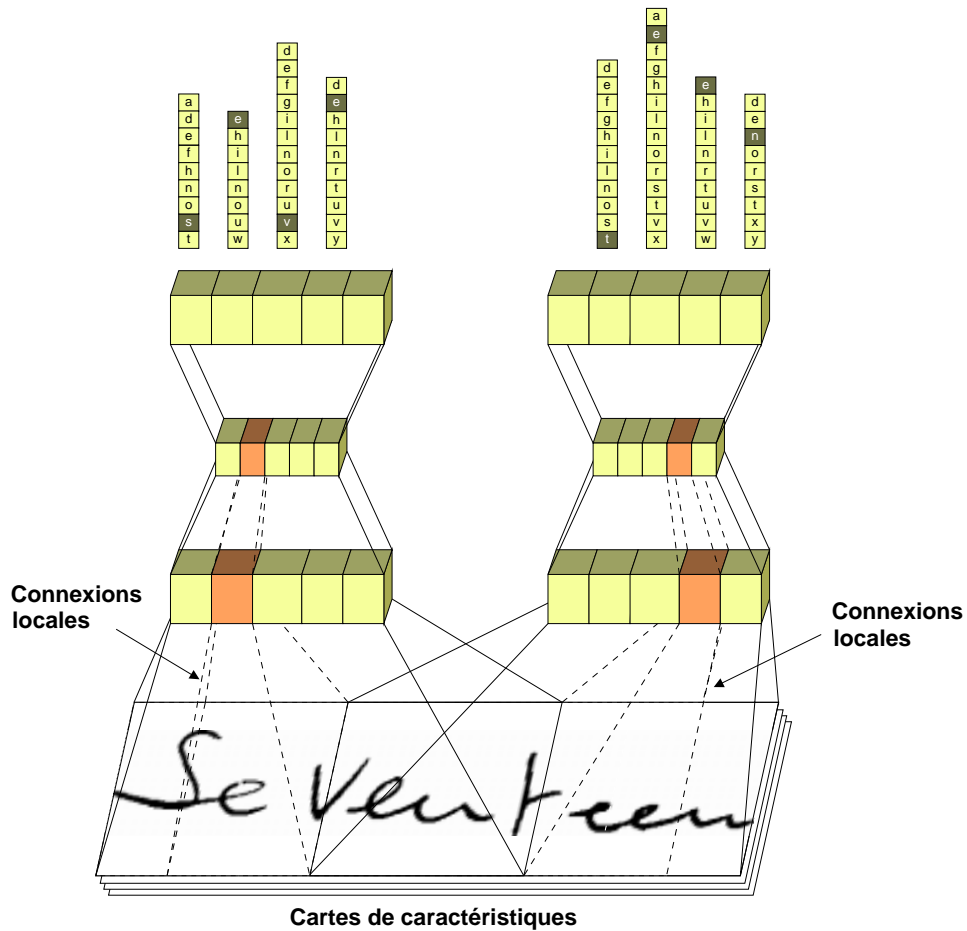


FIG. 7.2 – Architecture possible pour la reconnaissance de chaînes de lettres.

compte de la longueur de chaque mot à reconnaître. Par exemple, pendant l'apprentissage nous pouvons utiliser la longueur du mot comme information a priori. Puis, pour la phase de reconnaissance, nous devons traiter l'image pour des longueurs différentes. Ensuite, nous pouvons utiliser les scores obtenus pour chaque longueur (scores de pré-reconnaissance de lettres) afin de reconnaître le mot à l'aide d'information lexicale par une technique de programmation dynamique ou un réseau de neurones dynamique utilisant un mécanisme d'activation interactif. D'ailleurs, l'utilisation d'un mécanisme d'activation interactif peut nous permettre de modéliser l'accès lexical et l'effet de supériorité du mot [McC181].

7.3 Architecture

Notre système présente une architecture hiérarchique comprenant les étapes suivantes : pré-traitement, extraction de caractéristiques, pré-reconnaissance de lettres et reconnaissance de mots (Fig. 7.3).

1. **Pré-traitement.** Cette étape reçoit en entrée une image en niveaux de gris et donne en sortie une image en niveaux de gris de hauteur constante.
2. **Extraction de caractéristiques.** Cette étape reçoit en entrée une image en niveaux de gris et donne en sortie un ensemble de cartes de caractéristiques.
3. **Pré-reconnaissance de lettres.** Cette étape reçoit en entrée les caractéristiques obtenues dans l'étape précédente et donne en sortie une liste de scores pour chaque longueur de mot. Cette étape est implémentée à l'aide de réseaux de neurones à trois couches, à connexions locales et à poids partagés (TDNN).
4. **Reconnaissance de mots.** Cette étape reçoit en entrée les listes de scores obtenues dans l'étape précédente et donne en sortie une liste de scores représentant le résultat de la reconnaissance de mots. Cette étape est implémentée à l'aide d'un réseau de neurones dynamique à trois couches basé sur le modèle d'activation interactif.

7.4 Pré-traitement

Les images de notre base de données présentent différents types de variations (de taille, de qualité, d'épaisseur, d'inclinaison). Par ailleurs, la position relative des tracés est plus importante que la forme du mot [Dooi91] et la partie singulière du mot (ascendants, descendants et boucles) constitue la partie la plus informante du mot cursif [Simo92]. Il est donc important de prendre en compte la position et la taille de la ligne de base dans le processus de normalisation. D'ailleurs, dans certains systèmes, la détection de la ligne de base est très importante parce que d'elle dépend la précision des algorithmes de détection de primitives (d'ascendants et de descendants) [Côté97].

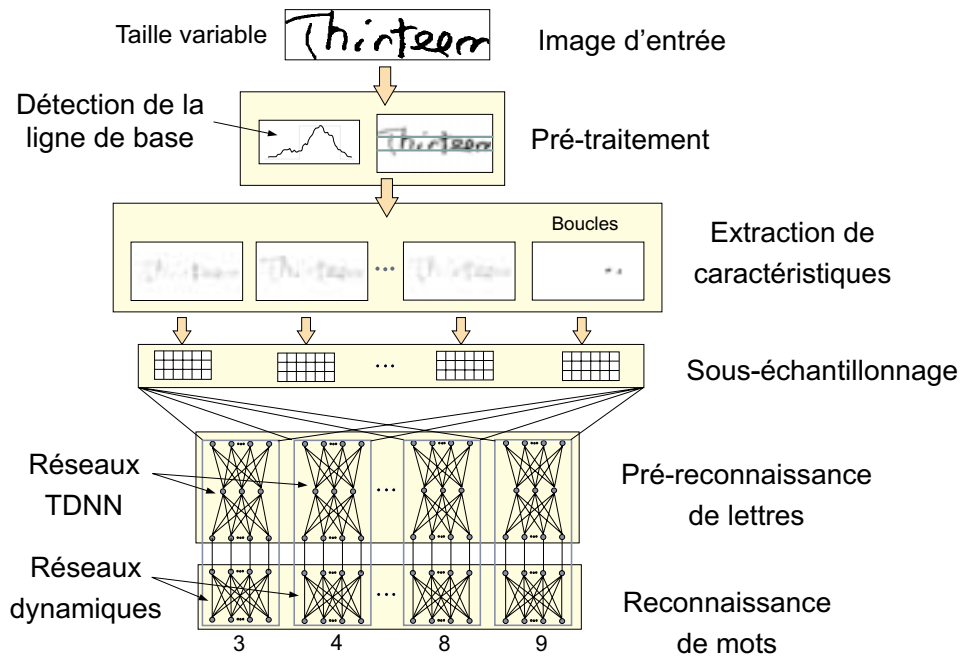


FIG. 7.3 – Architecture du système.

Dans la première étape, nous calculons l’histogramme des transitions horizontales. Après avoir détecté la position de la ligne de base, nous centrons le mot par rapport au centre de la ligne de base. Puis, nous mettons à l’échelle l’image résultante afin d’obtenir une image en niveaux de gris de hauteur constante (Fig. 7.4).

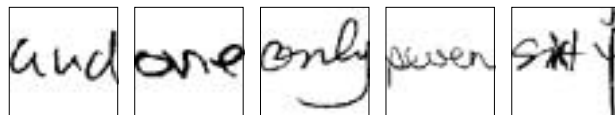


FIG. 7.4 – Normalisation de la taille des images.

La performance de certains systèmes est en général affectée par des variations de l’épaisseur du tracé. Pour cette raison, il est usuel d’utiliser une étape de squelettisation dont la fonction est d’éliminer l’épaisseur du tracé. Malheureusement, la squelettisation élimine généralement des informations importantes. Pour cette raison, d’autres approches ont plutôt utilisé des représentations qui sont indépendantes de l’épaisseur du tracé. Cependant, ces approches sont sensibles à des erreurs dans l’estimation de l’épaisseur du tracé [Zhu99]. D’autres approches se contentent donc d’uniformiser l’épaisseur du tracé par l’ajout ou suppression de pixels au moyen de masques binaires [Hu96]. Par ailleurs, les systèmes basés

sur la transformée de Hough requièrent en général que la variabilité de l'épaisseur du tracé soit traitée dans une étape indépendante [Kalv95]. Dans notre cas, nous ne cherchons pas à réduire l'image à sa représentation en squelette. Nous cherchons plutôt à uniformiser et/ou normaliser l'épaisseur du tracé.

D'abord, nous estimons l'épaisseur moyenne du tracé à l'aide d'une technique que nous avons présentée dans le chapitre 4. Puis, nous normalisons l'épaisseur du tracé à l'aide d'une technique de corrélation. Plus précisément, nous utilisons une technique de corrélation avec des disques circulaires afin de provoquer la diffusion des contours. Cette technique présente la particularité que, lorsque le diamètre du disque est égal à l'épaisseur du tracé, l'image résultante présente des pics au centre du tracé. Cette propriété peut être exploitée pour normaliser l'épaisseur du tracé. La corrélation peut être implémentée à l'aide de méthodes efficaces comme la transformée rapide de Fourier (FFT), voir chapitre 4.

La figure 7.5 montre quelques exemples des images obtenues par cette méthode. Nous pouvons remarquer que cette méthode est particulièrement efficace pour des images de caractères parce que dans ce cas les tracés sont plus épais. C'est d'ailleurs pour cette raison que nous avons considéré l'utilisation de cette méthode comme optionnelle pour des mots.

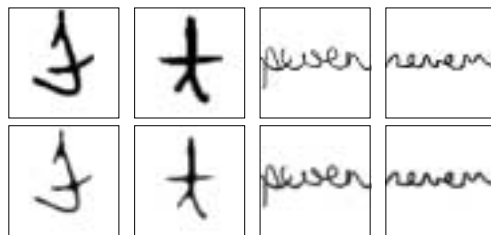


FIG. 7.5 – Normalisation de l'épaisseur du tracé. En haut, des images de test. En bas, les images normalisées.

7.5 Extraction de caractéristiques

L'extraction de caractéristiques est l'une des étapes les plus importantes de tout système de reconnaissance. L'extraction de caractéristiques vise à extraire de l'image l'information pertinente permettant de distinguer plus facilement les différentes classes.

Tous les mots et caractères cursifs peuvent être représentés soit par des segments de droite et des courbes ayant des orientations et positions relatives données, soit par des points caractéristiques de l'image (extrémités de segments, extrémums de courbure), soit par des primitives telles que des ascendants, des descendants, des boucles et des cavités. Nous pouvons donc reconnaître chaque caractère ou mot à partir d'informations telles que l'orientation et la position relative de chacune de ses primitives. Donc, pour cette partie, nous utilisons la méthode générale d'extraction de caractéristiques que nous avons présentée dans la section 4.4.

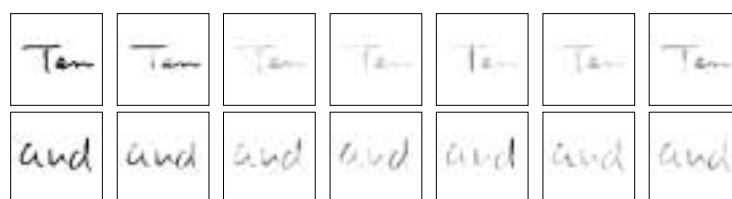


FIG. 7.6 – Cartes de caractéristiques directionnelles. A gauche, des images de test. A droite, les cartes de caractéristiques respectives.

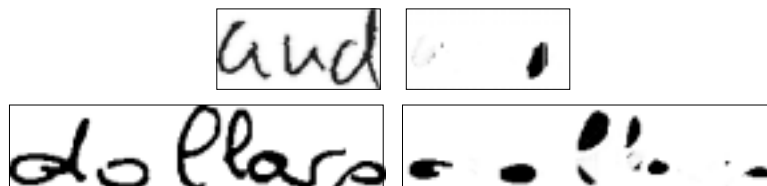


FIG. 7.7 – Cartes de boucles. A gauche, les images de test. A droite, les cartes de caractéristiques respectives.

Les figures 7.6 et 7.7 montrent quelques exemples des cartes de caractéristiques obtenues. Nous pouvons constater que cette méthode présente une bonne tolérance aux déconnexions du tracé (comparer avec la figure 6.3). En fait, toutes les boucles fermées et même certaines boucles ouvertes sont détectées par cette méthode.

7.6 Pré-reconnaissance de lettres

Ce module reçoit en entrée les cartes de caractéristiques obtenues dans l'étape précédente et donne en sortie une liste de scores. Nous avons utilisé des réseaux de neurones

pour cette tâche car ils constituent d'excellents classifieurs et sont capables d'approcher des probabilités a posteriori [Ney95]. D'ailleurs, des réseaux de neurones statiques ont été appliqués avec succès à la reconnaissance de l'écriture manuscrite [Breu97]. A titre d'exemple, des réseaux de neurones de type TDNN (Time Delay Neural Network) combinés avec la programmation dynamique ont été utilisés avec succès à la reconnaissance de champs cursifs [LeeS95]. Dans ce cas, comme la reconnaissance est contrainte par les extrémités du mot, les lettres externes sont mieux reconnues que les lettres internes [Leco94b, Côté97, LeeS95, Mart96].

En général, la reconnaissance d'une lettre est affectée par la présence, dans le champ réceptif, de tracés appartenant à des lettres adjacentes (lateral masking, visual interference) [Boum73]. Pour cette raison, dans le cas de la première et la dernière lettre, certains systèmes procèdent à l'élimination de ces tracés par une méthode d'analyse de composantes connexes [Guil95a]. Un inconvénient de cette méthode est qu'elle produit usuellement des caractères composés (Fig. 7.8). Une meilleure solution à ce problème est donc d'utiliser une méthode plus coûteuse de segmentation (des modèles de Markov cachés, des réseaux TDNN ou des techniques similaires).



FIG. 7.8 – Caractères composés produits par l'analyse de composantes connexes [Guil95a].

Des études expérimentales ont montré que le système visuel reconnaît un nombre limité de lettres à la fois [Liu00]. Il semble en effet que le système visuel soit contraint par le nombre de détecteurs de lettres. Le système visuel pourrait ne pas tester un grand nombre d'hypothèses de segmentation comme certains systèmes le suggèrent. D'autre part, les propriétés d'invariance du système visuel peuvent être modélisées par l'utilisation de répliques de détecteurs couvrant tout le champ visuel. Pour cela, les approches TDNN sont particulièrement adaptées [LeeS95]. D'ailleurs, l'utilisation de poids partagés se traduit généralement par une bonne performance.

Le module pour la pré-reconnaissance de lettres est montré dans la figure 7.9. Il comporte plusieurs réseaux de neurones à connexions locales et à poids partagés reconnaissant chacun des chaînes de lettres de longueur donnée. Chaque sous-réseau est en fait une réplique d'un seul réseau à poids partagés. Comme dans le cas des réseaux de type TDNN, ce sous-réseau est composé de plusieurs répliques d'un seul détecteur de lettres. On calcule donc la sortie du réseau pour chaque longueur de chaîne (3, 4, ..., 9). C'est à dire que notre système traite l'image à différentes échelles. A la différence d'autres approches connexionnistes, notre réseau de neurones opère sur les caractéristiques obtenues dans l'étape précédente (les caractéristiques directionnelles et les boucles).

Nous optimisons le réseau en minimisant l'erreur d'entropie croisée par la méthode de descente du gradient. Pour la phase d'apprentissage, nous normalisons chaque image en utilisant la longueur du mot comme information a priori. Pour la phase de reconnaissance, nous calculons la sortie du réseau pour chaque longueur différente (3 à 9). Notre approche correspond donc à une approche multi-échelle utilisant des réseaux de neurones à poids partagés de type TDNN.

7.7 Reconnaissance de mots

Pour la reconnaissance de mots, nous utilisons un réseau de neurones dynamique qui reçoit en entrée les scores obtenus dans l'étape précédente et donne en sortie une liste de scores de classification. Ce réseau modélise l'accès lexical et utilise un mécanisme d'activation interactif [McCl81].

Le réseau comprend trois couches : une couche de nœuds d'entrée, une couche d'unités-lettre et une couche d'unités-mot. Chaque nœud de la couche d'entrée est connecté à seulement une unité lettre. Dans la deuxième couche, les unités sont arrangées en groupes selon la position de chaque lettre. Chaque unité lettre est alors connectée à une ou plusieurs unités-mots, en fonction des mots dont elle fait partie à la position appropriée. Par exemple, l'unité "five" est activée par les unités 'f', 'i', 'v' et 'e' de la première, seconde, troisième et

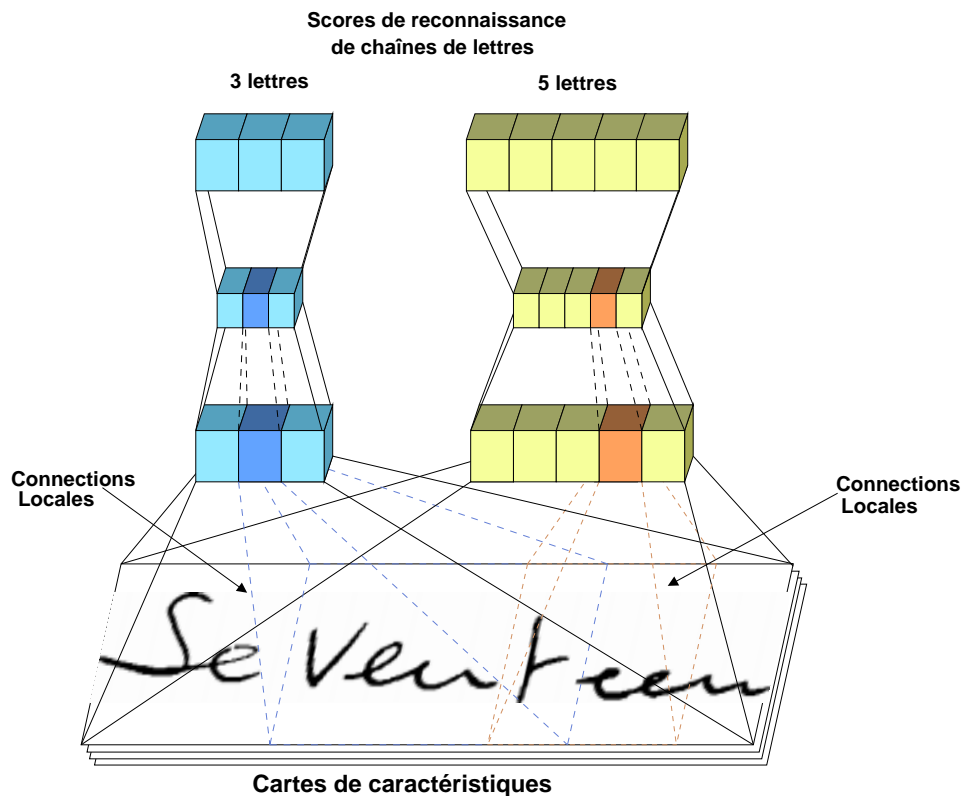


FIG. 7.9 – Architecture pour la pré-reconnaissance de lettres.

quatrième positions respectivement.

Une caractéristique importante de notre réseau est qu'il ne comporte aucune connexion inhibitrice (Fig. 7.10). Ceci est dû au fait que l'inhibition peut provoquer des pertes d'information avant que les unités-mots aient accumulé suffisamment d'information. Ce qui peut se traduire par des erreurs dues à une prise de précision prématurée [Côté97]. D'autre part, nous considérons que le processus ascendant a pour fonction de propager l'information disponible vers le niveau haut. En principe, la reconnaissance doit être possible en utilisant exclusivement un processus ascendant car la capacité discriminante du réseau dépend en grande partie de ce processus [Alme90]. A titre d'exemple, la capacité de stockage des réseaux de Hopfield est en général basse [Hert91]. Par contre, la capacité de stockage d'un réseau statique (feed-forward) est en général importante. Dans ce cadre, le processus descendant a pour fonction de compléter le processus de reconnaissance en propageant l'information accumulée dans le niveau mots. Donc, ce processus doit éventuellement résoudre

des ambiguïtés et compléter des lettres manquantes (pattern completion).

Dans le module précédent, les lettres externes sont en général mieux reconnues que les lettres internes. Pour cette raison, les unités correspondant à la première et dernière lettres atteindront plus rapidement leurs niveaux de saturation. Cela veut dire que notre système peut simuler la reconnaissance de l'extérieur à l'intérieur du mot.

A la différence du système PERCEPTO qui utilise le modèle de vérification et le modèle d'activation interactif combinés avec une méthode heuristique de mise en correspondance floue, nous utilisons le modèle d'activation interactif afin de modéliser l'accès lexical et nous tenons compte de l'effet de supériorité du mot par la pré-reconnaissance de lettres.

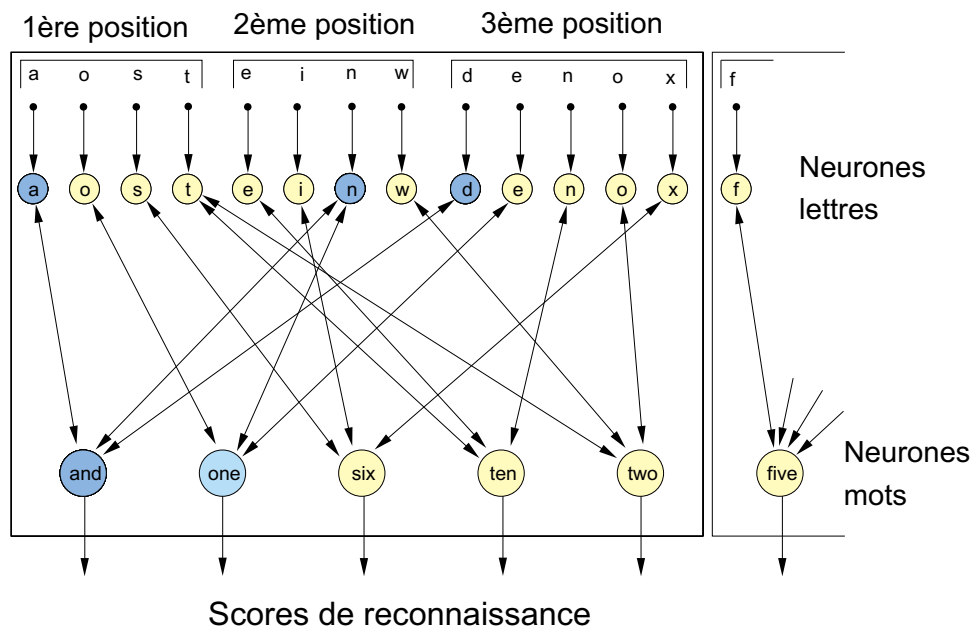


FIG. 7.10 – Détail des connexions entre les niveaux correspondant aux mots de 3 et 4 lettres. (Notons que le réseau ne comporte aucune connexion inhibitrice et qu'il n'existe pas de connexions entre niveaux.)

7.7.1 Apprentissage

Les valeurs des connexions du modèle d'activation interactif sont usuellement choisies soit empiriquement, soit par des méthodes heuristiques [Côté98, McCl81]. Un inconvénient de ces approches est qu'elles ne prennent pas en compte la capacité informante des lettres. A l'état stable, le modèle d'activation interactif est équivalent à un réseau de neurones

statique. Donc, l'apprentissage du réseau peut être effectué en utilisant les mêmes méthodes d'apprentissage que pour d'autres réseaux de neurones dynamiques [Alme90].

Nous avons donc utilisé l'algorithme d'apprentissage suivant :

1. Initialiser les poids du réseau,
2. choisir au hasard un exemple à apprendre,
3. recalculer les sorties des unités,
4. actualiser les poids en utilisant la règle delta et
5. répéter les pas 2 à 4 pendant un certain nombre d'itérations.

Cependant, avant d'utiliser cet algorithme d'optimisation, nous devons transformer le réseau en une forme plus adéquate. Il est en particulier nécessaire de déterminer explicitement la fonction de transfert.

Soit η_i l'activation totale, soient M et m les valeurs maximale et minimale, soit θ_i le taux de décroissance, soit r_i le niveau de repos, soit N_i le voisinage de l'unité i . A l'état stable, la sortie $a_i(\eta)$ de l'unité i peut être exprimée par :

$$a_i(\eta_i) = f(\eta_i) = \frac{\theta_i \gamma_i + \frac{M}{2}(|\eta_i| + \eta_i) + \frac{m}{2}(|\eta_i| - \eta_i)}{\theta_i + |\eta_i|} \quad (7.1)$$

avec

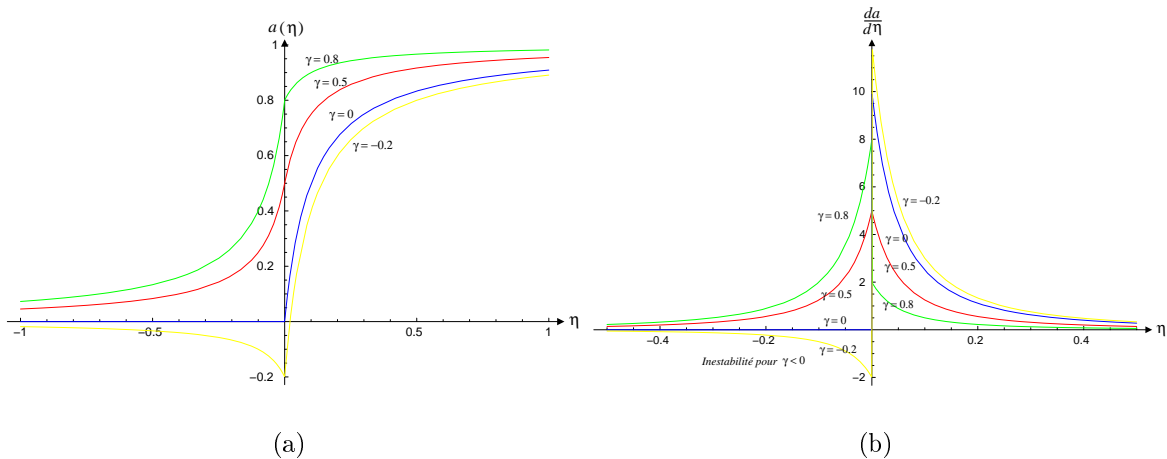
$$\eta_i = \sum_{j \in N_i} w_{ij} a_j \quad (7.2)$$

Cette fonction peut prendre les formes indiquées dans le tableau 7.1. Afin de pouvoir interpréter les valeurs de sortie dans un cadre probabiliste, nous utilisons une fonction de transfert sigmoïdale avec $M = 1$, $m = 0$ et $\theta = 0,01$. Par ailleurs, comme la stabilité du réseau requiert que nous ayons $\frac{d}{d\eta} f(\eta) > 0$, nous devons utiliser $\gamma \geq 0$ (Fig. 7.11).

Soit x_i l'entrée, soit y_i la sortie désirée de l'unité i , soit α le taux d'apprentissage. Les poids de connexion w_{ij} sont mis à jour par la formule :

$$w_{ij}^{t+1} = w_{ij}^t + \alpha(y_i - f(\eta_i))x_i$$

M	m	θ	γ	Forme
1	0	0	0,5	Fonction de Heaviside
1	0			Fonction sigmoïdale
1	-1	0	0	Fonction signe
1	-1		0	Fonction de Hill

TAB. 7.1 – Cas spéciaux de la fonction de transfert $f(\eta)$.FIG. 7.11 – Influence du paramètre γ sur la forme de la fonction de transfert. a) La fonction de transfert $f(\eta)$ et b) sa dérivée ($M = 1$, $m = 0$ et $\theta = 0, 1$).

Dans le cas d'une fonction sigmoïdale, nous pouvons démontrer facilement que cette formule minimise l'erreur d'entropie croisée. Ce qui veut dire que les sorties peuvent correspondre à des probabilités.

La capacité de stockage du réseau est en général un facteur très important. Par exemple, la capacité de stockage des mémoires de Hopfield est approximativement $N/\ln N$ (où N représente le nombre d'unités). D'ailleurs, la capacité de ce type de réseau ($162/\ln 162 \approx 31,8$) est inférieure à celle requise par notre application (32). Cela est dû au fait que la matrice de connexions doit être symétrique.

Quant à notre réseau, nous avons observé que sa capacité dépend du paramètre γ comme le montre la figure 7.12. D'ailleurs, comme ce paramètre provoque des distorsions qui peuvent se traduire par une instabilité du réseau, nous avons remplacé la formule (7.2) par :

$$\eta_i = \sum_{j \in N_i} w_{ij} a_j + b_i \quad (7.3)$$

où b_i est le biais de l'unité i et $\gamma = 0,5$.

Nous avons optimisé ce réseau en utilisant des exemples non bruités de chaque classe et nous avons analysé les valeurs des connexions résultantes. Nous avons vu que certaines connexions en sortie d'un neurone-lettre sont pratiquement nulles lorsque la lettre correspondante fait partie de plus d'un mot. Cela veut dire que cette lettre est très peu informante. Cela est dû au fait que certaines lettres n'apportent aucune information pour la discrimination de certains mots (la lettre 'n' de 'and' et 'one'). Par ailleurs, les connexions descendantes (mot-lettre) ont toutes la même valeur. De même, les connexions non-nulles en entrée d'un neurone-mot ont toutes la même valeur, mais varient d'un mot à l'autre et leur somme a la même valeur pour tous les mots. Également, le biais b_i a la même valeur pour tous les neurones. Le réseau présente donc très peu de paramètres et ne nécessite pratiquement pas de phase d'apprentissage. D'ailleurs, comme le réseau ne comporte que des connexions excitatrices, il est possible d'utiliser d'autres fonctions de transfert à support positif. Par exemple, nous pouvons utiliser $\gamma = 0$, $M = 1$ et $m = 0$ qui correspond à une variante de la fonction de Hill. Dans ce cas, le biais b_i de chaque neurone est égal à zéro.

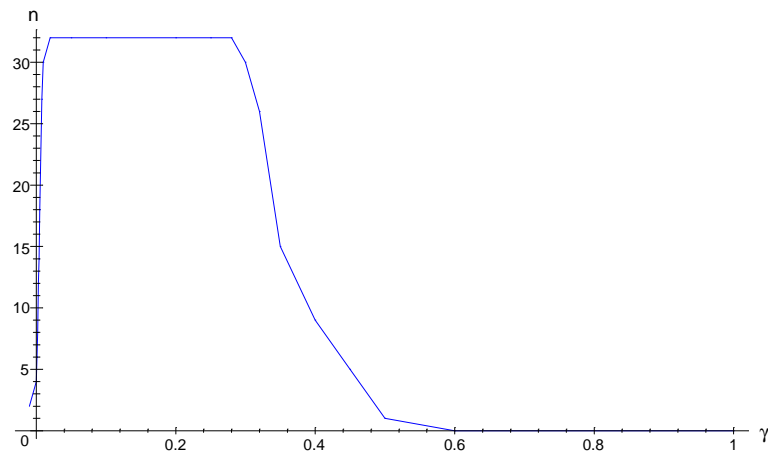


FIG. 7.12 – Nombre de mots appris en fonction de γ lorsque le réseau est utilisé comme mémoire associative. ($M = 1$, $m = 0$, $\theta = 0,01$, $nunits = 162$, $noutputs = 32$)

Le réseau effectue une séparation linéaire de l'espace d'entrée. D'ailleurs, pour des mots de même longueur, la frontière de décision implémentée par le réseau est déjà optimale.

Par exemple, la frontière de décision pour les mots 'and' et 'one' est donnée par :

$$x_{1a} + x_{3d} = x_{1o} + x_{3e}$$

où x_{1a} , x_{1o} représentent les entrées correspondant aux lettres 'a' et 'o' dans la première position et x_{3d} , x_{3e} représentent les entrées correspondant aux lettres 'd' et 'e' dans la troisième position. Ce qui correspond à l'équation de l'hyperplan restant le plus éloigné possible des deux classes ('and' et 'one') représentées par un point dans l'hypercube de Hamming. En général, cette condition implique que la somme des poids à l'entrée d'une unité-mot soit constante (c. à d. $\sum_j w_{ij} = CTE$).

7.7.2 Reconnaissance

Afin de modéliser l'accès lexical, nous utilisons un mécanisme activation interactif.

Donc, le processus de reconnaissance est comme suit :

1. initialiser les unités à leurs niveaux de repos ($a_i^0 = 0$),
2. présenter un vecteur d'entrée \mathbf{x} ,
3. calculer les valeurs d'activation η_i de toutes les unités par la formule (7.3),
4. mettre à jour les sorties de toutes les unités par la formule :

$$a_i^{t+1} = a_i^t - \mu(\theta_i + |\eta_i|)(a_i^t - f(\eta_i))$$

5. répéter les pas 3 à 4 pendant un nombre prédéfini de cycles ou jusqu'à ce que la valeur d'une unité-mot quelconque atteigne une valeur prédéfinie.

7.8 Résultats

Nous avons testé notre système sur une base de données contenant 3.410 images binaires de montants littéraux. Le nombre de classes est égal à 32.

Afin d'obtenir une estimation non biaisée de la performance du système, nous avons utilisé 10% des échantillons de la base de données pour l'ensemble de test et la partie restante pour l'ensemble d'apprentissage. Nous avons traité chaque image et nous avons effectué l'extraction de caractéristiques. Dans l'ensemble d'apprentissage, chaque mot a été codé sous forme de chaîne de lettres en utilisant le codage 1 parmi n pour chaque position lettre. Puis, nous avons optimisé le module de reconnaissance de chaînes de lettres. Finalement, nous avons créé le module de reconnaissance de mots. Ce module pouvait être ensuite optimisé afin de prendre en compte l'information de fréquence de mot. Cependant, nous voulions savoir quelle était la performance du système sans l'utilisation de cette information.

Pour les tests, nous avons fixé le nombre de zones horizontales en fonction de la longueur du mot. Donc, nous avons utilisé 18, 22, 26, 30, 34, 38 et 42 zones horizontales pour des mots de 3, 4, 5, 6, 7, 8, et 9 lettres respectivement. Egalement, le nombre de zones verticales N_v a été fixé à 6 et le nombre d'orientations N_θ a été fixé à 6 de telle sorte que le nombre total de caractéristiques est égal à 756 pour des mots de 3 lettres. Pour la détection de boucles, nous avons utilisé fixé N_θ à 6. Dans la première couche du module de reconnaissance de chaînes de lettres, la taille du champ réceptif est égale à 4×6 avec un recouvrement de 2 pixels en direction horizontale. Dans la couche caché, le nombre de cartes de caractéristiques est égal à 4 et la taille du champ réceptif est égale à 4×1 avec un recouvrement horizontal de 2 pixels. Dans la couche de sortie, le nombre d'unités est égal au nombre de lettres multiplié par 26.

Nous avons implémenté ce système à l'aide de la bibliothèque de fonctions décrite dans [Hoek96]. Comme elle ne comporte aucune fonction de support pour la création de réseaux de neurones dynamiques, nous avons implémenté nos propres fonctions. D'ailleurs, nous avons adapté la fonction d'apprentissage des réseaux de neurones pour minimiser l'erreur d'entropie croisée. Également, nous avons implémenté les fonctions de transfert softmax et IAMsigmoid requises par notre module de reconnaissance de mots.

Les résultats préliminaires sont montrés dans le tableau 7.2. La performance du système pour la reconnaissance de mots de 3, 4 et 5 lettres est de 90%, 81% et 89% respectivement.

La performance du système pour la reconnaissance de mots de 3 et de 4 lettres est de 79%. Il est important de souligner que nous utilisons une base de données relativement petite et que nous n'avons utilisé aucune information de fréquence de classe (mot) et que nous n'avons pas fait varier la taille du réseau utilisé pour la reconnaissance de chaînes de lettres afin de déterminer sa complexité.

Longueur	% correct
3	90
3, 4	79
4	81
5	89

TAB. 7.2 – Performance du système pour des mots de différentes longueurs.

7.9 Extensions possibles du système

Toute méthode de reconnaissance de l'écriture implique usuellement l'utilisation d'un certain type de segmentation [Leco94a] : soit une segmentation en primitives (boucles, ascendants, cavités ou autres types de primitives), soit une segmentation en régions (champs réceptifs, fenêtres d'observation), soit une segmentation en lettres, en graphèmes ou en pseudo-lettres. D'ailleurs, comme le suggèrent certaines études sur la vision humaine, la segmentation semble être l'une des fonctions du système visuel. Par exemple, le modèle de reconnaissance par composantes assume que les objets sont reconnus par l'analyse de leurs parties [Bied85]. Ce modèle de la vision humaine présente la particularité d'être résistant à des occlusions. D'ailleurs, certaines expériences ont montré que les objets sont parfaitement reconnus lorsque leurs caractéristiques distinctives (geons) restent visibles. Dans ce cadre, les points de concavité constituent des candidats pour la segmentation des objets et les parties résultant de la segmentation sont censées être faciles à reconnaître.

Certaines études sur la reconnaissance de mots imprimés ont suggéré que les graphèmes puissent constituer les unités de base du processus de lecture humaine [Rey00]. En psychologie de la lecture, les graphèmes sont considérés comme la représentation écrite des

phonèmes tandis que, dans le domaine de la reconnaissance de l'écriture, les graphèmes sont considérés comme les unités de base de la segmentation du mot cursif et sont censés correspondre à des classes faciles à reconnaître [Leco90, Leco91]. Ainsi, un caractère se décompose au maximum en deux ou trois graphèmes, et à l'inverse un graphème est composé, au maximum, de deux ou trois caractères. D'ailleurs, des systèmes basés sur la segmentation du mot en graphèmes ont atteint des résultats très intéressants dans plusieurs études [Kner98a, Kner98b, Gors99]. Usuellement, le mot est d'abord traité afin de corriger l'angle d'inclinaison, puis segmenté en graphèmes. Les graphèmes obtenus sont alors reconnus individuellement à l'aide de réseaux de neurones ou d'autres types de classifieurs. Ensuite, le mot est reconnu au moyen d'une technique basée sur la programmation dynamique ou des modèles de Markov cachés. Afin de tenir compte de ces faits, nous avons envisagé l'addition d'une étape de segmentation en graphèmes suivie d'une étape de reconnaissance de graphèmes (Fig. 7.13).

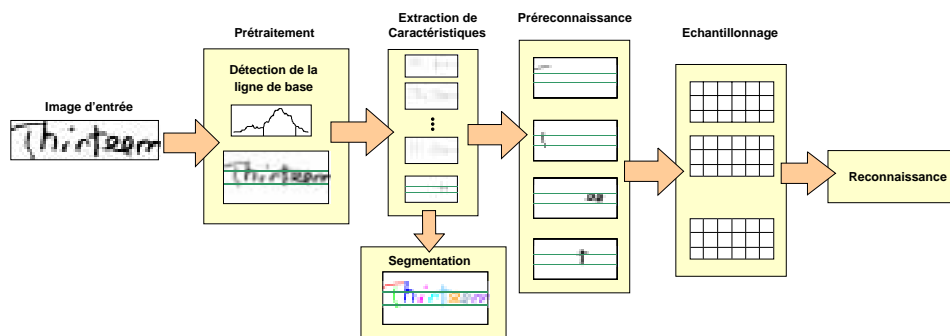


FIG. 7.13 – Extension possible du système.

Certaines études sur la vision humaine ont montré que, sous certaines conditions, le temps de réponse pour la détection d'un objet est indépendant du nombre d'objets dans la scène visuelle. Par exemple, lorsque deux textures diffèrent par la présence d'une caractéristique donnée, la ségrégation des textures semble être automatique (effet pop-out) et pré-attentive [Jule83]. A l'inverse, lorsque les objets comportent les mêmes types de primitives, le temps de réponse pour la détection d'un objet dépend du nombre d'objets dans la scène visuelle (distractors) [Trei85]. Ainsi, lorsque deux textures comportent les mêmes types de primitives (arrangées différemment), la segmentation des textures n'est

plus automatique (parallèle) mais séquentielle. Cela veut dire que la vision humaine doit comprendre deux modes distincts de fonctionnement [Trei85] : un mode pré-attentif de traitement en parallèle qui produit une représentation bidimensionnelle de la scène visuelle en termes de primitives primaires et un mode attentif qui cherche à détecter des objets reconnaissables ou des combinaisons de primitives parmi l'ensemble de primitives détectées. Certaines études ont montré que le temps de réponse pour la reconnaissance de caractères présentant une structure complexe dépend fortement de leurs complexités. A titre d'exemple, le temps de réponse pour la reconnaissance de caractères chinois semble être une fonction du nombre de tracés (stroke-number effect) [Yang98]. Donc, il est possible que la reconnaissance de caractères et de mots ait lieu principalement durant l'étape attentive. Par ailleurs, les caractéristiques de même type semblent être codées dans une même carte de caractéristiques [Trei86] en préservant l'information spatiale. C'est d'ailleurs pour cette raison que nous avons utilisé dans notre système une représentation de ce type. Pour le codage des graphèmes, nous avons donc envisagé une approche de ce type. C'est à dire que nous coderions dans chaque carte la présence d'une classe différente de graphèmes en préservant l'information spatiale. Puis, nous utiliserions cette représentation pour la pré-reconnaissance de lettres à l'aide d'un réseau de neurones à poids partagés (TDNN). Le résultat de cette étape serait ensuite utilisé pour la reconnaissance du mot inconnu à l'aide un réseau de neurones utilisant un mécanisme d'activation interactif.

Le système visuel humain est capable de reconnaître des caractères isolés (et des objets) lorsque certaines de leurs parties restent visibles. Ceci pourrait indiquer que chaque caractère (ou objet) est d'abord décomposé en ses parties, puis elles sont assemblées afin de former des caractères (ou objet) valides au moyen d'un mécanisme d'activation interactif. Pourvu qu'on puisse coder chaque caractère (ou objet) en termes d'un ensemble donné de primitives avec leurs positions (ascendants, descendants, boucles, etc.) et qu'on puisse implémenter un système capable d'obtenir cette représentation, il semble plausible d'utiliser un réseau de neurones comportant un mécanisme d'activation interactif afin de modéliser la reconnaissance humaine de caractères (et d'objets). Dans ce cas, la reconnaissance pourrait

bien avoir lieu de l'extérieur à l'intérieur du caractère (ou objet) si les primitives externes sont mieux reconnues que les primitives internes.

7.10 Conclusions

Nous avons présenté un système de reconnaissance de mots cursifs basé sur un modèle perceptif et des réseaux de neurones. Il comprend les étapes suivantes : pré-traitement, extraction de caractéristiques, pré-reconnaissance de lettres et reconnaissance de mots.

Notre système présente les caractéristiques suivantes :

- architecture connexionniste,
- mécanisme d'activation interactif,
- traitement parallèle de l'information,
- modèle d'accès lexical,
- prise en compte de l'effet de supériorité de mot,
- effet de régularité de séquences de lettres,
- incertitude dans la position de lettres,
- reconnaissance de l'extérieur à l'intérieur du mot et
- primitives biologiquement plausibles.

Nous avons considéré des problèmes tels que la capacité informative des lettres, la capacité discriminante et la capacité de stockage du réseau. La capacité de stockage et la capacité discriminante d'un réseau de neurones statique (opérant en mode ascendant ou feed-forward) sont en général importantes. Au contraire, la capacité d'un réseau de neurones dynamique (opérant en mode ascendant-descendant) est en général beaucoup moins importante (mémoire de Hopfield, modèle d'activation interactif). Nous avons donc conclu que, durant les premiers cycles perceptifs, la reconnaissance doit être effectuée presque exclusivement en mode ascendant afin d'accumuler suffisamment d'information discriminante. Puis, en mode ascendant-descendant (interactif) afin de résoudre des ambiguïtés et de compléter des lettres manquantes (pattern completion). Nous avons optimisé notre

module de reconnaissance de mots en minimisant l'erreur d'entropie croisée afin de tenir compte de la capacité informative des lettres. Nous avons vu pourquoi notre module de reconnaissance de mots ne devait pas comporter des connexions inhibitrices. Nous avons conclu que l'adaptation de notre système à un nouveau vocabulaire pouvant être effectuée de manière relativement facile et rapide.

Nos résultats préliminaires indiquent que la performance du système est de 90%, 81%, 89% pour des mots de 3, 4, 5 lettres respectivement.

Dans le module de reconnaissance de chaînes de lettres, les lettres externes sont en général mieux reconnues que les lettres internes. Pour cette raison, les unités correspondant à la première et dernière lettres atteignent rapidement leurs niveaux de saturation. Cela veut dire que le processus de reconnaissance de notre système a lieu de l'extérieur vers l'intérieur du mot comme dans la reconnaissance humaine.

Comme la recherche de l'architecture optimale constitue l'un des inconvénients de l'utilisation des réseaux de neurones, des études ultérieures consisteraient à optimiser chacun des modules de notre système. D'autres études consisteraient à implémenter les modifications décrites dans la section précédente visant à améliorer la performance du système. Cette amélioration semble être possible à condition que les graphèmes soient identifiés avec certitude. Dans ce cas, il reste cependant à déterminer le nombre suffisant de classes de graphèmes afin d'obtenir une représentation complète du mot. La reconnaissance de chaînes de lettres devrait alors être moins complexe car l'espace de représentation doit être linéairement séparable. Ceci est dû au fait que la reconnaissance de lettres peut être effectuée pratiquement par de simples combinaisons de graphèmes, ce qui peut être effectué en utilisant de simples réseaux linéaires.

Par ailleurs, les résultats préliminaires montrent que la performance de notre système est bonne malgré le fait que nous n'avons utilisé que deux types de primitives : ascendants-descendants et boucles. Il serait cependant nécessaire de compléter cet ensemble de primitives pour améliorer la performance de notre système. Comme nous n'avons utilisé aucune information sur les fréquences des mots, la performance de notre système pourrait être

améliorée en optimisant le module de reconnaissance de mots à l'aide de méthodes optimales de séparation linéaire telles que les “support vector machines”. Il est même possible d'explorer d'autres alternatives comme l'utilisation de réseaux récurrents ou de modèles de Markov cachés pour la reconnaissance de chaînes de lettres. Par ailleurs, comme la lecture humaine comporte aussi un processus global, il serait également intéressant de fusionner notre système avec un système holistique.

Enfin, nous pouvons explorer l'utilisation d'autres types de caractéristiques résistantes au bruit et aux déconnexions du tracé comme les caractéristiques qui peuvent être extraites par la méthode que nous avons présentée dans le chapitre 4 (cavités, points caractéristiques).

Conclusions générales

Dans cette thèse, nous avons présenté une méthode de reconnaissance de caractères cursifs segmentés, une méthode holistique et une méthode perceptive de reconnaissance de mots cursifs isolés.

Nous avons étudié l'application d'une technique robuste de détection de droites au problème de l'extraction de caractéristiques. Cette technique présente une bonne tolérance au bruit et aux déconnexions du tracé. Son application à des images de mots cursifs nous permet de capturer l'information contenue dans la partie singulière du mot cursif. Afin de capturer l'information contenue dans la zone centrale du mot cursif, nous avons proposé une méthode générale qui détecte des boucles. Cette méthode nous permet également de calculer des caractéristiques plus complexes (cavités, points caractéristiques ou encore des caractéristiques directionnelles).

Nous avons testé et comparé les caractéristiques obtenues par notre méthode avec celles obtenues par d'autres méthodes. Nos résultats indiquent que les caractéristiques obtenues par notre méthode sont usuellement moins vulnérables au bruit et aux déformations que celles dérivées des contours. Nous avons vu que l'optimisation d'un réseau de neurones par la minimisation de l'erreur d'entropie croisée équivaut à la minimisation de l'information mutuelle empirique. Ce qui veut dire que nous pouvons évaluer, en même temps, la capacité informative des caractéristiques utilisées et la performance de notre système.

Nous avons passé en revue les différents modèles de lecture et les différents systèmes perceptifs de reconnaissance de l'écriture. Puis, nous avons proposé un nouveau système perceptif de reconnaissance de mots cursifs. Comme l'effet de supériorité de mot peut être

attribué en partie à la régularité de chaînes de lettres, nous avons incorporé dans notre système un module de pré-reconnaissance de lettres basé sur des réseaux de neurones à poids partagés de type TDNN. Une particularité de notre système est que les lettres externes sont mieux reconnues que les lettres internes parce que la reconnaissance est contrainte par les extrémités du mot. Pour la reconnaissance de mots, nous utilisons un module basé sur le modèle d'activation interactif. Grâce aux propriétés de ces deux modules, notre système est capable de modéliser plusieurs effets psychologiques. Par exemple, le processus de reconnaissance de notre système s'effectue de l'extérieur à l'intérieur du mot comme dans la reconnaissance humaine.

En résumé, notre système perceptif présente les caractéristiques suivantes :

- une architecture connexionniste,
- des mécanismes d'activation interactifs,
- un traitement parallèle de l'information,
- un modèle d'accès lexical,
- la prise en compte de l'effet de supériorité de mot,
- la prise en compte de l'effet de régularité de chaînes de lettres,
- la prise en compte de l'incertitude dans la position des lettres,
- la reconnaissance de l'extérieur vers l'intérieur du mot,
- une adaptation rapide et simple à un nouveau vocabulaire.

Nous avons abordé le problème de la capacité de stockage du modèle d'activation interactif. La capacité de stockage et la capacité discriminante d'un réseau statique (opérant en mode ascendant ou feed-forward) sont en général importantes. Au contraire, la capacité d'un réseau de neurones dynamique (opérant en mode ascendant-descendant) est en général beaucoup moins importante (mémoire de Hopfield, modèle d'activation interactif). Nous avons donc conclu que, durant les premiers cycles perceptifs, la reconnaissance doit être effectuée presque exclusivement en mode ascendant afin d'accumuler suffisamment d'information discriminante. Puis, en mode ascendant-descendant (interactif) afin de résoudre des ambiguïtés et de compléter des lettres manquantes (pattern completion). Nous avons opti-

misé notre module de reconnaissance de mots en minimisant l'erreur d'entropie croisée afin de tenir compte de la capacité informante des lettres. Nous avons vu pourquoi notre module de reconnaissance de mots ne doit pas comporter des connexions inhibitrices. Nous avons conclu que l'adaptation de notre système à un nouveau vocabulaire peut être effectuée de manière relativement simple et rapide.

En ce qui concerne notre méthode holistique, nous avons vu qu'elle présente une bonne performance, bien que nous n'ayons pas corrigé l'angle d'inclinaison. D'ailleurs cette méthode est plus rapide que notre méthode perceptive à cause du nombre relativement réduit de caractéristiques locales utilisées (caractéristiques directionnelles et boucles). En ce sens, nous concordons avec l'idée que la lecture rapide doit comporter au moins deux processus complémentaires opérant en parallèle et de façon interactive : un processus global rapide et un processus plus lent au niveau des lettres.

Une conclusion importante de notre étude est que le fait que le processus de lecture humain ait lieu de l'extérieur vers l'intérieur du mot puisse résulter de ce que les lettres externes sont mieux reconnues que les lettres internes.

Des études ultérieures pourraient porter sur l'optimisation de chacun des modules de notre système perceptif. Il est également possible d'explorer d'autres alternatives comme l'utilisation de réseaux de neurones récurrents ou de modèles de Markov cachés pour la reconnaissance de chaînes de lettres. Comme la lecture humaine comporte aussi un processus global, il serait aussi possible de fusionner notre système avec un système holistique. Cette fusion pourrait être effectuée facilement en ajoutant une entrée additionnelle à chaque unité-mot. D'autre part, il est possible d'explorer l'utilisation d'autres types de caractéristiques résistantes au bruit, aux déconnexions et aux déformations. Il serait par exemple intéressant de vérifier si l'optimisation de notre méthode de détection de boucles, de cavités et de points caractéristiques peut se traduire par des améliorations dans la performance de notre système. Un des problèmes des approches précédentes est d'ailleurs dû principalement à l'inefficacité des méthodes de détection de primitives. Il serait enfin intéressant d'adapter un réseau de type LeNet pour utiliser ce type de caractéristiques au lieu de l'image brute

et de tester sa performance pour la reconnaissance de caractères cursifs segmentés.

Bibliographie

- [Ali77] Ali, F. and Pavlidis, T. “Description and recognition of handwritten numerals”, in Proc. Workshop on Picture Data Processing and Management, pp. 26–32, 1977.
- [Alme90] Almeida, L.B. and Neto, J.P. “Recurrent Backpropagation and Hopfield Networks”, Neurocomputing, Vol. F68, pp. 27–36, 1990.
- [Attn54] Attneave, F. “Some informational aspects of visual perception”, Psychological Review, No. 61, pp. 183–193, 1954.
- [Avery69] Avery, G.C. and Day, R.H. “Basis of the horizontal-vertical illusion”, Journal of Experimental Psychology, Vol. 81, pp. 376–380, 1969.
- [Bair86] Baird, H.S., Kahan, S., and Pavlidis, T., “Components of an Omnifont Page Reader”, in Proc. Int. Conf. on Pattern Recognition, pp. 344–348, 1986.
- [Ball81] Ballard, D. H. , “Generalizing the Hough transform to detect arbitrary shapes”, Pattern Recognition, Vol. 13, No. 2, pp. 111–122, 1981.
- [Baro73] Baron, J. and Thurston, I. “An analysis of the word-superiority effect”, Cognitive Psychology, Vol. 17, pp. 207–228, 1973.
- [Bart96] Bartneck, N. “The role of handwriting recognition in future reading systems”, in Proc. Int. Workshop on Frontiers in Handwriting Recognition, pp. 147–176, 1996.
- [Baye92] Bayer, T., Kressel, U., and Hammelsbeck, M., “ Segmenting Merged Characters”, in Proc. Int. Conf. on Pattern Recognition, Vol. 2, pp. 346–349, 1992.

- [Beck79] Becker, C.A. "Semantic context and word frequency effects in visual word recognition", *Journal of Experimental Psychology : Human Perception and Performance*, Vol. 5, pp. 252–259, 1979.
- [Bela92] Belaïd, A. and Belaïd, Y. *Reconnaissance des Formes - Méthodes et applications*, InterEditions, Paris, 1992.
- [Belk96] Belkacem, B., " Une application industrielle de reconnaissance d'adresses", in *Actes du Colloque National sur l'Écrit et le Document (CNED'96)*, pp. 259–266, 1996.
- [Bell93] Bellegarda, E.J., Bellegarda, J.R., et al, "A Probabilistic Framework for On-line Handwriting Recognition", in *Proc. Int. Workshop on Frontiers in Handwriting Recognition (IWFHR-3)*, pp. 225–234, 1993.
- [Berc92] Bercu, S., Delyon, B. and Lorette, G. "Segmentation pour une méthode de reconnaissance d'écriture cursive «en ligne»", in *Actes du Colloque National sur l'Écrit et le Document (CNED'92)*, pp. 144–151, 1992.
- [Berc93] Bercu, S., and Lorette, G., "On-line Handwritten Word Recognition : An Approach Based on Hidden Markov Models", in *Proc. Int. Workshop on Frontiers in Handwriting Recognition (IWFHR'93)*, pp. 385–390, 1993.
- [Bert80] Berthod, M., and Ahyan, S., "On Line Cursive Script Recognition : A Structural Approach with Learning", in *Proc. Int. Conf. on Pattern Recognition*, pp. 723–725, 1980.
- [Bied85] Biederman, I. "Human Image Understanding : Recent Research and a Theory", in *Computer Vision, Graphics and Image Processing*, Vol. 32, pp. 29–73, 1985.
- [Boum68] Bouma, H. and Andriessen, J.J. "Perceived orientation of isolated line segments", *Vision Research*, Vol. 8, No. 5, pp. 493–507, 1968.
- [Boum71] Bouma, H. "Visual recognition of isolated lower-case letters", *Vision Research*, Vol. 11, No. 6, pp. 459–474, 1971.

- [Boum73] Bouma, H. "Visual interference in the parafoveal recognition of initial and final letters of words", *Vision Research*, Vol. 13, No. 4, pp. 767–782, 1973.
- [Bozi82] Bozinovic, R., and Srihari, S., "A String Correction Algorithm for Cursive Script Recognition", *Pattern Analysis and Machine Intelligence*, Vol. 4, No. 6, pp. 655–663, 1982.
- [Bozi89] Bozinovic, R., and Srihari, S., "Off-line Cursive Script Recognition", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 1, pp. 68–83, 1989.
- [Bram95] Bramall, P.E., Higgins, C.A. "A Cursive Script-Recognition System Based on Human Reading Models", *Machine Vision and Applications*, Vol. 8, No. 4, pp. 224–231, 1995.
- [Breu94] Breuel, T. M., "Design and Implementation of a System for Recognition of Handwritten Responses on US Census Forms", in *Proc. International Association for Pattern Recognition Workshop on Document Analysis Systems*, pp. 109–134, 1994.
- [Breu97] Breuel, T. M., "Handwritten character recognition using neural networks", in *Handbook of Neural Computation*, 1997.
- [Bure92] Burel, G., Pottier, I., and Catros, J.Y. "Recognition of Handwritten Digits by Image Processing and Neural Networks", in *Proc. of Int. Joint Conference on Neural Networks*, Vol. 3, pp. 666–671, 1992.
- [Burg92] Burges, C.J.C., Be, J.I., and Nohl, C.R., "Recognition of Handwritten Cursive Postal Words using Neural Networks", in *Proc. USPS Adv. Tech. Conf*, pp. 117–, 1992.
- [Case82] Casey, R.G., and Nagy, G., "Recursive segmentation and classification of composite patterns", in *Proc. Int. Conf. on Patt. Recog.*, pp. 1023–1026, 1982.
- [Case86] Casey, R.G., "Text OCR by Solving a Cryptogram", in *Proc. Int. Conf. on Pattern Recognition*, pp. 349–351, 1986.

- [Case92a] Casey, R.G., "Segmentation of Touching Characters in Postal Addresses", in Proc. USPS Advanced Technology Conf., 1992.
- [Case92b] Casey, R.G., and Takahashi, H. "Experience in segmenting and classifying the NIST data base", From Pixels To Features III : Frontiers in Handwriting Recognition, S. Impedovo et J.C. Simon (Eds.), pp. 5–16, 1992.
- [Case96] Casey, R.G. and Lecolinet, E. "A Survey of Methods and Strategies in Character Segmentation", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 18, No. 7, pp. 690–706, 1996.
- [Cast96] Castellano, G. and Sandler, M. B. "Handwritten Digits Recognition Using Hough Transform and Neural Networks", in Proc. Int. Symposium on Circuits and Systems, pp. 313–316, 1996.
- [Cesa90] Cesar, M., and Shinghal, R., "Algorithm for Segmenting Handwritten Postal Codes", Int. J. Man Machine Studies, Vol. 33, No. 1, pp. 63–80, 1990.
- [Chan92] Chan, C. K. Sandler, M. B. "A neural network shape recognition system with Hough transform input feature space", in Proc. Int. Conf. on Image Processing and its Applications, pp. 197 –200, 1992.
- [Chang91] Chang, R.L.P., Pavlidis, T. "Fuzzy Decision Tree Algorithms", Fuzzy Models for Pattern Recognition—Methods That Search for Structure in Data, IEEE Press, New York, pp. 321–325, 1991.
- [ChenMY93] Chen, M.Y., and Kundu, A., "An Alternative to Variable Duration HMM in Handwritten Word Recognition", in Proc. Int. Workshop on Frontiers in Handwriting Recognition (IWFHR-3), pp. 82–91, 1993.
- [ChenMY94] Chen, M.Y., Kundu, A. and Zhou, J. "Off-line Handwritten Word Recognition Using a Hidden Markov Model Type Stochastic Network", IEEE Trans. Patt. Anal. and Mach. Intell., Vol. 16, No. 5, pp. 481–496, 1994.

- [ChenC93] Chen, C., and DeCurtins, J., “Word Recognition in a Segmentation-Free Approach to OCR”, in Proc. Int. Conf. Document Analysis and Recognition, pp. 573–576, Oct. 1993.
- [Cheng89] Cheng, F. Hsu, W. and Chen, M. “Recognition of Handwritten Chinese Characters by Modified Hough Transform Techniques”, IEEE Trans. on Patt. Anal. and Mach. Intell., Vol. 11, No. 4, pp. 429–439, 1989.
- [Cher93] Cheriet, M., “Reading Cursive Script by Parts”, in Proc. Int. Workshop on Frontiers in Handwriting Recognition (IWFHR-3), pp. 108–110, May 1993.
- [Cher92] Cheriet, M., Huang, Y.S., and Suen, C.Y., “Background Region-Based Algorithm for the Segmentation of Connected Digits”, in Proc. Int. Conf. on Pattern Recognition, Vol. 2, pp. 619–622, Sep. 1992.
- [Chha93] Chhabra, A.K., An, Z., et al, “HighOrder Statistically Derived Combinations of Geometric Features for Handprinted Character Recognition”, in Proc. Int. Conf. on Doc. Anal. and Recog., pp. 397–401, 1993.
- [Colt93] Coltheart, M. Curtis, B., Atkins, P. and Haller, M. “Models of Reading Aloud : Dual-Route and Parallel-Distributed-Processing Approaches”, Psychological Review, Vol. 100, pp. 589–608, 1993.
- [Colt94] Coltheart, M. and Rastle, K. “Serial processing in reading aloud : Evidence for dual-route models of reading”, Journal of Experimental Psychology : Human perception and Performance, Vol. 20, pp. 1197–1211, 1994.
- [Côté97] Côté, M. Utilisation d’un modèle d’accès lexical et de concepts perceptifs pour la reconnaissance d’images de mots cursifs, Thèse de Doctorat de l’Ecole Nationale Supérieure des Télécommunications de Paris, Paris, France, 1997.
- [Côté98] Côté, M., Lecolinet, E. et al. “Automatic Reading of Cursive Scripts Using a Reading Model and Perceptual Concepts : the PERCEPTO system”, International Journal of Document Analysis and Recognition, Vol. 1, No. 1, pp. 3–17, 1998.

- [Davi86] Davis, R.H. and Lyall, J. "Recognition of Handwritten Characters - A Review", *Image and Vision Computing*, Vol. 4, No. 4, pp. 208–218, 1986.
- [Davi86b] Davies, E.R. "Image Space Transforms for Detecting Straight Edges in Industrial Images", *Pattern Recognition Letters*, Vol. 4, No. 3, pp. 185–192, 1986.
- [Dean81] Deans, S.R. "Hough Transform from the Radon Transform", *IEEE Trans. on Patt. Anal. and Mach. Intell.*, Vol. PAMI-3, No. 2, pp. 185–188, 1981.
- [Dijk92] Dijkstra, A. and Van Heuven, W.J.B. "The BIA-model and bilingual word recognition", *Localist Connectionist Approaches to Human Cognition*, London, Lawrence Erlbaum Associates, 1992.
- [Dima97] Dimauro, G., Impedovo, S., Pirlo, G. and Salzo, A. "A Multi-Expert Signature Verification System for Bankcheck Processing", *Int. J. on Pattern Recognition and Artificial Intelligence*, Vol. 11, No. 5, pp. 827–844, 1997.
- [Dooi91] Dooijes, E.H. and Hamstra-Bletz, E. "A Topological Approach to Handwriting Understanding", in *Proc. Int. Conf. on Doc. Anal. and Recog. (ICDAR'91)*, Vol. 2, pp. 594–602, 1991.
- [Duda75] Duda, R.O. and Hart, P.E. "Use of the Hough transform to detect lines and curves in pictures", *Communications of the Association of Computing Machinery*, Vol. 15, pp. 11–15, 1975.
- [Dune96] Duneau, L., Dorizzi, B., "Online Cursive Script Recognition : A User-Adaptive System for Word Identification", *Pattern Recognition*, Vol. 29, No. 12, , pp. 1981–1994, Dec. 1996.
- [Dunn92] Dunn, C.E., and Wang, P.S.P., "Character Segmenting Techniques for Handwritten Text - A Survey", in *Proc. Int. Conf. on Pattern Recognition*, Vol. 2, pp. 577–580, Aug. 1992.
- [Earn62] Earnest, L.D., "Machine Recognition of Cursive Writing", *Information Processing*, pp. 462–466, London, 1962.

- [Edel97] Edelman, S. "Computational theories of object recognition", *Trends in Cognitive Sciences*, Vol. 1, No. 8, pp. 296–304, 1997.
- [Ehri75] Ehrich, R.W., and Koehler, K.J., "Experiments in the Contextual Recognition of Cursive Script", *IEEE Trans. Computers*, Vol. 24, No. 2, pp. 182–194, Feb. 1975.
- [Elli92] Elliman, D.G. and Connor, P.J. "Orientation and scale invariant symbol recognition using a hidden Markov model", *IEE Int. Conf. on Image Processing and its Applications*, pp. 331–334, 1992.
- [Fara79] Farag, R.F.H. "Word Level Recognition of Cursive Script", *IEEE Trans. on Computers*, Vol. c-28, No. 2, pp. 172– 175, 1979.
- [Fava90] Favata, J.T., and S.N. Srihari, "Recognition of Handwritten Words for Address Reading", in *Proc. of USPS Advanced Technology Conference*, pp. 191–205, 1990.
- [Fava92] Favata, J.T., and Srihari, S.N., "Recognition of General Handwritten Words Using a Hypothesis Generation and Reduction Methodology", in *Proc. USPS Advanced Technology Conf.*, pp 237–245, 1992.
- [Fava96] Favata, J.T., Srikantan, G. "A Multiple Feature/Resolution Approach to Handprinted Digit and Character-Recognition", *Int. Journal of Imaging Systems and Technology*, Vol. 7, No. 4, pp. 304–311, 1996.
- [Fenr92] Fenrich, R., "Segmenting of Automatically Located Handwritten Numeric Strings", in *From Pixels to Features III : Frontiers in Handwriting Recognition*, pages 47 –59. Elsevier Science, 1992.
- [Fors76] Forster, K.I. "Accessing the Mental Lexicon", in *New approaches to language mechanisms*, E. C. J. Walker and R. J. Wales (Eds.), 1976.
- [Fors81] Forster, K.I. "Priming and the effects of sentence and lexical context on naming time : Evidence for autonomous lexical processing", *Journal of Experimental Psychology*, Vol 33A, pp. 465–495, 1981.

- [Frid89] Friday, P.D., Leedham, C.G., "A Pre-Segmenter for Separating Characters in Unconstrained Hand-Printed Text", in Proc. Int. Conf. on Image Processing, pp. 299–303, 1989.
- [Fuku82] Fukushima, K., and Miyake, S. "Neocognitron : a new algorithm for pattern recognition tolerant of deformations and shifts in position", Pattern Recognition, Vol. 15, pp. 455–469, 1982.
- [Fuku93] Fukushima, K., and Imagawa, T., "Recognition and Segmentation of Connected Characters with Selective Attention", Neural Networks, Vol. 6, No. 1, pp. 33–41, 1993.
- [Fuji92] Fujisawa, H., Nakano, Y., and Kurino, K., "Segmentation Methods for Character Recognition : From Segmentation to Document Structure Analysis", in Proc. IEEE, Vol. 80, No. 7, pp. 1079–1092, 1992.
- [Gade96] Gader, P. D. and Khabou, M. A. "Automatic Feature Generation for Handwritten Digit Recognition", IEEE Trans. on Patt. Anal. and Mach. Intell., Vol. 18, No. 12, pp. 1256–1261, 1996.
- [Gade94] Gader, P., Gillies, A. and Hepp, D. "Handwritten Character Recognition", in Digital Image Processing Methods, Edward Dougherty (ed.), pp. 223– 261, 1994.
- [Gade92] Gader, P., Magdi, M., and Chiang, J-H., "Segmentation-Based Handwritten Word Recognition", in Proc. USPS Advanced Technology Conf., 1992.
- [Gail79] Gaillat, G., and Berthod, M. "Panorama des techniques d'extraction de traits caractéristiques en lecture optique des caractères", Revue Technique Thomson-CSF, Vol. 11, No. 4, pp. 943–959, 1979.
- [Gamb75] Gambardella, G. "Representation of the spatial-frequency analysis performed by the visual system", Journal of the Optical Society of America, Vol. 65, No. 1, pp. 99–100, 1975.

- [Garr91] Garris, M.D., Wilkinson, R.A., and Wilson, C.L. "Methods for Enhancing Neural Network Handwritten Character Recognition", in Proc. of International Joint Conference on Neural Networks, Vol. 1, pp. 695–700, 1991.
- [Gilli92] Gillies, A.M., "Cursive Word Recognition Using Hidden Markov Models", in Proc. USPS Advanced Technology Conf., pp. 557–562, 1992.
- [Gillo94] Gilloux, M. "Hidden Markov Models in handwriting recognition", in Fundamentals in Handwriting Recognition, NATO ASI, Series F : Computer & Systems Sciences, Springer Verlag, pp. 264–287, 1994.
- [Gillo93] Gilloux, M., and Bertille, J.M. "Recognition of Handwritten Words in a Limited Dynamic Vocabulary", in Proc. Int. Workshop on Frontiers in Handwriting Recognition, pp. 417–422, 1993.
- [Gluc67] Glucksman, B. A. "Classification of mixed-font alphabets by characteristic loci", in Proc. IEEE Comp. Conf., pp. 138–141, 1967.
- [Gluc71] Glucksman, H. A. "Multicategory classification of patterns represented by high order vectors of multilevel measurements", IEEE Trans. on Computers, Vol. 20, pp. 1593–1598, 1971.
- [Gors93] Gorski, N.D. "Off-line recognition of bad quality handwritten words using prototypes", Fundamentals in Handwriting Recognition, S. Impedovo, Nato Asi Series, Series F : Computer and Systems Sciences, Vol. 124, pp. 199–217, 1993.
- [Gors98] Gorski, N., Anisimov, V., et al. "A New A2iA Bankcheck Recognition System", in Proc. IEE Workshop on Handwriting Analysis and Recognition, pp. 24/1–24/6, 1998.
- [Gors99] Gorski, N., Anisimov, V., et al. "A2iA Check Reader : A Family of Bank Check Recognition Systems", in Proc. Int. Conf. on Doc. Anal. and Recog., pp. 523–526, 1999.

- [Govi90] Govindan, V.K., Shivaprasad, A.P. "Character Recognition : A Review", *Pattern Recognition*, Vol. 23, pp. 671–683, 1990.
- [Grot93] Grother, P. J., and Candela, G.T. "Comparison of Handprinted Digit Classifiers", Report NISTIR 5209, National Institute of Standards and Technology, 1993.
- [Guil94] Guillevic and Suen, C.Y. "Cursive Script Recognition : A Sentence Level Recognition Scheme", in *Proc. Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pp. 216–223, 1994.
- [Guil95b] Guillevic and Suen, C.Y. "Cursive Script Recognition applied to the Processing of Bank Cheques", in *Proc. Int. Conf. on Document Analysis and Recognition (ICDAR)*, pp. 11–14, 1995.
- [Guil95a] Guillevic, D., *Unconstrained Handwriting Recognition Applied to the Processing of Bank Cheques*, PhD thesis, Computer Science Department, Concordia University, Montreal, 1995.
- [Guil97] Guillevic and Suen, C.Y. "HMM Word Recognition Engine", in *Proc. Int. Conf. on Document Analysis and Recognition (ICDAR'97)*, Vol. 2, pp. 544–547, 1997.
- [Hanc92] Hancock, P.J.B., Baddeley, R.J. and Smith, L.S. "The principal components of natural images", *Network*, Vol. 3, pp. 61–70, 1992.
- [Harm98] Harm, M.W. *Division of Labor in a Computational Model of Visual Word Recognition*, PhD Thesis, University of Southern California, 1998.
- [Harmo72] Harmon, L.D., "Automatic Recognition of Print and Script", in *Proc. IEEE*, Vol. 60, No. 10, pp. 1165–1177, 1972.
- [Haye80] Hayes, K.C., "Reading Handwritten Words Using Hierarchical Relaxation", *Computer Graphics and Image Processing*, Vol. 14, pp. 344–364, 1980.
- [Henn68] Hennis, R.B., "The IBM 1975 Optical Page Reader : System Design", *IBM J. Research and Development*, pp. 346–353, 1968.

- [Hert91] Hertz, J., Krogh, A. and Palmer, R.G. Introduction to the theory of neural computation, Lectures Notes of the Santa Fe Institute of Studies in the Sciences of Complexities, AddisonWesley, 1991.
- [Hetc98] Hetch, H. and Bader, H. "Perceiving topological structure of 2D patterns", *Acta Psychologica*, Vol. 99, pp. 255–292, 1998.
- [Heut94] Heutte, L., Reconnaissance de caractères manuscrits : Application à la lecture automatique des chèques et des enveloppes postales, Thèse de Doctorat de l'Université de Rouen, 1994.
- [Heut94] Heutte, L., Lecourtier, Y. and Moreau, J.V. "Une Nouvelle Méthode d'extraction de Projections et de Profils pour la Reconnaissance de Chiffres Manuscrits", in *Actes du Colloque National sur l'Écrit et le Document (CNED'94)*, pp. 1–10, 1994.
- [Higg85] Higgins, C.A., and Whitrow, R., "On-line Cursive Script Recognition", in *Proc. Int. Conf. Human Computer Interaction*, pp. 139–143, 1985.
- [Higg91] Higgins, C.A., and Bramall, P.E., "Handwriting recognition based on the human reading process", in *Proc. Int. Conf. on Doc. Anal. and Recog. (ICDAR'91)*, Vol. 1, No. 1, pp. 155–167, 1991.
- [Hint95] Hinton, G.E., Revow, M., and Dayan, P. "Recognizing Handwritten Digits Using Mixtures of Linear Models", *Advances in Neural Information Processing Systems*, Vol. 7, pp. 1015–1022, Cambridge MA, 1995.
- [Ho92] Ho. T. K. A Theory of Multiple Classifier Systems and its Application to Visual Word Recognition, PhD Thesis, State University of New York at Buffalo, 1992.
- [Hoek96] Hoekstra, A., Kraaijveld, M.A., de Ridder, D. and Schmidt, W.F. The Complete SPRLIB & ANNLIB : Statistical Pattern Recognition and Artificial Neural Network Library, User's Guide and Reference Manual, Pattern Reco-

- gnition Group, Faculty of Applied Physics, Delft University of Technology, The Netherlands, 1996.
- [Holt92] Holt, M., Beglou, M., and Datta, S., "Slant-Independent Letter Segmentation for Off-line Cursive Script Recognition", in *From Pixels to Features III : Frontiers in Handwriting Recognition*, S. Impedovo and J. C. Simon (eds.), Elsevier Science, pp. 41–46, 1992.
- [Hu96] Hu, J., Yu, D. and Yan, H. "Algorithm for stroke width compensation of handwritten characters", *Electronics Letters*, Vol. 32, No. 24, pp. 2221–2222, 1996.
- [Huan93] Huang, Y. S. and Suen, C. Y. "Combination of Multiple Classifiers with Measurement Values", in *Proc. Int. Conf. on Doc. Analysis and Recog.*, pp. 598–601, 1993.
- [Huan94] Huang, Y. S. and Suen, C. Y. "A Method for Combining Multiple Classifiers - A Neural Network Approach", *Proc. Int. Conf. on Patt. Recog.*, pp. 473–475, 1994.
- [Huan95] Huang, Y. S. and Suen, C. Y. "A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals", *IEEE Trans. on Patt. Anal. and Mach. Intell.*, Vol. 17, No. 1, pp. 90–94, 1995.
- [Hube68] Hubel, D. H. and Wiesel, T. N. "Receptive fields and functional architecture of monkey striate cortex", *Journal of Physiology*, Vol. 195, pp. 215–43, 1968.
- [Hube79] Hubel, D.H. and Wiesel, T.N. "Brain mechanisms of vision," *Scientific American*, vol. 241, pp. 150–162, 1979.
- [Hull92] Hull, J.J., Khoubyari, S., and Ho, T.K. "Word Image Matching as a Technique for Degraded Text Recognition", in *Proc. Int. Conf. on Pattern Recognition*, pp. 665–668, 1992.

- [Hull82] Hull, J.J., Srihari, S.N., "Experiments in Text Recognition with Binary N-Gram and Viterbi Algorithms", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 4, No. 5, pp. 520–530, 1982.
- [Hull86] Hull, J.J., Srihari, S.N., "A Computational Approach to Visual Word Recognition : Hypothesis Generation and Testing", in *Proc. Computer Vision and Pattern Recognition*, pp. 156–161, 1986.
- [Hwan97] Hwang, Y., Bang, S. "Recognition of unconstrained handwritten numerals by radial basis function network classifier", *Pattern Recognition Letters*, vol. 18, no. 7, pp. 657–664, 1997.
- [Idan92] Idan, Y., Auger, J., et al. "La reconnaissance de caractères manuscrits par méthodes neuronales dans le projet Neuroscript", in *Actes du Colloque National sur l'Écrit et le Document (CNED'92)*, pp. 168–177, 1992.
- [Ill88] Illingworth, J. and Kittler, J. "A Survey of the Hough Transform", *Computer Vision Graphics and Image Processing*, No. 44, pp. 87–116, 1988.
- [Imme98] Immerkær, J. "Some remarks on the straight line Hough transform", *Pattern Recognition Letters*, Vol. 19, pp. 1133–1135, 1998.
- [Itoh90] Itoh, N. and Takahashi, H. "A Hand-written Numeral Verification Method Using Distribution Maps of Structural Features", *Proc. SPIE Int. Conf. on Image Comm. and Workstations*, vol. 1258, 1990.
- [Jaco94] Jacobs, A.M. and Grainger, J. "Models of visual word recognition - sampling the state of the art", *Journal of Experimental Psychology : Human Perception and Performance ; Special Section : Modeling Visual Word Recognition*, Vol. 20, No. 6, pp. 1311–1334, 1994.
- [Jaco94] Jacobs, R.A. and Kosslyn, S.M. "Encoding Shape and Spatial Relations : The Role of Receptive Field Size in Coordinating Complementary Representations", *Cognitive Science*, Vol. 18, pp. 361–386, 1994.

- [Jule83] Julesz, B. and Bergen, J.R. "Textons, The Fundamental Elements in Preattentive Vision and Perception of Textures", *Bell System Technical Journal*, Vol. 62, No. 6, pp. 1619–1645, 1983.
- [Jung96] Jung, D., Krishnamoorthy, M., et al. "N-Tuple Features for OCR revisited", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 7, pp. 734–745, 1996.
- [Kaha87] Kahan, S., Pavlidis, T. and Baird, H. S. "On the recognition of printed characters of any font and size", *IEEE Trans. on Patt. Anal. and Mach. Intell.*, vol. PAMI-9, pp. 274–288, 1987.
- [Kalv95] Kälviäinen, H., Hirvonen, P., Xu, P. and Oja, E. "Probabilistic and non-probabilistic Hough transforms : overview and comparisons", *Image and Vision Computing*, Vol. 13, No. 4, pp. 239–252, 1995.
- [Kane89] Kanerva, P. "Contour-Map Encoding of Shape for Early Vision", *Advances in Neural Information Processing Systems*, Vol. 2, pp. 282–289, 1989.
- [Kim00] Kim, J.H., Kim, K.K. et al. "A Methodology of Combining HMM and MLP Classifiers for Cursive Word Recognition", in *Proc. Int. Conf. on Pattern Recognition*, pp. 319–322, 2000.
- [Kimu97] Kimura, F., Kayahara, N. et al "Machine and Human Recognition of Segmented Characters from Handwritten Words", in *Proc. Int. Conf. on Doc. Anal. and Recog.*, pp. 866–869, 1997.
- [Kimu93] Kimura, F., Shridhar, M., and Narasimhamurthi, N., "Lexicon Directed Segmentation-Recognition Procedure for Unconstrained Handwritten Words", in *Proc. Int. Workshop on Frontiers in Handwriting Recognition (IWFHR-3)*, pp. 122–131, 1993.
- [Kimu92] Kimura, F., Tsuruoka, S., et al., "Context Directed Handwritten Word Recognition for Postal Service Applications", in *Proc. USPS Advanced Technology Conf.*, 1992.

- [Kner98a] Knerr, S. Augustin, E. et al, "Hidden Markov model based word recognition and its application to legal amount reading on French checks", *Computer Vision and Image Understanding*, Vol. 70, No. 3, pp. 404–419, 1998.
- [Kner98b] Knerr, S. and Augustin, E. "A neural network-hidden Markov model hybrid for cursive word recognition", in *Proc. Int. Conf. on Pattern Recognition*, Vol. 2, pp. 1518–1520, 1998.
- [Komo77] Komori, K., Kawatani, T., Ishii, K. and Iida, Y. "A feature concentration method for character recognition, in *Proc. Int. Fed. Information Processing Soc. Congr.*, pp. 29–34, 1977.
- [Kova68] Kovalevsky, V.A., *Character Readers and Pattern Recognition*, Spartan Books, Washington, 1968.
- [Kova80] Kovalevsky, V.A., *Image Pattern Recognition*, Vol. 8, Springer Verlag, 1980.
- [Kris87] Krishnapuram, R. and Casasent, D. "Hough Space Transformations for Discrimination and Distortion Estimation", *Computer Vision, Graphics and Image Processing*, Vol. 38, pp. 299–316, 1987.
- [Krzy91] Krzyzak, A., Dai, W., Suen, C.Y. "On the Recognition of Handwritten Characters Using Neural Networks", *Pattern Recognition—Architectures, Algorithms & Applications*, World Scientific, Singapore, pp. 115–135, 1991.
- [Kund89] Kundu, A., He, Y., and Bahl, P., "Recognition of Handwritten Words : First and Second Order Hidden Markov Model Based Approach", *Pattern Recognition*, Vol. 22, No. 3, pp. 283–297, 1989.
- [Kuri98] Kuriki, S., Takeuchi, F. and Hirata, Y. "Neural processing of words in the human extrastriate visual cortex", *Cognitive Brain Research*, Vol. 6, pp. 193–203, 1998.
- [Kush85] Kushnir, M., Abe, K., and Matsumoto, K. "Recognition Of Handprinted Hebrew Characters Using Features Selected In The Hough Transform Space", *Pattern Recognition*, Vol. 18, pp. 103–114, 1985.

- [Lall97] Lallican, P.M. and Viard-Gaudin, C. "A Kalman Approach for Stroke Order Recovering from Off-Line Handwriting", in Proc. Int. Conf. on Document Analysis and Recognition, Vol. 2, pp. 519–522, 1997.
- [Lam95] Lam, L., Suen, C.Y. et al. "Automatic Processing of Information on Cheques", in Proc. Int. Conf. on Systems, Man & Cybernetics, Vancouver, Canada, pp. 2353–2358, 1995.
- [Lam93] Lam, S.W., Javanbakht, L. and Srihari, S.N. "Anatomy of a form reader", in Proc. Int. Conf. on Doc. Anal. and Recog. (ICDAR'93), pp. 506–509, 1993.
- [Lamy95] Lamy, M.B., Reconnaissance de caractères manuscrits par combinaison de modèles connexionnistes, Thèse de Doctorat de l'Université Paris VI, 1995.
- [Leco89] Lecolinet, E., Moreau, J.V. "A New System for Automatic Segmentation and Recognition of Unconstrained Zip Codes", in Proc. Scandinavian Conf. Image Analysis, Vol. 1, pp. 585–592, 1989.
- [Leco90] Lecolinet, E., "Segmentation d'images de mots manuscrits : application à la lecture de chaînes de caractères majuscules alphanumériques et à la lecture de l'écriture cursive", Ph. D. Thesis, Université Pierre et Marie Curie (Paris VI), Paris, France, 1990.
- [Leco91] Lecolinet, E. and Crettez, J. "A Grapheme-based Segmentation Technique for Cursive Script Recognition", in Proc. Int. Conf. on Doc. Anal. and Recog. (ICDAR'91), pp. 740–748, 1991.
- [Leco93] Lecolinet, E., "A New Model for Context-Driven Word Recognition", in Proc. IEEE Symp. Document Analysis and Information Retrieval, Las Vegas, pp. 135–145, 1993.
- [Leco94a] Lecolinet, E. and Baret, O. "Cursive Word Recognition : Methods and Strategies", Fundamentals in Handwriting Recognition, S. Impedovo, NATO ASI Series F : Computer and System Sciences, Vol. 124, pp. 235–263, 1994.

- [Leco94b] Lecolinet, E. "Cursive script recognition by backward matching", in *Advances in Handwriting and Drawing : A Multidisciplinary Approach*, pp. 117–135, 1994.
- [LeCu90] LeCun, Y., Boser, B., et al. "Handwritten Digit Recognition with a Back-Propagation Network", *Advances in Neural Information Processing Systems*, vol. 2, pp. 396–404, 1990.
- [LeeCK95] Lee, D. S. and Srihari, S. N. "A Theory of Classifier Combination : The Neural Network Approach", in *Proc. Int. Conf. on Doc. Anal. and Recog. (ICDAR'95)*, Vol. 1, pp. 42–45, 1995.
- [LeeCK99] Lee, C.K. and Leedham, G. "Empirical design of a holistic verifier for automatic sorting of handwritten Singapore postal addresses", in *Proc. Int. Conf. on Document Analysis and Recognition (ICDAR'99)*, pp. 733 –736, 1999.
- [LeeS95] Lee, S. and Lee, E. "Integrated Segmentation and Recognition using a Recurrent Neural Network", in *Proc. Int. Conf. on Doc. Anal. and Recog. (ICDAR'95)*, pp. 413–416, 1995.
- [LeeY91] Lee, Y. "Handwritten digit recognition using k nearest-neighbor radial-basis function, and backpropagation neural networks", *Neural Computation*, vol. 3, pp. 440–449, 1991.
- [Legge97] Legge, G.E., Klitz, T.S. and Tjan, B.S. "Mr. Chips : An Ideal-Observer Model of Reading", *Psychological Review*, Vol. 104, No. 3, pp. 524–553, 1997.
- [Lema93] Lemarié, B. "Practical Implementation of a Radial Basis Function Network for Handwritten Digit Recognition", in *Proc. Int. Conf. on Doc. Anal. and Recog. (ICDAR'93)*, pp. 412–415, 1993.
- [Lero91] Leroux, M., Salome, J.C., and Badard, J. "Recognition of Cursive Script Words in a Small Lexicon", in *Proc. Int. Conf. Document Analysis and Recognition*, pp. 774–782, 1991.

- [Lero97] Leroy, A. Reconnaissance d'écriture manuscrite dynamique par approche descendante - caractérisation du style de l'écriture et application, PhD Thesis, Université de Rennes, 1997.
- [Lian93] Liang, S., Ahmadi, M., and Shridhar, M., "Segmentation of Touching Characters in Printed Document Recognition", in Proc. Int. Conf. Document Analysis and Recognition, Tsukuba City, Japan, pp. 569–572, 1993.
- [Lind77] Lindsay, P.H. and Norman, D.A. Human information processing : An introduction to psychology, New York : Academic Press, 1977.
- [Liu00] Liu, L. and Arditi, A. "Apparent string shortening concomitant with letter crowding", Vision Research, Vol. 40, pp. 1059–1067, 2000.
- [Lo95] Lo, R.C. and Tsai, W.H. "Gray-scale Hough transform for thick line detection in gray-scale images", Pattern Recognition, Vol. 38, No. 5, pp. 647–661, 1995.
- [Lu93] Lu, Y., "On the Segmentation of Touching Characters", Int. Conf. on Document Analysis and Recognition, Tsukuba, Japan, pp. 440–443, 1993.
- [Lugt64] Lugt, A.V. "Signal Detection by Complex Spatial Filtering", IEEE Trans. on Information Theory, Vol. IT-10, pp. 139–145, 1964.
- [Madh93] Madhvanath, S., and Govindaraju, V., "Holistic Lexicon Reduction", in Proc. Int. Workshop on Frontiers in Handwriting Recognition, Buffalo, NY, pp. 132–141, 1993.
- [Maie86] Maier, M., "Separating Characters in Scripted Documents", in Proc. Int. Conf. on Pattern Recognition, Paris, pp. 1056–1058, 1986.
- [Mand92] Mandalia, A. D. Pandya, A. S. and Sudhakar, R. "A Hybrid Approach to Recognize Handwritten Alphanumeric Characters", in Proc. IEEE Int. Conf. on Systems Man and Cybernetics, Vol. 1, pp. 723–726, 1992.
- [Mant86] Mantas, J. "An Overview of Character Recognition Methodologies", Pattern Recognition, Vol. 19, No. 6, pp. 425–430, 1986.

- [Mant87] Mantas, J. "Methodologies in Pattern Recognition and Image Analysis : A Brief Survey", *Pattern Recognition*, Vol. 20, No. 1, pp. 1–6, 1987.
- [Marr82] Marr, D. *Vision*, W.H. Freeman and Co., San Francisco, 1982.
- [Marr80] Marr, D. and Hildreth, E. "Theory of edge detection", in *Proc. Royal Soc. London, Series B*, No. 207, pp. 187–217, 1980.
- [Mart96] Martin, G. L., "The impact of letter classification learning on reading", in *Proc. Conf. of the Cognitive Science Society*, 1996.
- [Masin83] Masin, S.C. and Vidotto, G. "A magnitude estimation study of the inverted-T illusion", *Perception and Psychophysics*, Vol. 33, pp. 582–584, 1983.
- [Mats85] Matsumoto, K. "Recognition of Handprinted Hebrew Characters Using Features Selected in the Hough Transform Space", *Pattern Recognition*, Vol. 18, pp. 103–114, 1985.
- [McCl85] McClelland, J.L. "Putting knowledge in its place : a scheme for programming parallel processing structures on the fly", *Cognitive Science*, No. 9, pp. 113–146, 1985.
- [McCl86] McClelland, J.L. "The programmable blackboard model of reading", in *Parallel Distributed Processing : Explorations in the microstructure of cognition*, Vol. II : Psychological and biological models, MIT Press/Bradford Books, pp. 122–169, 1986.
- [McCl81] McClelland, J.L. and Rumelhart, D.E. " An interactive activation model of context effects in letter perception : part 1", *Psychological Review*, Vol. 88, pp. 375–407, 1981.
- [Méni92] Ménier, G. and Lorette, G. "Segmentation et reconnaissance en ligne d'écriture cursive à l'aide de plusieurs niveaux d'information contextuelle", in *Actes du Colloque National sur l'Écrit et le Document (CNED'92)*, Nancy, pp. 318–324, 1992.

- [Milg90] Milgram, M., Jobert, M., and Lamy, B. "A Segmentation Free Approach to Symbol Extraction and Recognition From Image Document", in Proc. IAPR Workshop on Machine Vision Applications (MVA'90), Tokyo, 1990.
- [Mitt89] Mitchell, B.T. and Gillies, A.M. "A Model-Based Computer Vision System for Recognizing Handwritten Zip Codes", Machine Vision and Applications, no. 2, pp. 231–243, 1989.
- [Mohi94] Mohiuddin, K.M. and Mao, J. "A comparative study of different classifiers for handprinted character recognition", Pattern Recognition in Practice IV, pp. 437–449, 1994.
- [Mont95] Montoliu, L., Architecture multi-agents et réseaux connexionnistes - Application à la lecture de chèques manuscrits, Thèse de Doctorat de l'Ecole Polytechnique, 1995.
- [Mood89] Moody, J. and Darken, C.J. "Fast learning in networks of locally-tuned processing units", Neural Computation, Vol. 1, pp. 281–294, 1989.
- [Mora93] Morasso, P., Gismondi, L., et al. "Incremental learning in on-line recognition of cursive handwriting", in Proc. ICOHD'93, pp. 213–215, 1993.
- [More91] Moreau, J.V., Plessis, B., Bougeois, O. and Plagnaud, J.L. "A Postal Check Reading System", in Proc. Int. Conf. on Document Analysis and Recognition, pp. 758–766, 1991.
- [Mori92] Mori, S., Suen, C.Y., Yamamoto, K. "Historical Review Of Ocr Research And Development", in Proc. IEEE, Vol. 80, No. 7, pp. 1029–1058, 1992.
- [Mort69] Morton, J. "Interaction of information in word recognition", Psychological Review, Vol. 76, pp. 165–178, 1969.
- [Moze87] Mozer, M.C., "Early Parallel Processing in Reading : A Connectionist Approach", in Attention and Performance, M. Coltheart, 1987.
- [Murp86] Murphy, L.M. "Linear feature detection and enhancement in noisy images via the Radon transform", Pattern Recognition Letters, No. 4, pp. 279–284, 1986.

- [Nag86] Nag, R., Wong, K.H., and Fallside, F. "Script Recognition Using Hidden Markov Models", in Proc. Int. Conf. on Acoustics Speech and Signal Processing, pp. 2071–2074, 1986.
- [Ney95] Ney, H. "On the Probabilistic Interpretation of Neural Network Classifiers and Discriminative Criteria", IEEE Trans. on Patt. Anal. and Mach. Intell., Vol. 17, No. 2, pp. 107–119, 1995.
- [Ohta93] Ohta, K., Kaneko, I., et al. "Character Segmentation of Address Reading/Letter Sorting for the Ministry of Posts and Telecommunications of Japan", NEC Research and Development, Vol. 34, No. 2, pp. 248–256, 1993.
- [O'Re90] O'Regan, J.K. "Eye movements and reading", in Eye Movements and their role in visual and cognitive processes, Elsevier Science, New York, pp. 395–453, 1990.
- [Oula89] Ouladj, H., Lorette, G., et al, ">From Primitives to Letters : A structural Method to Automatic Cursive Handwriting Recognition", in Proc. Scandinavian Conf. on Image Analysis, Finland, pp. 593–598, 1989.
- [Paqu91] Paquet, T., Lecourtier, Y. "Handwriting Recognition : Application on Bank Cheques", in Proc. Int. Conf. on Document Analysis and Recognition, pp. 749–757, 1991.
- [Paris96] Parisse, C, "Global Word Shape Processing in off-line Recognition of Handwriting", IEEE Trans. on Patt. Anal. and Mach. Intell., Vol. 18, No. 4, pp. 460 –464, 1996.
- [Pariz95] Parizeau, M. and Plamondon, R., "Fuzzy-Syntactic Approach to Allograph Modelling for Cursive Script Recognition", IEEE Trans. on Patt. Anal. and Mach. Intell., Vol. 17, No. 7, pp. 702 –712, 1995.
- [Park91] Park, J. and Sandberg, I. W. "Universal Approximation Using Radial-Basis-Function Networks", in Neural Computation, Vol. 3, No. 2, pp. 246–257, 1991.

- [Paru82] Parui, S.K., Chaudhuri, B.B., and Majunder, D.D., "A Procedure for Recognition of Connected Handwritten Numerals", *Int. J. Systems Science*, Vol. 13, No. 9, pp. 1019–1029, 1982.
- [Pizlo97] Pizlo, Z., Salach-Golyska, M. and Rosenfeld, A. "Curve detection in a noisy image", *Vision Research*, Vol. 37, No. 9, pp. 1217–1241, 1997.
- [Ples93] Plessis, B., Sicsu, A., et al, "A Multi-Classifer Strategy for the Recognition of Handwritten Cursive Words", in *Proc. Int. Conf. on Document Analysis and Recognition*, pp. 642–645, Tsukuba, Japon, 1993.
- [Pommier93] Pommier, G. *Naissance et renaissance de l'écriture*, PUF, Paris, 1993.
- [Popple00] Popple, A.V. and Levi, D.M. "A new illusion demonstrates long-range processing", *Vision Research*, Vol. 40, pp. 2545–2549, 2000.
- [Powa96] Powalka, R.K.; Sherkat, N. and Whitrow, R.J. "Multiple recognizer combination topologies", to be published in the book of selected papers from the Seventh IGS conference, M. Simner, G. Leedham, A. Thomassen (Eds.), IOS Press, 1996.
- [Rayn96] Rayner, K.; Sereno, S. and Raney, G.E. "Eye Movement Control in Reading : A Comparison of Two Types of Models", *Journal of Experimental Psychology : Human Perception and Performance*, Vol. 22, No. 5, pp. 1188–1200, 1996.
- [Reic69] Reicher, G.M. "Perceptual recognition as a function of meaningfulness of stimulus material", *Journal of Experimental Psychology*, No. 81, pp. 275–280, 1969.
- [Revo96] Revow, M.; Williams, C.K.I.; Hinton, G.E. "Using generative models for handwritten digit recognition", *IEEE Trans. on Patt. Anal. and Mach. Intell.*, Vol. 18, No. 6, pp. 592 –606, 1996.
- [Rey00] Rey, A., Ziegler, J.C. and Jacobs, A.M. "Graphemes are perceptual reading units", *Cognition*, Vol. 75, B1–B12, 2000.

- [Ridd96] De Ridder, D., Hoekstra, A. and Duin, R.P.W., "Feature extraction in shared weights neural networks", in Proc. Annual Conference of the Advanced School for Computing and Imaging, Delft, Jun. 1996.
- [Roch93] Rocha, J., and Pavlidis, T., "New Method for Word Recognition Without Segmentation", in Proc. SPIE Character Recognition Technologies, Vol. 1906, pp. 74–80, 1993.
- [Rume74] Rumelhart, D. E. and Siple, P. "The process of recognizing tachistoscopically presented words", *Psychological Review*, Vol. 81, pp. 99–118, 1974.
- [Rume82] Rumelhart, D.E. and McClelland, J.L., " An interactive activation model of context effects in letter perception : part 2", *Psychological Review*, Vol. 89, No. 1, pp. 60–94, 1982.
- [Sayr73] Sayre, K.M., "Machine Recognition of Handwritten Words : A Project Report", *Pattern Recognition*, Vol. 5, No. 3, pp. 213–228, 1973.
- [Sche95] Schenkel, M., Guyon, I. and Henserson, D., "On-line Cursive Script Recognition using Time Delay Neural Networks and Hidden Markov Models", *Machine Vision and Applications*, Vol. 8, pp. 215–223, 1995.
- [Scho93] Schomaker, L., "Using stroke- or character-based self-organizing maps in the recognition of on-line, connected cursive script", *Pattern Recognition*, Vol. 26, No. 3, pp. 443–450, 1993.
- [Schw95] Schwenk, H., and Milgram, M., "Transformation Invariant Autoassociation with Application to Handwritten Character Recognition", *Advances in Neural Information Processing Systems*, Vol. 7, Cambridge MA, 1995.
- [Seid89] Seidenberg, M.S., & McClelland, J. L., "A distributed model of word recognition and naming", *Psychological Review*, Vol. 96, No. 4, pp. 523–568, 1989.
- [Self59] Selfridge, O.G., "Pandemonium : a paradigm for learning", in *The mechanisation of thought processes*. H.M.S.O., London, 1959.

- [Seni94] Seni, G., and Cohen, E., "External Word Segmentation of Off-line Handwritten Text Lines", *Pattern Recognition*, Vol. 27, No. 1, pp. 41–52, 1994.
- [Seni93] Senior, A.W., and Fallside, F., "An Off-line Cursive Script Recognition System Using Recurrent Error Propagation Networks", in *Proc. Int. Workshop on Frontiers in Handwriting Recognition (IWFHR-3)*, pp. 132–141, 1993.
- [Shri84] Shridhar, M., Badreldin, A., "Handwritten Numeral Recognition By Tree Classification Methods", *Image and Vision Computing*, Vol. 2, pp. 143–149, 1984.
- [Shri84b] Shridhar M., and Badreldin, A., "High accuracy character recognition algorithm using Fourier and topological descriptors", *Pattern Recognition*, Vol. 17, No. 5, pp. 515–524, 1984.
- [Shri86] Shridhar M., and Badreldin, A., "Recognition of Isolated and Simply Connected Handwritten Numerals", *Pattern Recognition*, Vol. 19, No. 1, pp. 1–12, 1986.
- [Sima92] Simard, P., Victorri, B., et al, "Tangent Prop – A formalism for specifying selected invariances in a adaptative network", in *Advances in Neural Information Processing Systems*, Vol. 4, pp. 895–903, 1992.
- [Simo92] Simon, J.C., "Off-line Cursive Word Recognition", in *Proc. of the IEEE*, Vol. 80, No. 7, pp. 1150–1161, 1992.
- [Sinh93] Sinha, R.M.K., Prasada, B., et al, "Hybrid Contextual Text Recognition with String Matching", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 9, pp. 915–925, Sep. 1993.
- [Siy91] Siy, P., Chen, C.S., "Fuzzy Logic for Handwritten Numeral Character Recognition", *Fuzzy Models for Pattern Recognition—Methods That Search for Structure in Data*, IEEE Press, New York, pp. 321–325, 1991.
- [Srih91] Srihari, S. N. "High Performance Reading Machines", in *Pattern Recognition : Architectures, Algorithms and Applications*, pp. 271–300, 1991.

- [Srih93] Srihari, S. N. "Recognition of handwritten and machine-printed text for postal address interpretation", *Pattern Recognition Letters*, No. 14, pp. 291–302, 1993.
- [Srih87] Srihari, S.N. and Bozinovic, R. M., "A Multilevel Perception Approach to Cursive Script Recognition", in *Artificial Intelligence Journal*, North Holland, Vol. 33, No. 2, pp. 217–255, 1987.
- [Srih99] Srihari, S.N., Yang, W. and Govindaraju, V. "Information Theoretic Analysis of Postal Address Fields for Automatic Address Interpretation", in *Proc. Int. Conf. on Doc. Anal. and Recog. (ICDAR'99)*, pp. 309–312, 1999.
- [Srik96] Srikantan, G. Lam, S. W. and Srihari, S. N. "Gradient-Based Contour Encoding for Character Recognition", *Pattern Recognition*, Vol. 29, No. 7, pp. 1147–1160, 1996.
- [Stei00] Steinhertz, T., Rivlin, E. and Intrator, N. "Offline cursive script word recognition - a survey", in *Proc. Int. Journal on Document Analysis and Recognition*, No. 2, pp. 90–110, 2000.
- [Suen83] Suen, C.Y., "Handwriting Generation, Perception, and Recognition", *Acta Psychologica*, Vol 54, pp. 295–312, 1983.
- [Suen86] Suen, C.Y., "Character Recognition by Computer and Applications", in *Handbook of Pattern Recognition and image Processing*, Yong T. Y. and Fu K. S. (Eds.), 1986.
- [Suen80] Suen, C.Y., Berthod, M., and Mori, S. "Automatic Recognition of Handprinted Characters : The State of the Art", in *Proc. of the IEEE*, Vol. 68, No. 4, pp. 469–487, 1980.
- [Taft93] Taft, M., *Reading and the Mental Lexicon*, in *Essays in Cognitive Psychology*, Lawrence Erlbaum Associates, 1993.

- [Taka91] Takahashi, H., "A neural net OCR using geometrical and zonal pattern features", in Proc. Int. Conf. on Doc. Anal. and Recog. (ICDAR'91), pp. 821–828, 1991.
- [Tapp82] Tappert, C.C., "Cursive Script by Elastic matching", IBM Journal for Research and Development, Vol. 26, pp. 765–771, Nov. 1982.
- [Tapp84] Tappert, C.C., "Adaptive on-line hand-writing recognition", in Proc. Int. Conf. on Pattern Recognition, Vol. 2, pp. 1004–1007, 1984.
- [Tapp88] Tappert, C.C., Suen, C.Y. and Wakahara, T., "On-line Handwriting Recognition - A Survey", in Proc. IEEE Int. Conf. on Pattern Recognition, pp. 1123–1132, Rome, Italy, 1988.
- [Tapp90] Tappert, C.C., Suen, C.Y. and Wakahara, T., "The State of the Art in On-Line Handwriting Recognition", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 12, No. 8, pp. 787–808, 1990.
- [Tayl83] Taylor, I. and Taylor, M.M., *The Psychology of Reading*, Academic Press, New York, 1983.
- [Tayl84] Taylor, M.M., "The bilateral cooperative model of reading : a human paradigm for artificial intelligence", in *Artificial and human intelligence*, Elsevier Science, pp. 239–250, 1984.
- [Thép94] Thépaut, A., and Autret, Y., "Une méthode de reconnaissance de chiffres manuscrits basée sur l'association de deux réseaux connexionistes", in *Actes du Colloque National sur l'Écrit et le Document (CNED'94)*, pp. 119–125, Rouen, France, 1994.
- [Trei85] Treisman, A. "Preattentive Processing in Vision", in *Computer Vision Graphics and Image Processing*, Vol. 31, No. 2, pp. 156–177, 1985.
- [Trei86] Treisman, A., "Features and objects in visual processing", *Scientific American*, Vol. 255, No. 5, pp. 106–115, 1986.

- [Trie96] Trier, O.D., Jain, A.K., Taxt, T., “Feature-Extraction Methods for Character-Recognition : A Survey”, *Pattern Recognition*, Vol. 29, No. 4, pp. 641–662, 1996.
- [Trup97] Trupin, E. “Du traitement automatisé de formulaires”, in *Document numérique*, Vol. 1, No. 1, pp. 39–53, 1997.
- [Ullm84] Ullman, S. “Visual Routines”, in *Cognition*, No. 18, pp. 97–159, 1984.
- [Uttal75] Uttal, W.R. *An autocorrelation theory of form detection*, Hillsdale, N.J., Lawrence Erlbaum (eds), 1975.
- [Wagn74] Wagner, R.A., and Fischer, M.J., “The string-to-string correction problem”, *Journal of the ACM*, Vol. 21, No. 1, pp. 168–173, 1974.
- [Wang94] Wang, J., and Jean, J., “Segmentation of Merged Characters by Neural Networks and Shortest Path”, *Pattern Recognition*, Vol. 27, No. 5, pp. 649–658, 1994.
- [Ward87] Ward, J.R. and Phillips, M.J., “Digitizer technology : Performance characteristics and the effects on the user interface”, *IEEE Computer Graphics and Applications*, Vol. 7, pp. 31–44, 1987.
- [Weso96] Wesolkowski, S. “Cursive script recognition : A survey”, in *Handwriting and Drawing Research : Basic and Applied Issues*, IOS Press, Amsterdam, pp. 267–284, 1996.
- [West93] Westall, J.M., and Narasimha, M.S., “Vertex Directed Segmentation of Handwritten Numerals”, *Pattern Recognition*, Vol. 26, No. 10, pp. 1473–1486, 1993.
- [Whit98] Whitaker, D. and McGraw, P.V. “Geometric representation of the Mechanisms underlying human curvature detection”, *Vision Research*, Vol. 38, pp. 3843–3848, 1998.
- [Wilk92] Wilkinson, R.A., “Comparison of Massively Parallel Segmenters”, *National Institute of Standards and Technology, technical report*, Gaithersburg, Md., 1992.

- [Yaco96] Yacoubi, E., Modélisation Markovienne de l'écriture manuscrite - Application à la reconnaissance des adresses postales, Thèse de Doctorat de l'Université de Rennes I, 1996.
- [Yaco99] Yacoubi, E., Gilloux, M., et al. "An HMM-Based Approach for Off-Line Unconstrained Handwritten Word Modeling and Recognition", *IEEE Trans. on Patt. Anal. and Mach. Intell.*, Vol. 21, No. 8, pp. 752–760, 1999.
- [Yang98] Yang, Y.Y. "Adaptive Recognition of Chinese Characters : Imitation of Psychological Process in Machine Recognition", *IEEE Trans. on Systems, Man and Cybernetics - Part A : Systems and Humans*, Vol. 28, No. 3, pp. 253–265, 1998.
- [Yani94] Yanikoglu, B.A., and Sandon, P.A., "Recognizing Off-line Cursive Handwriting", in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 397–403, 1994.
- [Youn96] Young, R.C.D. and Chatwin, C.R. "Computation of the forward and inverse Radon transform via the central slice theorem employing a non-scanning optical technique", in *Proc. of the SPIE*, vol. 2752, pp. 306–16, 1996.
- [Zhu99] Zhu, X., Shi, Y. and Wang, S. "A New Distinguishing Algorithm of Connected Character Image Based on Fourier Transform", in *Proc. Int. Conf. on Doc. Anal. and Recog. (ICDAR'99)*, pp. 788–791, 1999.