



HAL
open science

Modèles de Markov multi-flux pour la reconnaissance de l'écriture manuscrite multi-scripts

Yousri Kessentini

► **To cite this version:**

Yousri Kessentini. Modèles de Markov multi-flux pour la reconnaissance de l'écriture manuscrite multi-scripts. Traitement du texte et du document. Université de Rouen, 2009. Français. NNT : . tel-01415112

HAL Id: tel-01415112

<https://hal.science/tel-01415112>

Submitted on 12 Dec 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Laboratoire d'Informatique, de Traitement de l'Information et des
Systèmes

UNIVERSITE DE ROUEN
U.F.R DES SCIENCES ET TECHNIQUES

Thèse en vue de l'obtention du diplôme de
Docteur de l'Université de Rouen

Discipline: Informatique

**Modèles de Markov multi-flux pour la
reconnaissance de l'écriture manuscrite
multi-scripts**

Yousri Kessentini

Soutenue le 5 Mai 2009 devant le jury composé de :

M	Abdel	Belaid	Université Nancy 2	Rapporteur
M	Abdelmajid	Ben Hamadou	ISIMS de Sfax	Co-directeur de thèse
M	Christian	Viard-Gaudin	Université de Nantes	Examineur
M	Laurent	Heutte	Université de Rouen	Examineur
M	Noureddine	Ellouze	ENIT de Tunis	Rapporteur
M	Thierry	Paquet	Université de Rouen	Directeur de thèse
M	Vincent	Poulain d'Andecy	ITESOFT	Invité

A la mémoire de ma mère
A ma femme Sourour et mon fils Rayane
A mon père et mes frères

Remerciements

Ce travail de thèse a été réalisé dans le cadre d'une co-tutelle entre le laboratoire LITIS de l'université de Rouen (France) et le laboratoire MIRACL de l'université de Sfax (Tunisie).

Je tiens à remercier tous ceux qui m'ont aidé de près ou de loin au cours de mes recherches :

- Thierry Paquet, pour avoir accepté d'encadrer mes travaux en me faisant bénéficier de ses précieux conseils. Son suivi constant et attentif, le dynamisme qu'il a su me communiquer m'ont permis de mener à bien ce travail. Qu'il trouve ici l'expression de ma profonde gratitude.
- Abdelmajid Ben Hamadou, pour avoir accepté de co-diriger mes travaux. Pour l'aide et les conseils qu'il m'a prodigués durant les années de thèse, je voudrais ici témoigner ma reconnaissance.

Je remercie chaleureusement Nouredine Ellouze et Abdel Belaid pour avoir accepté le difficile rôle de rapporteur de ces travaux ainsi que Vincent Poulain d'Andecy, Christian Viard-Gaudin et Laurent Heutte d'avoir accepté de prendre part à mon jury.

Je voudrais également remercier tous les membres des laboratoires LITIS et MIRACL pour l'amitié et l'aide qu'ils m'ont témoignés durant cette thèse.

Enfin, je voudrais exprimer toute ma reconnaissance à mes proches et à mes amis qui m'ont toujours soutenu et encouragé pendant ce long travail.

Résumé

Les travaux présentés dans cette thèse concernent le développement et l'évaluation d'un système de reconnaissance de l'écriture manuscrite multi-scripts. L'objectif est de proposer un système de reconnaissance opérant indépendamment de la nature du script. Pour cela l'approche proposée est sans segmentation explicite, ceci afin de ne pas introduire une technique de conditionnement de l'information dépendant du type d'écriture analysée. Des caractéristiques bas niveaux sont combinées à travers une approche multi-flux permettant de combiner d'une manière asynchrone, différentes sources d'information à travers des modèles de Markov coopératifs. Des extensions des règles de combinaison sont envisagées dans le cadre de la théorie des fonctions de croyances transférables. L'évaluation du système de reconnaissance proposé est menée sur différentes bases de données de référence correspondant aux deux scripts Arabe (IFN-ENIT) et Latin (IRONOFF et RIMES). Les résultats montrent que le système proposé donne de bons résultats aussi bien sur le Latin que sur l'Arabe comparables à ceux des meilleurs systèmes présentés dans la littérature.

Mots-clés : Reconnaissance de l'écriture manuscrite, mutli-scripts, combinaison de l'information, Modèles de Markov Cachés multi-flux, fonction de croyance, théorie de Dempster-Shafer.

Abstract

This PhD thesis is meant to be applied to multi-script handwritten word recognition. The proposed system is script independent. It proceeds without explicit segmentation of handwriting into graphemes and makes use of low level feature sets irrespective of the scripts. Features are then combined according to the multi-stream paradigm. Other track has been explored in this PhD thesis exploiting the Dempster-Shafer theory to propose a new combination rules in the multi-stream HMM. The developed system has been tested on publicly available databases : the benchmark database IFN/ENIT (Arabic script), IRONOFF and RIMES databases (Latin script). For both scripts the results show significant improvement while using a multi-stream approach.

Keywords : Off-Line handwriting recognition, Hidden Markov Models, multi-script, multi-stream, information combination, belief functions, Dempster-Shafer theory.

Table des matières

Introduction générale	17
1 État de l’art sur la classification	21
1.1 Introduction	21
1.2 Les approches par appariement de formes	22
1.3 Les approches statistiques	22
1.3.1 Classifieur Bayésien	23
1.3.2 K plus proches voisins	24
1.3.3 SVM	24
1.4 Les approches structurelles et syntaxiques	25
1.5 Les réseaux de neurones	26
1.6 Les approches Markoviennes	27
1.6.1 Exemple 1 : modèles de Markov observables	28
1.6.2 Exemple 2 : modèles de Markov cachés	29
1.6.3 Modèles pseudo bidimensionnels : PHMM ou Planar HMM	39
1.6.4 Modèles bidimensionnels : champs de Markov	41
1.7 Les modèles hybrides : MMC et réseaux de neurones	46
1.8 Modèles de Markov multi-flux où “multi-stream”	47
1.8.1 Formalisme multi-flux	48
1.8.2 Stratégies de combinaison	50
1.8.3 Aspects de décodage multi-flux	51
1.8.4 Apprentissage	53
1.9 Les techniques de combinaison de classifieurs	56
1.9.1 La combinaison séquentielle	56
1.9.2 La combinaison parallèle	57
1.9.3 Les architectures hybrides	58

1.10	Conclusion	58
2	La reconnaissance de l'écriture manuscrite hors-ligne	61
2.1	Introduction	61
2.2	Traitement de l'écriture manuscrite multi-script	62
2.2.1	Complexité du traitement de l'écriture manuscrite	63
2.2.2	Caractéristiques de l'écriture Arabe	64
2.2.3	Traitement de l'écriture multi-script	66
2.3	Architecture d'un système de reconnaissance de mots manuscrits isolés	69
2.3.1	Pré-traitement	70
2.3.2	Extraction des primitives	72
2.3.3	Classification	73
2.4	Stratégies de reconnaissance de mots isolés	74
2.4.1	Les approches globales	75
2.4.2	Les approches analytiques	76
2.4.3	Conclusion	79
2.5	État de l'art sur les systèmes de reconnaissance de l'écriture manuscrite	80
2.5.1	Application des modèles $1D$	80
2.5.2	Application des modèles $1D^{1/2}$	85
2.5.3	Application des modèles $2D$	87
2.5.4	Application des approches neuro-Markovienne	90
2.5.5	Application des modèles multi-flux	91
2.5.6	Application des approches non-markoviennes	93
2.6	Conclusion	95
3	Un système multi-flux pour la reconnaissance de mots manuscrits multi-script	99
3.1	Introduction	99
3.2	Description du système de reconnaissance	100
3.2.1	Pré-traitements	100
3.2.2	Extraction des caractéristiques	103
3.2.3	Modélisation des lettres et des mots	106
3.2.4	Apprentissage	107
3.3	Expérimentations et résultats	108
3.3.1	La base de mots Arabe IFN/ENIT	109

3.3.2	Résultats sur la base IFN/ENIT	110
3.3.3	Comparaison avec d'autres systèmes	111
3.3.4	La base de mots Latin IRONOFF	113
3.3.5	La base de mots Latin RIMES	116
3.3.6	Adaptation des probabilités de transition du MMC composite	117
3.3.7	Conclusion	118
3.3.8	Extension au cas à N-flux	118
3.4	Conclusion	121
3.4.1	Performance du système de reconnaissance sur une base multi-lingue	122
3.5	Conclusion	124
4	Application du Modèle des Croyances Transférables à la combinaison d'informations multi-flux	127
4.1	Introduction	127
4.2	Outils de base des fonctions de croyance	128
4.2.1	Masse de croyance élémentaire	129
4.2.2	Transformations de la fonction de masse	131
4.2.3	Règles de combinaison	132
4.2.4	Prise de décision à partir de fonctions de croyance . .	134
4.3	Opérations avancées sur les fonctions de croyance	135
4.3.1	Combinaison de fonctions de croyance définies sur un cadre de discernement différent	135
4.3.2	Construction de modèles de fonctions de croyance à partir des fonctions de vraisemblances	137
4.3.3	Le déconditionnement pour la fusion de FDM condi- tionnelles	139
4.4	Approche proposée	140
4.4.1	Les différentes étapes de l'approche proposée	140
4.4.2	Définition des cadres de discernements	142
4.4.3	Construction des fonctions de masse	142
4.4.4	Combinaison des fonctions de masse	143
4.4.5	Complexité de l'approche proposée	144
4.4.6	Évaluation de l'approche proposée sur la reconnais- sance de mots	146

4.5	MMC crédibiliste	147
4.5.1	L'algorithme de Viterbi crédibiliste	148
4.5.2	Complexité de l'approche	151
4.6	Conclusion	152
	Conclusion générale	155
	Bibliographie	161

Table des figures

1.1	Exemple de modélisation de l'évolution du temps	28
1.2	Quelques exemples d'architecture de MMC	31
1.3	Calcul des variables Forward (a) et Backward (b)	32
1.4	Exemple d'architecture d'un PHMM, d'après [Belaid 97].	40
1.5	(a) Système de voisinage d'ordre 1, (b) système de voisinage d'ordre 2, (c) hiérarchie de systèmes de voisinages, (b) et (c) cliques associées au système d'ordre 1, (d)+(e)+(f)+(g)+(h) cliques associées au système d'ordre 2, d'après [Saon 97].	43
1.6	Ensemble de sites associés à un NSHP	44
1.7	Exemple de NSHP-HMM appliqué sur une lettre d'après [Saon 97].	45
1.8	Structure générale d'un modèle multi-flux	49
1.9	Topologie d'un modèle parallèle à 2 flux.	52
1.10	Topologie du modèle MMC résultant du modèle à 2 flux de la Figure 1.9.	53
1.11	Combinaison séquentielle de classifieurs	57
1.12	Combinaison parallèle de classifieurs	57
1.13	Combinaison hybride de classifieurs	58
2.1	Difficultés de l'écriture manuscrite	63
2.2	Classification de l'écriture selon Tappert	64

2.3	Quelques caractéristiques de l'écriture arabe selon [Zheng 04]. (1) Écriture de droite à gauche . (2) Un mot arabe contient trois parties cursives. (3) Un mot se compose de six caractères. (4) Quelques caractères ne sont pas connectés du côté gauche avec le caractère suivant. (5) Le même caractère avec diffé- rentes formes selon sa position dans le mot. (6) Différents ca- ractères avec différentes tailles. (7) Différents caractères avec différents nombre de points. (8) Différents caractères ont le même nombre de points mais à des positions différentes. . . .	65
2.4	Signes diacritiques en Arabe	65
2.5	Alphabet arabe	67
2.6	Exemple d'écriture manuscrite Arabe selon [Amara 00]. . . .	68
2.7	Exemples d'écritures imprimées Arabes selon [Amara 00]. . .	68
2.8	schéma général d'un système de reconnaissance de l'écriture manuscrite	69
2.9	Exemple de segmentation explicite réalisée dans [El-Yacoubi 99]	77
2.10	Exemple de segmentation implicite réalisée dans [Tay 01] . .	78
2.11	Séquence de symboles décrivant un mot dans [El-Yacoubi 99]	82
2.12	HMM modélisant une lettre dans [El-Yacoubi 99]	82
2.13	Modèle global de reconnaissance du mot "SET", d'après [El-Yacoubi 99]	83
2.14	Extraction de vecteurs de primitives avec des fenêtres incli- nées [Al-Hajj 05]	84
2.15	Topologie d'un HMM modélisant un caractère selon [Schambach 08]	84
2.16	PHMM pour un caractère imprimé, d'après [Kuo 94].	86
2.17	Architecture du PHMM, d'après [Amara 99].	87
2.18	Le modèle HMMRF proposé par Park, d'après [Park 95]. . .	89
2.19	Vue d'ensemble du système de reconnaissance hybride MMC/RN [Koerich 03]	91
2.20	DBN : modèle couplé par états	93
2.21	Génération des points de segmentation	94
2.22	Séparation en boîtes englobantes	94
2.23	Segmentation optimale en caractères	95
3.1	Exemples de pré-traitements	101

3.2	Estimation des positions des lignes de base des mots	102
3.3	Images des contours du mot.	103
3.4	Différents types des points du contour supérieur.	104
3.5	Extraction des caractéristiques dans les 3 zones d'écriture . .	104
3.6	Les 5 types de configurations locales autour d'un pixel de fond P105	
3.7	Les 5 types de configurations locales autour d'un pixel de fond P106	
3.8	Exemples de la base de référence IFN-ENIT, d'après [Pechwitz 02]	109
3.9	IFN/ENIT : Résultats de reconnaissance avec 2 flux	111
3.10	IRONOFF : Résultats de reconnaissance avec 2 flux	114
3.11	Architecture du système de reconnaissance multi-script	122
3.12	Exemples de confusion du système de reconnaissance multi- script	124
4.1	Représentation abstraite des mécanismes dans le Modèle des Croyances Transférables.	128
4.2	Représentation des éléments focaux de certaines classes de fonctions de croyance.	131
4.3	Raffinement de cadre de discernement : Θ est un raffinement de Ω	136
4.4	Différentes étapes de l'approche de combinaison proposée . .	141

Liste des tableaux

2.1	Les caractéristiques utilisées pour la reconnaissance de l'écriture manuscrite arabe	73
2.2	Quelques systèmes de reconnaissance de l'écriture manuscrite arabe	75
3.1	IFN/ENIT : Résultats de reconnaissance avec un seul flux . .	110
3.2	IFN/ENIT : Comparaison de l'approche multi-flux aux approches par fusion des décisions et des représentations	111
3.3	IFN/ENIT : Comparaison aux systèmes présentés dans la compétition ICDAR'05	112
3.4	IFN/ENIT : Comparaison aux systèmes présentés dans la compétition ICDAR'07 (ID 01 : MITRE; IDs 02-04 : CACI; ID 05 : CEDAR; ID 06 : MIE; IDs 07-08 : SIEMENS; IDs 09-12 : UOB-ENST; ID 13 : ICRA; ID 14 : PARIS V)	113
3.5	IRONOFF : Résultats de reconnaissance avec un seul flux . .	114
3.6	IRONOFF : Comparaison de l'approche multi-flux aux approches par fusion des décisions et des représentations	115
3.7	IRONOFF : Comparaison aux systèmes présentés dans la littérature	115
3.8	Performances sur la base RIMES	117
3.9	Performances avec et sans adaptation sur la base IRONOFF .	118
3.10	Performances sur la base IRONOFF-Chèque	120
3.11	Performances sur la base IFN/ENIT	121
3.12	Résultats de reconnaissance sur la base multi-lingue	123
3.13	Taux de confusion sur la base multi-lingue	123
4.1	Performances sur la base IRONOFF-Chèque	146

Introduction générale

Si la lecture est une activité courante et naturelle pour l'homme, puisqu'il parvient à reconnaître des messages écrits parfois très dégradés, elle représente pour la machine une tâche très complexe. Il y a trente ans, le contexte technologique et méthodologique était bien différent, et l'on n'imaginait pas reconnaître automatiquement des pages entières de documents. Les performances des machines actuelles et les recherches menées depuis ont permis de concrétiser certains systèmes de lecture automatique, non seulement des documents imprimés et dactylographiés mais également plus récemment ceux manuscrits, problème pourtant jugé plus difficile du fait de la variabilité de l'écriture manuscrite. Parmi ces études quelques approches sont maintenant mises en oeuvre dans des applications industrielles telles que la lecture d'adresses postales pour le tri automatisé du courrier, la lecture de chèques bancaires ou encore le traitement automatique de formulaires. Néanmoins la reconnaissance de documents moins contraints dans leur mise en forme et dans le lexique utilisé, reste encore un problème très difficile à résoudre et fera encore l'objet de nombreux travaux de recherche les prochaines années. Généralement, le script le plus utilisé dans les systèmes d'informations est le Latin (français, anglais, ...). Cependant, ces dernières années, avec la croissance des moyens de communication, d'autres alphabets non latins ont intégré ces systèmes d'informations. Ceci a entraîné l'apparition de documents multi-lingues ou multi-script où l'écriture figure sous différentes langues dans le même document. La reconnaissance de tels documents nécessite des techniques de traitement plus génériques.

Le sujet traité dans cette thèse concerne la reconnaissance de l'écriture manuscrite multi-script. Nous proposons dans ce cadre une approche de

reconnaissance indépendante de la nature du script. L'approche proposée opère sans segmentation explicite des mots en graphèmes. Cette tâche est particulièrement délicate conformément au dilemme de Sayre [Sayre 73] : "il faut segmenter pour reconnaître, mais il faut reconnaître pour segmenter", d'autant plus quand il s'agit de l'écriture multi-script, dans ce cas, une méthode de segmentation d'un script donné peut s'avérer inopérante sur un autre script. D'autre part, les approches markoviennes conviennent bien à modéliser l'écriture multi-script du fait qu'elles permettent de modéliser les différentes variations morphologiques des caractères ainsi que les problèmes épineux de segmentation. De plus, nous avons choisi de travailler avec des caractéristiques bas niveaux, basées sur les directions et les densités des pixels. Ces caractéristiques sont combinées à travers une approche markovienne multi-flux, permettant une combinaison asynchrone de plusieurs sources d'informations. Nous évaluons le système de reconnaissance proposé sur des bases publiques de mots Arabes et Latins. Nous montrons que l'approche proposée donne de bons résultats aussi bien sur ces deux scripts indépendamment, que sur une base multi-lingue.

Ce rapport est organisé selon quatre chapitres : dans le chapitre 1, nous passons en revue les principales méthodes de classification utilisées dans le cadre de la reconnaissance de l'écriture. Un accent particulier est donné aux approches markoviennes utilisées dans ce travail. Nous présentons une synthèse assez détaillée des concepts fondamentaux des MMC classiques appelés MMC unidimensionnels, ainsi que les extensions de ces modèles.

Le chapitre 2 dresse un état de l'art sur le domaine de la reconnaissance de l'écriture. Nous mettons l'accent sur la complexité du traitement de l'écriture manuscrite en général, et l'écriture multi-script en particulier, en nous limitant à l'étude des deux langues Arabe et Latine. Nous présentons l'architecture optimale d'un système de reconnaissance d'écriture manuscrite et nous proposons un panorama des différentes stratégies proposées de reconnaissance de mots isolés. Nous tentons d'en tirer partie pour envisager leur application à l'écriture multi-script. La dernière partie de ce chapitre est consacrée à l'étude de quelques systèmes de reconnaissance de l'écriture présentés dans la littérature.

Le troisième chapitre présente notre contribution à la reconnaissance hors-ligne de l'écriture multi-script. Nous présentons dans un premier temps l'architecture de notre système de reconnaissance de l'écriture à travers ses différentes étapes de pré-traitement, d'extraction de caractéristiques et de classification. La partie suivante est consacrée à l'évaluation des performances de notre système de reconnaissance. Pour cela, différentes bases de données de référence ont été utilisées. Les résultats montrent que le système proposé donne de bons résultats aussi bien sur le Latin que sur l'Arabe. Ils sont comparables aux meilleurs systèmes présentés dans la littérature. L'évaluation du système de reconnaissance proposé sur une base de mots multi-lingue montre que le système parvient à bien identifier et reconnaître les mots multi-lingues avec un taux de confusion très faible.

Dans le quatrième chapitre, d'autres pistes sont explorées afin d'exploiter la théorie des modèles de croyances transférables (MCT) et de l'appliquer dans le cadre de la problématique de reconnaissance de l'écriture. Nous présentons les mécanismes de base des MCT ainsi que quelques outils plus avancés pour présenter les modèles de Markov crédibilistes. Nous décrivons par la suite une approche de combinaison d'information à base de MCT dans un modèle multi-flux et nous l'évaluons dans le cadre de la reconnaissance de mots manuscrits.

Le mémoire se termine par une conclusion sur nos apports à la reconnaissance de l'écriture manuscrite, et des perspectives de recherche à court, moyen et long termes.

Chapitre 1

État de l'art sur la classification

1.1 Introduction

La reconnaissance des formes consiste en une automatisation de tâches de perception artificielle réalisées usuellement par le cerveau et le système sensoriel humain. Son objectif est de classifier des entités en catégories à partir d'observations effectuées sur celles-ci. Ce domaine a été depuis longtemps bien étudié à travers différentes applications telles que la vision par ordinateur, la reconnaissance de la parole ou la reconnaissance de l'écriture... Dans le processus complet d'une application de reconnaissance des formes, la classification joue un rôle important en se prononçant sur l'appartenance d'une forme à une classe. On dispose aujourd'hui, dans le domaine de la reconnaissance de formes, d'un grand nombre de classifieurs qui sont plus ou moins bien adaptés à la reconnaissance de l'écriture. Cependant, cela n'a pas permis de mettre en évidence la supériorité incontestable d'une méthode de classification sur une autre.

D'après [Jain 00], il existe quatre grandes familles de classifieurs : le pattern matching (ou "appariement de formes"), les approches structurelles ou syntaxiques, la classification statistique, et les réseaux de neurones.

Ce chapitre a pour objectif de décrire quelques méthodes de classification automatique appliquées à la reconnaissance de l'écriture. Un accent particulier a été mis sur les approches Markoviennes qui constituent l'une des bases de notre travail.

1.2 Les approches par appariement de formes

Les approches par appariement de formes (Template Matching en anglais) visent à comparer une forme à des représentants de chaque classe via une mesure de similarité. Le meilleur appariement au sens de cette mesure permet ainsi de décider que la forme à reconnaître appartient à tel modèle plutôt qu'à tel autre. Il y a trois types de mesures qui sont couramment utilisées pour juger de la qualité de mise en correspondance (appariement) [Heutte 94] : les mesures de ressemblance du type intercorrélacion ou intercorrélacion normée ; les mesures de dissemblance telles que les distances de Hamming, Chebychev ou euclidienne ; les mesures de similarité du type Jaccard, Yule. Bien que ces approches soient parfaitement adaptées à la reconnaissance de l'écriture imprimée monofontes, elles sont peu adaptées à la reconnaissance de l'écriture manuscrite car la très forte variabilité des caractères manuscrits impliquerait un nombre très important de représentants pour chaque classe.

1.3 Les approches statistiques

Avec une stratégie de reconnaissance de type statistique, les formes à identifier sont en général vues comme des points dans un espace à n dimensions, n étant le nombre de caractéristiques de l'espace. Chaque forme x appartenant à la classe u_i est vue comme une observation générée aléatoirement par la distribution de probabilité de la classe u_i : $p(x/u_i)$. Ces techniques de classification sont dites statistiques car elles font intervenir des fonctions de décisions statistiques. Pour un ensemble d'apprentissage donné, on peut construire les frontières de décision de deux manières différentes. La première solution consiste à générer les frontières implicitement à partir des distributions de probabilité de chaque classe (approches modélisantes : fenêtres de Parzen, mixture de gaussiennes, K plus proches voisins). Le deuxième type d'approche consiste à estimer explicitement les frontières de décision entre les classes (approches discriminantes).

Nous nous limitons à présenter dans la suite quelques méthodes de classification les plus utilisées pour la reconnaissance de l'écriture. Nous renvoyons aux [Belaid 92, Bishop 06] pour plus de détails sur les approches statistiques.

1.3.1 Classifieur Bayésien

L'application des méthodes statistiques bayésiennes à la reconnaissance des formes a été formalisée par Chow [Chow 65]. Dans la classification bayésienne, on construit les frontières de décision à partir des distributions de probabilités de chaque classe : $p(x/u_i)$. Lorsque ces densités de probabilités sont connues, on peut obtenir directement les probabilités a posteriori d'appartenance de la forme à chaque classe en appliquant la règle de Bayes :

$$p(u_i/x) = \frac{p(x/u_i) \cdot p(u_i)}{p(x)}$$

La décision se fait alors en choisissant pour x la classe qui minimise le risque conditionnel $R(u_i/x)$:

$$R(u_i/x) = \sum_{j=1}^{n_c} L(u_i, u_j) \cdot p(u_j/x)$$

où $L(u_i, u_j)$ désigne le coût de mauvaise classification, c'est-à-dire le coût engendré par la décision u_i à la place de la vraie classe u_j .

Cependant les densités de probabilités $p(x/u_i)$ ne sont généralement pas connues, et elles doivent être estimées à partir d'un ensemble d'apprentissage. Afin de simplifier l'estimation de ces densités on suppose que leur forme est connue à l'avance. Classiquement, la distribution gaussienne est souvent considérée du fait que cette distribution ne requière que l'estimation de deux paramètres (moyenne et matrice de covariance) et à l'existence de méthodes efficaces comme l'estimateur du maximum de vraisemblance [Scott 00] pour l'estimation des ces paramètres. Une fois ces paramètres estimés, la décision se fait naturellement par la règle de Bayes.

L'inconvénient de ce type d'approche est qu'il introduit un grand nombre de paramètres pour avoir des distributions de probabilités précises, surtout en grande dimension. En particulier, l'estimation des matrices de covariances pour chaque classe demande un nombre d'exemple dans la base d'apprentissage très important, ce qui la rend peu adaptée aux problèmes à grande dimension.

1.3.2 K plus proches voisins

La méthode des k plus proches voisins (K-PPV) [Cover 67] est une méthode qui cherche à déterminer directement la partition de l'espace de représentation en classes, sans faire d'hypothèse sur la nature de la distribution sous-jacente, ni sur la nature des surfaces séparatrices idéales.

En se basant sur le principe que si deux points sont suffisamment "proches" dans l'espace des caractéristiques alors il y a de grandes chances pour qu'ils appartiennent à la même classe, la classification d'une forme inconnue par K-PPV consiste à affecter cette forme à la classe de son plus proche voisin en la comparant aux formes des différentes classes de références nommées prototypes. Elle permet de déterminer les k formes les plus proches de la forme à reconnaître suivant un critère de similarité. Une stratégie de décision permet d'affecter des valeurs de confiance à chacune des classes en compétition et d'attribuer la classe la plus vraisemblable (au sens de la métrique choisie) à la forme inconnue. Le critère de similarité entre deux formes communément utilisé est basé sur la distance euclidienne (pour des raisons de rapidité de calcul). Le principal inconvénient de cette méthode est qu'elle est trop gourmande en temps de calcul parce que, pour que cette méthode soit précise, il faut fournir un très grand nombre d'échantillons, et en espace mémoire, vu qu'elle nécessite de calculer autant de distance sur l'espace de caractéristiques qu'il y a d'échantillons dans la base d'apprentissage.

1.3.3 SVM

Les machines à vecteurs de support (SVM) appelées aussi classifieurs à marge optimale ou encore séparateurs à vaste marge ont été introduites par [Vapnik 95]. Leur principe est de maximiser la marge entre les classes. Il faut donc déterminer l'hyperplan maximisant cette marge. Les SVM offrent des performances intéressantes pour la reconnaissance de caractères manuscrits [LeCun 98, Liu 02], mais ils sont peu applicables à la reconnaissance de mots (sauf éventuellement avec une segmentation explicite des mots en lettres). En effet, tout comme les K-PPV, ils travaillent avec des données en dimension fixe et ne permettent donc pas d'introduire la variabilité de longueur des mots. De plus, ils ont l'inconvénient d'être assez lents en phase d'apprentissage comme en phase de reconnaissance.

1.4 Les approches structurelles et syntaxiques

Les approches structurelles ou syntaxiques cherchent à structurer l'information en décrivant l'organisation géométrique (ou la structure) d'une forme complexe à partir des composantes les plus élémentaires appelées primitives. Elles reposent sur une représentation hiérarchique des formes. Chaque forme est vue comme un ensemble de sous-formes qu'on appelle "pattern", elles mêmes composées de patterns plus petites. Les plus petits patterns (i.e. primitives) sont des caractéristiques, par exemple un segment de droite, une occlusion ou un arc pour les caractères manuscrits.

Les méthodes structurelles et syntaxiques se déclinent en fonction du type de structure utilisée pour décrire la forme :

1. Dans les méthodes structurelles proprement dites [Baird 88], la structure utilisée est de type graphe qui permet de représenter la forme à l'aide de primitives et des relations entre ces primitives. La phase de décision consiste alors à comparer les graphes de représentation de la forme à reconnaître et ceux des formes prototypes selon le principe du pattern Matching.
2. Dans les méthodes syntaxiques [Belaïd 92], la structure retenue pour coder la forme est une liste ordonnée utilisant un alphabet dont chaque terme représente un élément de la forme à décrire. La phase de décision dans ce cas consiste à analyser la liste à l'aide de règles syntaxiques comme on pourrait le faire pour analyser une phrase du langage naturel à l'aide d'une grammaire. Les règles syntaxiques (grammaire) doivent être inférées à partir des exemples d'une base d'apprentissage.

Contrairement aux méthodes statistiques, nécessitant un grand nombre d'échantillons dans l'espace de représentation, les méthodes structurelles et syntaxiques n'utilisent qu'un nombre réduit de prototypes pour représenter une classe. Cela permet d'une part de réduire le coût global de mise en correspondance entre les représentations inconnues et les représentations prototypes et d'autre part, d'inclure des prototypes représentant des formes très particulières afin de prendre en compte la variabilité de ces formes à l'intérieur d'une même classe [Heutte 94]. Malgré ces avantages, ces approches sont très sensibles aux problèmes de segmentation qui modifient la structure des formes, ainsi qu'au bruit.

1.5 Les réseaux de neurones

Les réseaux de neurones ont connus un essor important grâce à un algorithme d'apprentissage très efficace, la rétropropagation du gradient de l'erreur [Zhang 00, Lecun 87, Bishop 95]. L'idée principale est qu'un neurone formel est capable de réaliser des calculs élémentaires comme la séparation d'un vecteur en deux classes, chaque classe étant déterminée par le poids du neurone. Le problème est alors de choisir quels coefficients affecter aux poids pour réaliser une séparation optimale. La multiplication des neurones permet de séparer plusieurs classes : il faut donc réaliser un choix sur la topologie du réseau.

Dans un modèle statistique, la connaissance, c'est-à-dire la distribution des classes, est représentée par un modèle mathématique (mélange de gaussiennes par exemple) dont les paramètres doivent être estimés. Ces modèles constituent une limitation puisqu'ils ne seront jamais qu'une approximation de la "forme" des classes. Selon Lecun [Lecun 87], le modèle connexionniste surmonte ce problème en représentant la connaissance sous la forme d'un réseau d'unités élémentaires reliées par des arcs pondérés. C'est dans ces connexions que réside la connaissance, et celle-ci peut prendre une forme plus variée qu'avec un modèle mathématique prédéfini.

Il existe de nombreuses topologies de réseaux de neurones :

- Les réseaux multicouches : ils sont organisés en couches, chaque neurone prend généralement en entrée les sorties de tous les neurones de la couche inférieure. Ils ne possèdent pas de cycles ni de connexions intra-couche. On définit alors une "couche d'entrée", une "couche de sortie", et n "couches cachées".
- Les réseaux à connexions locales : on reprend la même structure en couche que précédemment, mais avec un nombre de connexions limité : un neurone n'est pas forcément connecté à tous les neurones de la couche précédente.
- Les réseaux à connexions récurrentes : on a toujours une structure en couches, mais avec des retours ou des connexions possibles entre la sortie et l'entrée des neurones d'une même couche.
- Enfin dans les réseaux à connexions complètes, tous les neurones sont interconnectés, comme par exemple dans le modèle de Hopfield et la machine de Boltzmann.

Les réseaux de neurones sont bien adaptés à la reconnaissance de formes globales telles que des caractères isolés. Ils se limitent à la classification de formes simples car fondés sur une représentation en dimension fixe. Grâce à leur pouvoir discriminatif, ils présentent un couplage intéressant avec les modèles de Markov cachés dans le cadre de la reconnaissance des mots (voir section 1.7). Ils présentent toutefois quelques inconvénients tels que : la difficulté de contrôler le comportement du réseau durant l'apprentissage ; le problème de dimensionnement (on ne peut pas connaître a priori les dimensions du réseau pour un problème donné) ; le nombre d'exemples dans la base d'apprentissage doit être infini. Dans la pratique cette contrainte n'est évidemment jamais vérifiée. Pour un problème à m caractéristiques, il est communément admis qu'on dispose de suffisamment d'exemples avec m^2 exemples par classe.

1.6 Les approches Markoviennes

Les approches Markoviennes peuvent être considérées comme une sous-famille des approches statistiques. Contrairement aux approches précédemment décrites, les approches Markoviennes sont utilisées pour la modélisation de séquences.

Les modèles de Markov cachés (MMC) ont été introduits dans les années 70. Ce sont des outils statistiques qui reposent sur de fortes bases théoriques [Baum 70, Baum 72]. Ils ont étayé de nombreuses applications dans le domaine de la reconnaissance de la parole [Juang 91, Huang 90] mais aussi de la génétique (décryptage des séquences d'ADN)[Baldi 95] et le traitement automatique du langage naturel (traduction automatique, étiquetage de texte, reconstruction de texte bruités...) [Guyon 95, Katz 87].

A partir du début des années 90, les MMC ont connu un essor important en reconnaissance de l'écriture grâce à leur capacité d'intégration du contexte et absorption du bruit. Les différents travaux réalisés dans ce domaine reposent pour une grande part sur l'expérience accumulée dans le domaine de la reconnaissance de la parole où les MMC sont fréquemment employés. Car tout comme la parole, l'écriture se prête bien à une modélisation stochastique à tous les niveaux de la chaîne de reconnaissance : morphologique, lexical, syntaxique [Belaid 97].

Alors que la parole ne laisse pas de doute sur sa nature 1D, celle de l'écriture

ture manuscrite est plus discutable. En effet, suivant les points de vue et les méthodes, elle peut être interprétée comme étant purement 1D, ou elle peut être vue comme une image à part entière, enfin elle peut être considérée comme un signal à la fois 1D (signal temporel de la gauche vers la droite) et 2D (image). De ce fait, différentes approches markoviennes ont été proposées dans la littérature partant des approches unidimensionnelles (1D) en passant par les approches pseudo-2D ($1D^{1/2}$) et jusqu'aux approches bidimensionnelles 2D cherchant toutes à modéliser au mieux les différents aspects de l'écriture.

Ce sont ces approches qui sont utilisées dans ce travail et dont on détaillera le principe et le formalisme dans la suite.

1.6.1 Exemple 1 : modèles de Markov observables

Les modèles de Markov permettent de modéliser le comportement d'un système à partir des états observés sur ce système. Les observations sont une succession d'états.

Prenons l'exemple d'une station météo qui veut modéliser l'évolution du temps [Rabiner 89]. Regardons trois aspects du temps correspondant à trois états : état 1= s_1 : pluie, état 2= s_2 : nuage, état 3= s_3 : soleil. Chacun des états peut être suivi de l'un des trois autres, certaines transitions étant plus probables que d'autres. Donner un modèle M de l'évolution du temps revient à déterminer ces transitions (voir figure 1.1). Nous définissons alors une matrice A de transitions, où les termes a_{ij} correspondent à la probabilité d'évoluer de l'état s_i vers l'état s_j . Donner une prédiction à partir de ce

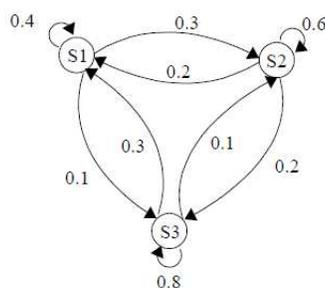


Figure 1.1. Exemple de modélisation de l'évolution du temps

modèle, c'est donner la probabilité que la succession d'états souhaités se réalise. Si l'on observe le temps «soleil» (état s_1) le lundi, la probabilité que

le temps du reste de la semaine soit « soleil, soleil, pluie, pluie, soleil, nuage » se calcule de la manière suivante : Soit $O = (s_3, s_3, s_3, s_1, s_1, s_3, s_2)$ une suite d'observations. La probabilité que cette observation se réalise est :

$$\begin{aligned}
 P(O/M) &= P(s_3, s_3, s_3, s_1, s_1, s_3, s_2) \\
 &= P(s_3, s_3, s_3, s_1, s_1, s_3) \times P(s_2/s_3, s_3, s_3, s_1, s_1, s_3) \\
 &= \dots \\
 &= P(s_3) \times P(s_3/s_3) \times P(s_3/s_3, s_3) \times P(s_1/s_3, s_3, s_3) \\
 &\quad \times P(s_1/s_3, s_3, s_3, s_1) \times P(s_3/s_3, s_3, s_3, s_1, s_1) \\
 &\quad \times P(s_2/s_3, s_3, s_3, s_1, s_1, s_3)
 \end{aligned}$$

Grâce L'hypothèse de Markov d'ordre 1 (dépendance à court terme), l'observation courante ne dépend que de l'état précédent :

$$P(s_t/s_{t-1}, s_{t-2}, \dots, s_0) = P(s_t/s_{t-1})$$

L'égalité précédente devient alors :

$$\begin{aligned}
 P(O/M) &= P(s_3) \times P(s_3/s_3) \times P(s_3/s_3) \times P(s_1/s_3) \times P(s_1/s_1) \times \\
 &\quad P(s_3/s_1) \times P(s_2/s_3) \\
 &= 1 \times (a_{33})^2 \times a_{31} \times a_{11} \times a_{13} \times a_{22} = 7.68 \times 10^{-4}
 \end{aligned}$$

1.6.2 Exemple 2 : modèles de Markov cachés

Le modèle de l'exemple précédent ne permet de travailler que si les observations que l'on réalise correspondent à des états du système, ce qui est une limitation en pratique. On considère à présent que les observations réalisées sur un système ne correspondent plus à des états de celui-ci. Il faut cependant pouvoir faire le lien entre les observations et les états. Considérons un système à trois états qui permettent de générer des chaînes du type « aaabb » :

- Le premier état ne permet de générer que des « a ».
- Le deuxième état peut générer des « a » et des « b » en proportion égale.
- Le troisième état ne permet de générer que des « b »

Ceci ce traduit par la matrice B suivante, dite matrice d'observation.

$$B = \begin{pmatrix} 1 & 0 \\ 0.5 & 0.5 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} P(o_t/s_j) \end{pmatrix}$$

Ce deuxième exemple, montre la différence entre un modèle de Markov observable et un modèle de Markov caché. L'adjectif "caché" employé pour caractériser le modèle, traduit le fait que l'état n'est pas observable mais accessible uniquement par les propriétés statistiques des lois d'émission.

On peut définir un modèle de Markov caché comme étant un processus doublement stochastique, constitué d'un processus sous-jacent non observable, mais qui peut être observé au travers d'un autre processus stochastique qui produit des séquences d'observations. Le processus caché est constitué d'un ensemble d'états modélisé par processus markovien. Le processus observable consiste en un jeu de sorties (observations), qui peuvent être émises par chaque état selon une fonction de densité de probabilité. Plus formellement, un MMC discret du premier ordre est défini par :

- $S = \{s_1, s_2, \dots, s_N\}$ l'ensemble des N états du modèle. On désigne un état au temps t par $q_t \in S$.
- $V = \{v_1, v_2, \dots, v_M\}$ l'ensemble discret des M symboles observables. On désigne un symbole observé au temps t par $o_t \in V$.
- $A = \{a_{i,j}\}_{1 \leq i,j \leq N}$, où $a_{i,j} = P(q_{t+1} = s_j | q_t = s_i)$, pour le modèle d'ordre 1. A est la matrice des probabilités de transitions entre états.
- $B = \{b_j(k)\}_{1 \leq j \leq N; 1 \leq k \leq M}$, où $b_j(k) = P(o_t = v_k | q_t = s_j)$. B est la matrice des probabilités d'émission dans les états.
- $\pi = \{\pi_i\}_{1 \leq i \leq N}$, où $\pi_i = P(q_1 = s_i)$. π est le vecteur des probabilités initiales des états.

Par simplification, on désignera un MMC par le triplet $\lambda = \{A, B, \pi\}$ [Rabiner 89]. La figure 1.2 montre quelques exemples d'architectures possibles de MMC.

A partir de ce modèle, trois problèmes sont posés :

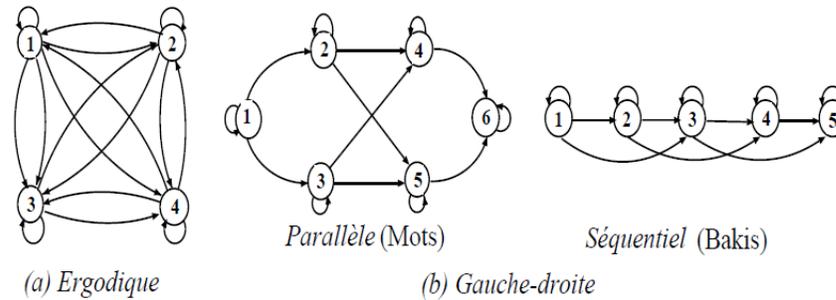


Figure 1.2. Quelques exemples d'architecture de MMC

- L'évaluation de la probabilité d'une séquence d'observation. Soit la séquence d'observations O et un modèle λ , il est primordial de savoir comment évaluer la probabilité de la séquence d'observation $P(O|\lambda)$.
- Apprentissage : Comment ajuster les paramètres du modèle λ pour maximiser $P(O|\lambda)$ sur un ensemble d'exemples de séquences ?
- La recherche du chemin le plus probable, ou estimation de la partie cachée, ou encore décision. Soit la suite d'observations O et un modèle λ , comment trouver la suite d'états Q qui soit optimale selon un certain critère ?

1.6.2.1 L'évaluation de la probabilité d'une séquence d'observation

Il existe plusieurs manières d'évaluer la probabilité d'une séquence d'observation, parmi lesquelles on peut citer l'évaluation directe et l'évaluation par les fonctions Forward-backward.

L'évaluation directe consiste à déterminer la probabilité d'une suite d'observations O , étant donné le modèle λ , en sommant sur tous les chemins d'états possibles Q des probabilités conjointes de O et de Q :

$$P(O|\lambda) = \sum_Q P(O, Q|\lambda) = \sum_Q P(O|Q, \lambda)P(Q|\lambda)$$

Selon les hypothèses Markoviennes :

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T}$$

Selon l'hypothèse d'indépendance des observations entre elles conditionnellement à la séquence d'états cachés Q :

$$P(O|Q, \lambda) = b_{q_1}(O_1)b_{q_2}(O_2) \dots b_{q_T}(O_T)$$

On déduit donc :

$$P(O|\lambda) = \sum_Q \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) a_{q_2 q_3} \dots a_{q_{T-1} q_T} b_{q_T}(O_T)$$

Pour un modèle à N états et pour T observations, cette méthode nécessite $(2T - 1).N^T$ multiplications et $N^T - 1$ additions (N^T étant le nombre de chemins possibles de longueur T), soit $2TN^T$ opérations, ce qui est en pratique rapidement incalculable pour des valeurs de T raisonnables. A titre d'exemple, pour $N = 5$ états et $T = 100$ observations, il nous faut, approximativement, $2.100.5^{100} \approx 10^{72}$ opérations.

Il existe une variante de l'évaluation directe qui procède selon l'algorithme Forward-Backward (figure 1.3), qui permet de réduire la complexité à N^2T .

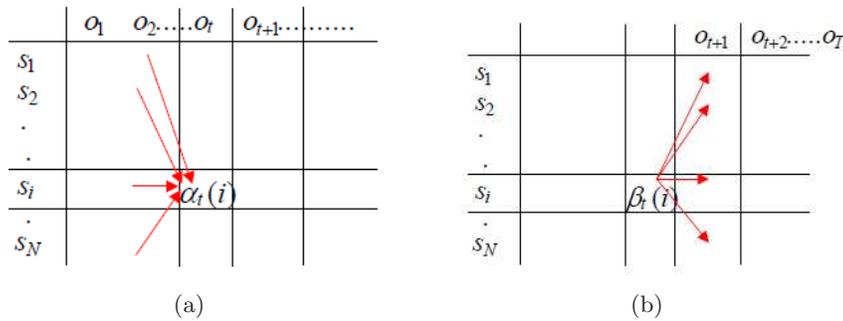


Figure 1.3. Calcul des variables Forward (a) et Backward (b)

On définit la variable Forward par : $\alpha_t(i) = P(O_1 O_2 \dots O_t, q_t = S_i | \lambda)$ correspondant à la probabilité d'être dans l'état S_i à l'instant t en ayant observé les premières observations $O_1 O_2 \dots O_t$. On a alors :

$$P(O|\lambda) = \sum_{i=1}^N P(O_1 O_2 \dots O_t, q_t = S_i | \lambda) = \sum_{i=1}^N \alpha_T(i)$$

La variable $\alpha_t(i)$ peut être calculée d'une manière récursive comme suit :

Algorithme 1: Algorithme Forward

Initialisation

$$\alpha_1(i) \leftarrow \pi_i b_i(o_1) \text{ avec } 1 \leq i \leq N$$

Induction

$$\alpha_{t+1}(j) \leftarrow \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(o_{t+1})$$

avec $1 \leq j \leq N$ et $1 \leq t \leq T - 1$

Terminaison

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$$

De la même manière, on peut définir la variable Backward par $\beta_t(i)$ par : $\beta_t(i) = P(O_{t+1}O_{t+2}\dots O_T, q_t = S_i|\lambda)$ correspondant à la probabilité d'être dans l'état S_i à l'instant t en ayant observé les observations de $t + 1$ jusqu'à la fin. La variable $\beta_t(i)$ peut être calculée d'une manière inductive comme suit :

Algorithme 2: Algorithme Backward

Initialisation

$$\beta_T(i) \leftarrow 1 \text{ avec } 1 \leq i \leq N$$

Induction

$$\beta_t(i) \leftarrow \left[\sum_{j=1}^N \beta_{t+1}(j) a_{ij} b_j(o_{t+1}) \right]$$

avec $1 \leq i \leq N$ et $T - 1 \leq t \leq 1$

Terminaison

$$P(O|\lambda) = \sum_{i=1}^N \pi_i b_i(o_1) \beta_1(i)$$

L'algorithme Forward-backward considère que l'observation peut se faire en deux temps : d'abord, émission du début de l'observation $O(1 : t)$ en aboutissant à l'état q_i au temps t , puis, émission de la fin de l'observation $O(t + 1 : T)$ sachant que l'on part de q_i au temps t . Le calcul de α se fait avec t croissant tandis que le calcul de β se fait avec t décroissant, d'où les expressions Forward-backward.

Dans ce cas, la probabilité d'observation est obtenue en prenant les valeurs de α et β à un instant t quelconque :

$$P(O|\lambda) = \sum_i \alpha_t(i) \beta_t(i)$$

Où $\alpha_t(i)$ est la probabilité d'émettre le début $O(1 : t)$ et d'aboutir à q_i à l'instant t , et $\beta_t(i)$ est la probabilité d'émettre la fin $O(t + 1 : T)$ sachant que l'on part de q_i à l'instant t .

Cependant, on utilise le plus souvent les valeurs obtenues pour deux cas particuliers ($t=0$) et ($t=T$), ce qui donne :

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i) = \sum_{i=1}^N \pi_i \beta_0(i)$$

Avec ces deux algorithmes, le nombre d'opérations se réduit à $N.(N + 1).(T - 1) + N$ multiplications et $N.(N - 1).(T - 1)$ additions soit une complexité en $O(N^2T)$. Ainsi, pour l'exemple précédent ($N = 5$ et $T = 100$), nous avons environ 3000 opérations à effectuer, contre 10^{72} opérations par la méthode directe, ce qui permet donc un gain très important en complexité de calcul.

1.6.2.2 La recherche du chemin le plus probable

La tâche essentielle d'un système utilisant les MMC est de déterminer le chemin correspondant à l'observation, c'est-à-dire de trouver, grâce au modèle, la meilleure suite d'états $Q = (q_1, q_2, \dots, q_T)$ qui maximise la vraisemblance d'une séquence d'observations $O = (O_1, O_2, \dots, O_T)$.

Pour cela, on définit la variable intermédiaire $\delta_t(i)$ comme étant la probabilité du meilleur chemin amenant à l'état s_i à l'instant t , en étant guidé par les t premières observations :

$$\delta_t(i) = \max_{q_1, \dots, q_{t-1}} P(q_1, q_2, \dots, q_t = s_i, O_1, O_2, \dots, O_t | \lambda)$$

par induction, on calcule

$$\delta_{t+1}(j) = \max_i (\delta_t(i) a_{ij}) b_j(o_{t+1})$$

en gardant trace, lors du calcul, de la suite d'états qui donne le meilleur chemin amenant à l'état s_i à t dans un tableau ψ .

Pour formaliser cette récurrence, on utilise l'algorithme de Viterbi [Fornéy 73, Viterbi 67] qui n'est qu'une variante de la fonction forward en transformant la somme en max et en mémorisant le chemin pris. C'est un algorithme de programmation dynamique, c'est à dire une méthode de

résolution de problèmes d'optimisation qui repose sur une propriété de décomposabilité de la fonction à optimiser.

La programmation dynamique, fondée sur le principe d'optimalité de Bellman, optimise le coût d'une trajectoire entre deux points. Si une trajectoire peut à chaque instant prendre une valeur parmi N , pour chaque point intermédiaire chaque sous-trajectoire est aussi optimale. Ainsi, le calcul direct d'une trajectoire de longueur L qui peut prendre N valeurs aurait une complexité en N^L , mais le calcul d'une demi-trajectoire est en $N^{L/2}$. En itérant L fois ce processus, on obtient une approche "diviser pour régner" dont le coût est en $L.N^2$.

L'algorithme de Viterbi est le suivant :

Algorithme 3: Algorithme de Viterbi

Initialisation

$$\delta_1(i) \leftarrow \pi_i b_i(o_1) \quad 1 \leq i \leq N$$

$$\psi_1(i) \leftarrow 0$$

où $\psi_t(i)$ est le tableau où l'on range la suite des états parcourus.

Récursion

$$\delta_t(j) \leftarrow \max_i (\delta_{t-1}(i) a_{ij}) b_j(o_t) \quad \text{avec } 1 \leq i \leq N \quad 2 \leq t \leq T$$

$$\psi_t(j) \leftarrow \operatorname{argmax}_i (\delta_{t-1}(i) a_{ij}) \quad 1 \leq j \leq N \quad \text{avec } 2 \leq t \leq T$$

Terminaison

$$P^* \leftarrow \max_i (\delta_T(i))$$

$$q_T^* \leftarrow \operatorname{argmax}_i (\delta_T(i))$$

Calcul du meilleur chemin (backtracking)

$$q_t^* \leftarrow \psi_{t+1}(q_{t+1}^*) \quad t = T-1, T-2, \dots, 1$$

1.6.2.3 Apprentissage du modèle MMC

Le but de l'apprentissage est de déterminer les probabilités de transitions et d'observations qui maximisent la probabilité de la suite d'observations $P(O|\lambda)$. L'idée de l'apprentissage est d'utiliser des procédures de ré-estimation qui affinent le modèle petit à petit suivant les étapes suivantes :

- Choisir un ensemble initial de paramètres λ_0 .
- Calculer λ_1 à partir de λ_0 , λ_1 étant meilleure que λ_0 au sens d'un critère de vraisemblance.
- Répéter ce processus jusqu'à un critère de fin.

Dans le cas des modèles de Markov où les états cachés sont connus, on réalise un comptage de l'utilisation des transitions (éléments des matrices A et π) et des émissions des différents symboles (éléments de la matrice B). La méthode de réestimation est la suivante :

Estimation de a_{ij} :

$$a_{ij} = \frac{\text{Nbre de fois où la transition de } E_i \text{ vers } E_j \text{ est utilisée}}{\text{Nbre de transitions effectuées à partir de } E_i}$$

Estimation de $b_j(o_t)$:

$$b_j(o_t) = \frac{\text{Nbre de fois où le modèle s'est trouvé dans l'état } E_j \text{ en observant } v_k}{\text{Nbre de fois où le modèle s'est trouvé dans l'état } E_j}$$

Estimation de π_i :

$$\pi_i = \frac{\text{Nbre de fois où le modèle s'est trouvé dans l'état } E_i \text{ en émettant le 1er symbole}}{\text{Nbre de fois où le modèle a émis le premier symbole d'une séquence}}$$

On peut aisément déduire que cette méthode de réestimation conserve les propriétés indispensables suivantes :

$$\sum_{j=1}^N a_{ij} = 1 \quad \sum_{k=1}^M b_j(k) = 1 \quad \sum_{i=1}^N \pi_i = 1$$

Dans le cas des modèles de Markov où les états cachés sont inconnus au problème d'apprentissage, on dit que les séquences d'états sont des données manquantes. En effet, pour chaque séquence d'observation, la séquence d'états correspondante est cachée. Il s'agit donc d'un problème d'apprentissage à données manquantes. Aucune solution analytique n'existe pour l'estimation des paramètres du modèle à partir de données manquantes. L'algorithme de Baum-Welch [Baum 70] est le plus utilisé pour ce genre de problème. Son principe consiste à appliquer l'algorithme EM (expectation maximisation) aux modèles de Markov [Dempster 77]. Il permet d'utiliser des outils pour l'estimation des probabilités. On définit pour cela les fonctions ξ_t et γ_t .

$\xi_t(i, j)$ représente la probabilité de se trouver dans l'état E_i au temps t et dans l'état E_j au temps $t + 1$, pour un modèle donné et une séquence d'observation donnée.

$$\xi_t(i, j) = P(q_t = E_i, q_{t+1} = E_j | O, \lambda)$$

Par définition des fonctions Forward et Backward, on peut écrire $\xi_t(i, j)$ sous la forme :

$$\xi_t(i, j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)} = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}$$

γ_t représente la probabilité de se trouver dans l'état i au temps t pour un modèle et une séquence d'observations donnés. Par définition de $\xi_t(i, j)$, on peut donc écrire :

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j)$$

les formules de réestimation peuvent se réécrire en fonction de $\xi_t(i, j)$ et $\gamma_t(i)$:

Estimation de π_i :

$$\pi_i = \gamma_1(i) \quad 1 \leq i \leq N$$

Estimation de a_{ij} :

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

Estimation de $b_j(k)$:

$$b_j(k) = \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \nu_{nj}$$

Il a été établi par Baum et ses collaborateurs, qu'en utilisant ces formules de réestimation, si l'on définit un modèle courant $\lambda = \{A, B, \pi\}$ que l'on utilise pour déterminer un nouveau modèle $\lambda^* = \{A^*, B^*, \pi^*\}$, alors :

- Soit le modèle λ définit un point critique de la fonction de vraisemblance, auquel cas : $\lambda = \lambda^*$
- soit λ^* est tel que : $P(O|\lambda^*) > P(O|\lambda)$

L'algorithme converge donc vers un optimum local.

Dans le cas des HMMs continus, à la place de la matrice d'émission B , on a besoin de réestimer les paramètres de la densité de probabilité d'émission des observations pour chaque état. La loi paramétrique la plus utilisée dans le cas des HMMs est le mélange de gaussiennes. Dans ce cas, les paramètres à réestimer sont le vecteur moyen μ_{jm} , la matrice de covariance σ_{jm} et le

coefficient de pondération C_{jm} , qui est associé à la m^{me} composante de la mixture, où j désigne un état. Ces paramètres sont calculés comme suit :

$$C_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k)}{\sum_{t=1}^T \sum_{k=1}^M \gamma_t(j, k)}$$

$$\mu_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k) O_t}{\sum_{t=1}^T \gamma_t(j, k)}$$

$$\sigma_{jk} = \frac{\sum_{t=1}^T \gamma_t(j, k) \cdot (O_t - \mu_{jk})(O_t - \mu_{jk})'}{\sum_{t=1}^T \gamma_t(j, k)}$$

Remarques

1. **Apprentissage Viterbi** : Il existe une variante de cette méthode d'apprentissage, dite apprentissage par la méthode de Viterbi, qui réalise le comptage par l'algorithme de Viterbi plutôt que par les fonctions Backward-Forward. Dans ce cas, on ne s'intéresse plus à tous les chemins ayant généré une observation, mais seulement au plus probable.
2. **Apprentissage embarqué** : Appliquer un apprentissage simple tel qu'il est décrit ci-dessus à un problème de reconnaissance d'écriture, implique que l'on dispose de bases d'apprentissage distinctes pour les différents modèles de caractères. Il faut donc passer par une fastidieuse phase de segmentation-étiquetage de toutes les images de mots qui puisse nous fournir des images de lettres pour entraîner leurs modèles. Il serait donc intéressant d'exploiter la capacité des HMMs à s'aligner sur des observations correspondant à des mots complets sur des modèles de mots, pour finalement répercuter les informations sur les modèles de caractères. C'est l'objet de l'apprentissage dit embarqué (embedded training) des modèles de Markov cachés dont l'algorithme complet est le suivant :

L'algorithme converge vers l'apprentissage d'un modèle "moyen" de caractère qui tient compte du bruit et des situations ambiguës.

Algorithme 4: Algorithme d'apprentissage embarqué

Tantque vraisemblance augmente faire
début
 pour chaque mot de la base d'apprentissage faire
début
 – construire le MMC du mot en concaténant les modèles caractères
 – calculer les variables forward et backward
 – cumuler la vraisemblance de l'exemple
 – mettre à jour les variables ξ et γ impliquées par les caractères du modèle
fin
 ré-estimer les modèles de chaque caractère
fin

1.6.3 Modèles pseudo bidimensionnels : PHMM ou Planar HMM

L'utilisation des modèles de Markov 1D en reconnaissance automatique de l'écriture a permis d'obtenir des résultats intéressants pour certaines applications. Mais la nature 2D de l'écriture permet de penser que des améliorations importantes pourraient être apportées en étendant ces modèles aux images. Les PHMMs ou Planar HMMs ont été proposés initialement par Levin [Levin 92] comme solution intermédiaire entre les approches 2D conduisant à une complexité excessive et les approches 1D occultant les propriétés planaires de l'écriture. L'étude des PHMMs a par la suite été affinée dans [Agazi 93] et [Kuo 94].

Un PHMM est un HMM où la probabilité d'observation dans chaque état est donnée par un HMM secondaire. L'idée de base est de partager l'image en zones homogènes et de modéliser chacune d'entre elles par un HMM 1D, qu'il est convenu d'appeler modèle secondaire. Les états du modèles principal sont appelés super-états (voir figure 1.4).

Un PHMM donné par $\eta = (A, \pi, \Lambda)$ est défini dans [Bel, 97] de la manière suivante :

- $S = \{s_1, s_2, \dots, s_N\}$ l'ensemble des N super-états du modèle. On désigne un super-état pour la ligne y par $q_y \in S$.
- $V = \{v_1, v_2, \dots, v_M\}$ l'ensemble discret des M symboles. On désigne un symbole à la position (x, y) par $o_{xy} \in V$.
- $A = \{a_{i,j}\}_{1 \leq i,j \leq N}$, où $a_{i,j} = P(q_{y+1} = s_j | q_y = s_i)$, la matrice des

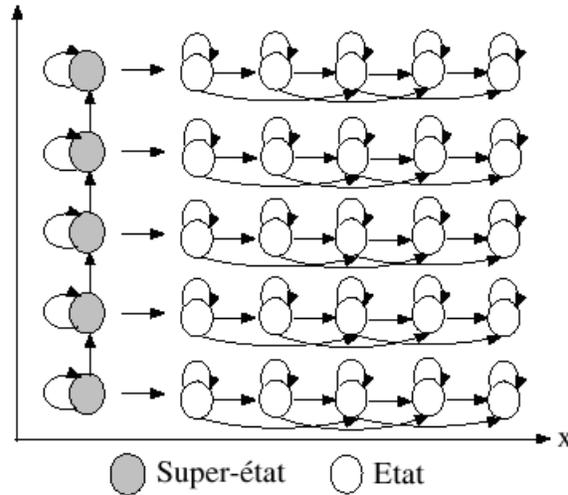


Figure 1.4. Exemple d'architecture d'un PHMM, d'après [Belaid 97].

transitions entre super-états.

- $\pi = \{\pi_i\}_{1 \leq i \leq N}$, où $\pi_i = P(q_1 = s_i)$. π est le vecteur des probabilités initiales des super-états.
- $\Lambda = \{\lambda^k\}_{1 \leq k \leq N}$, l'ensemble des HMMs secondaires associés aux super-états. $\lambda^k = \{A^k, B^k, \pi^k\}$, avec :
 - $S^k = \{s_1^k, s_2^k, \dots, s_{N^k}^k\}$ l'ensemble des N^k états du modèle. L'état localisé en (x, y) est noté par $q_{xy} \in S^k$.
 - $A^k = \{a_{i,j}^k\}_{1 \leq i, j \leq N^k}$, où $a_{i,j}^k = P(q_{x+1y} = s_j^k | q_{xy} = s_i^k)$, représente la matrice de transitions entre états.
 - $B^k = \{b_j^k(l)\}_{1 \leq j \leq N^k; 1 \leq l \leq M}$, où $b_j^k(l) = P(o_{xy} = v_l | q_{xy} = s_j^k)$.
 - $\pi^k = \{\pi_i^k\}_{1 \leq i \leq N^k}$, où $\pi_i^k = P(q_{1y} = s_i^k)$.

En reconnaissance de l'écriture, le modèle secondaire est souvent associé aux lignes où la forme est réellement observée, leurs architectures sont typiquement gauche-droite. Généralement, plusieurs lignes sont associées à chaque super-état. Le nombre de super-états dépend de la morphologie de

la forme et des principales zones horizontales d'observation que l'on veut mettre en évidence.

Procédure d'apprentissage des PHMMs

Les PHMMs étant à la base constitués de HMMs, l'estimation des paramètres des modèles secondaires est effectuée, le plus souvent par l'algorithme de Baum-Welch. Les paramètres du modèle principal sont généralement estimés par comptage.

Procédure de reconnaissance des PHMMs

Pour un modèle M et une observation O , la reconnaissance se fait par la suite en déterminant la séquence de super-états maximisant $P(Q|O, M)$. Cette procédure de reconnaissance se fait par un algorithme de Viterbi doublement intégré [Kuo 94] : un premier Viterbi sert à déterminer le meilleur chemin sur une ligne. Le deuxième Viterbi permet de calculer la probabilité d'associer une ligne à un super-état.

Les PHMMs présentent des propriétés d'élasticité 2D permettant un recalage des informations en entrée. Cependant, ils nécessitent une hypothèse d'indépendance entre colonnes qui n'est pas toujours vraie en pratique.

Une modélisation bidimensionnelle de l'image à l'aide des champs de Markov a été proposée pour résoudre ce problème. Ces modèles, définissant des contraintes bidimensionnelles, permettent de modéliser de manière beaucoup plus réaliste les interactions 2D de l'écriture.

1.6.4 Modèles bidimensionnels : champs de Markov

Contrairement aux PHMMs, les champs de Markov possèdent une véritable structure 2D dans la mesure où la probabilité d'un site du champ est conditionnée par les sites voisins et conditionne à son tour celles d'autres sites.

Un champ de Markov caché noté (X, Y) , est un processus doublement stochastique formé de deux champs aléatoires X et Y dont les variables sont indexées par un ensemble S de sites ou positions s dans l'image noté $S = \{s\}$. Dans le domaine de l'analyse d'image, les sites sont généralement les éléments sur une grille ou un maillage bidimensionnel appliqué à l'image. Le champ Y est le champ des observations et se note $Y = (Y_s)_{s \in S}$. Une

réalisation de ce champ est notée $Y = y$ et la valeur du champ au site s pour une réalisation donnée y se note $Y_s = y_s$. Les variables aléatoires Y_s de ce champ correspondent à des observations ou mesures effectuées sur l'image.

Le champ X est le champ caché des étiquettes et se note $X = (X_s)_{s \in S}$. Une réalisation de ce champ est notée $X = x$ et la valeur du champ au site s pour une réalisation donnée x se note $X_s = x_s$. Les variables aléatoires X_s de ce champ prennent leur valeur dans un ensemble fini $L = \{l_1, l_2, \dots, l_q\}$ de $q = |L|$ étiquettes discrètes. Ces étiquettes désignent les entités de la structure sous-jacente cachée. L'ensemble des réalisations possibles du champ X se note $\Omega = L^{|S|}$ où $|S|$ désigne le cardinal de l'ensemble S , c'est à dire le nombre total de sites de l'image.

La définition d'un champ aléatoire de Markov (Markov Random Field ou MRF) s'énonce de la manière suivante :

Un champ X de variables aléatoires X_s indexées par un ensemble de sites $S = s$ dont les relations sont définies selon un système de voisinage V , est un champ markovien si et seulement si il vérifie deux propriétés : la propriété de positivité et la propriété de dépendance Markovienne entre les variables.

– propriété de positivité :

$$\forall x \in q^{|S|} P(X = x) \geq 0$$

Cela signifie que quelle que soit la réalisation du champ X , sa probabilité est positive.

– propriété de Markov :

$$\forall s \in S \text{ et } \forall x \in q^{|S|} \\ P(X_s = x_s | X_r = x_r, r \in S - s) = P(X_s = x_s | X_r = x_r, r \in V_s)$$

Cette propriété implique que l'état du champ en tout site ne dépend que des états du champ sur les sites voisins. La figure 1.5 illustre quelques systèmes de voisinage de différents ordres ainsi que les

cliques associées à chacun.

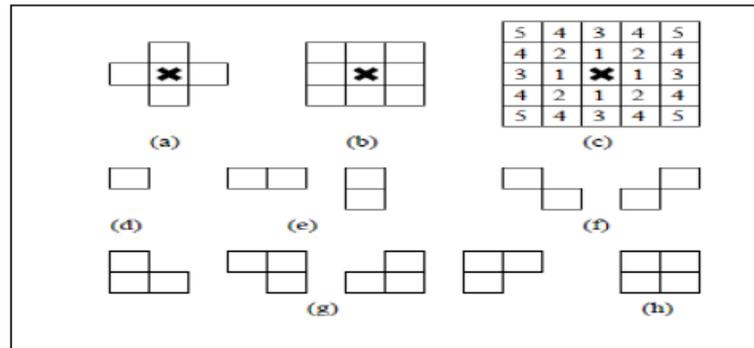


Figure 1.5. (a) Système de voisinage d'ordre 1, (b) système de voisinage d'ordre 2, (c) hiérarchie de systèmes de voisinages, (b) et (c) cliques associées au système d'ordre 1, (d)+(e)+(f)+(g)+(h) cliques associées au système d'ordre 2, d'après [Saon 97].

On distingue deux types de champs de Markov : les champs causaux sur lesquels il est possible d'induire une causalité en restreignant les voisinages à ceux permettant une progression naturelle du calcul de la probabilité du champ, et les champs non causaux. La plupart des champs de Markov utilisés en reconnaissance de l'écriture manuscrite font une hypothèse de causalité introduisant un sens de parcours. Les modèles les plus utilisés sont les réseaux de Markov (Markov Random Mesh ou MRM) [Park 96a] et les champs de Markov unilatéraux (Non-Symmetric Half-Plane Markov Chains ou NSHP).

Dans la suite, nous présentons en détail un modèle spécial baptisé NSHP-HMM [Saon 97] qui a été appliqué à la reconnaissance de l'écriture manuscrite.

1.6.4.1 Le modèle NSHP-HMM

Le NSHP-HMM est un modèle hybridant un champ de Markov observable et un HMM. Les champs de Markov fournissent les probabilités d'observation dans les états du HMM. L'idée est de combiner ces deux modèles afin d'ajouter une souplesse horizontale aux champs de Markov, et de lier statiquement les probabilités d'observations dans les états des HMMs grâce au champ Markovien. Ce dernier point se justifie par l'hypothèse markovienne classique, qui suppose que les probabilités d'observations sont indépendantes

entre états : l'utilisation d'un champ de Markov permet de lier les états entre eux grâce à son voisinage 2D, sans altérer cette hypothèse.

Pour décrire d'une manière plus formelle un NSHP-HMM, on commence par définir les chaînes de Markov à demi plan non symétrique (NSHP pour Non Symetric Half Plane) [Choisy 02].

Définition d'un NSHP

Étant donné un point X_{ij} d'une image X , le demi plan non symétrique (NSHP) Σ_{ij} du point X_{ij} , et un voisinage local Θ_{ij} , nous avons :

$$P(X_{ij}|X_{\Sigma_{ij}}) = P(X_{ij}|X_{\Theta_{ij}})$$

Cela peut se traduire du fait de l'hypothèse Markovienne bi-dimensionnelle : la probabilité d'un pixel connaissant son contexte global peut se ramener à sa probabilité connaissant un contexte local. La figure 1.6 illustre ce principe.

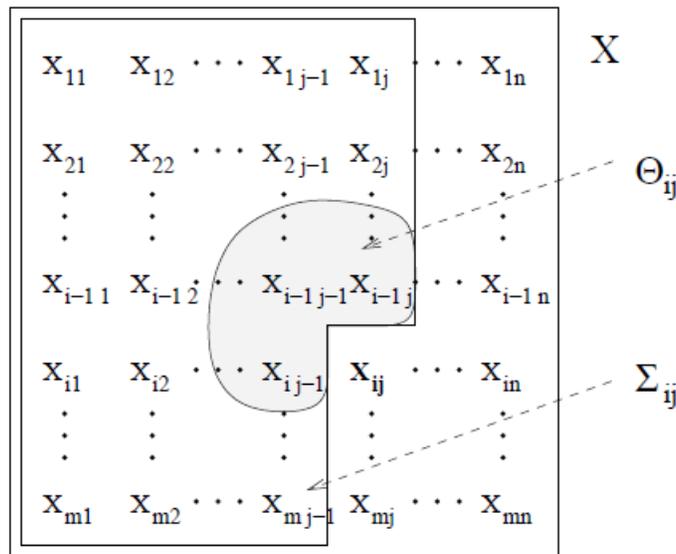


Figure 1.6. Ensemble de sites associés à un NSHP

Le calcul de la probabilité de masse cumulée d'une image X sachant un

NSHP a la propriété suivante :

$$P(X) = \prod_{(i,j \in X)} (X_{ij} | X_{\Sigma_{ij}}) = \prod_{(i,j \in X)} P(X_{ij} | X_{\theta_{ij}})$$

Ce calcul peut se décomposer par colonne, ce qui donne :

$$P(X) = \prod_{j=1}^n \prod_{i=1}^m P(X_{ij} | X_{\theta_{ij}})$$

L'idée originale proposée par Saon est d'associer les distributions statistiques des colonnes du NSHP aux états d'un HMM, d'où le nom NSHP-HMM du modèle pour Non Symetric Half Plane Hidden Markov Model. Une transition d'un état du modèle à un autre aura pour effet le changement de l'ensemble des distributions et, par conséquent, l'adaptabilité du modèle aux différentes propriétés de l'image. La Figure 1.7 montre un exemple d'application d'un tel modèle sur une image de lettre.

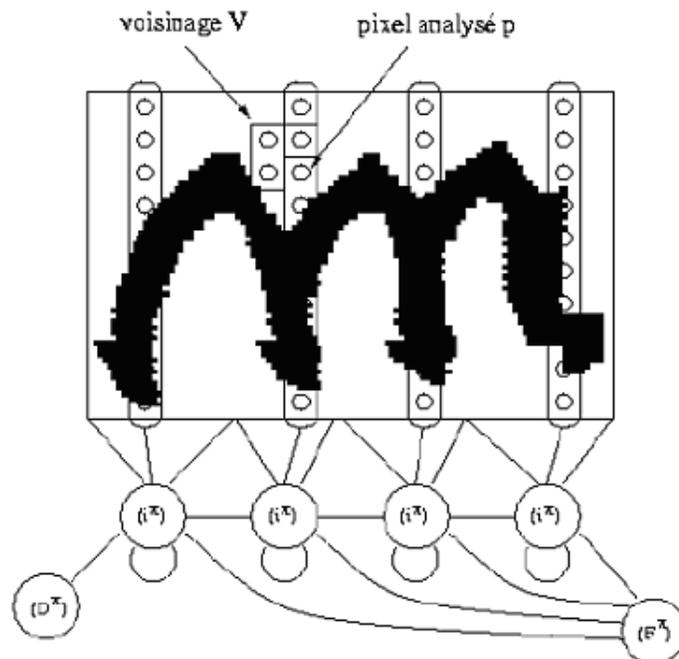


Figure 1.7. Exemple de NSHP-HMM appliqué sur une lettre d'après [Saon 97].

A partir de cela, un NSHP-HMM peut être défini comme suit :

Définition NSHP-HMM

Un NSHP-HMM d'ordre P est défini par :

- $V = \{0, 1\}$ où $\{\text{blanc}, \text{noir}\}$ l'ensemble du vocabulaire observable.
- $S = \{s_1, \dots, s_N, D, F\}$ l'ensemble des N états normaux du HMM sous-jacent, plus les deux états spécifiques.
- $A = \{a_{ij} \cup \{a_{Di}, a_{iF}\}\}_{1 \leq i, j \leq N}$ où $a_{i,j} = P(q_{t+1} = s_j | q_t = s_i)$, $a_{D,i} = P(q_1 = s_i | D)$, $a_{i,F} = P(F | q_T = s_i)$
- $B = \{b_i(y, \theta, C)\}$ où $s_i \in S$, $s_i \neq D, F$, la probabilité d'observation pour l'état i d'un pixel de couleur c , à la hauteur y , pour une configuration de voisinage donnée θ .

En conclusion, les paramètres déterminants du NSHP-HMM sont la hauteur des colonnes analysées (hauteur du modèle), la taille du voisinage (ordre du modèle), le nombre d'états du HMM sous-jacents et la structure du HMM (transitions autorisées).

Nous avons présenté à ce niveau les modèles markoviens avec leur différentes dimensionnalités. Afin de solliciter de systèmes plus robustes, des modèles couplant des HMMs à un ou plusieurs réseaux de neurones ont été proposés. Nous présentons le principe de ces approches hybrides dans ce qui suit.

1.7 Les modèles hybrides : MMC et réseaux de neurones

Les modèles hybrides désignent les modèles couplant un ou plusieurs réseaux de neurones à un MMC. Ce couplage est avantageux par rapport à l'usage classique d'un MMC ou d'un réseau de neurone pur, pour les tâches de reconnaissance de formes complexes comme c'est le cas de la reconnaissance de l'écriture manuscrite sans contrainte. En effet, les MMC sont appropriés à l'intégration du contexte et leur apport pour la segmentation et la modélisation de séquences. Cependant, la condition d'indépendance imposée par le modèle de Markov (chaque observation doit être indépendante des observations voisines) rend les MMC incapables de tirer avantage de la corrélation qui existe réellement dans les observations en écrit. De plus, les MMC sont souvent entraînés selon le critère MLE qui ne permet pas de réa-

liser une bonne discrimination lorsqu’il s’agit de classer des formes.

D’un autre côté, les réseaux de neurones ont un grand pouvoir discriminant à travers leur capacité à construire des frontières de décisions complexes dans des espaces de grandes dimensions. Cependant, ils ne peuvent modéliser une évolution temporelle à long terme. Ils sont donc mal adaptés au problème de traitement des séquences telles que l’écriture.

Une combinaison des deux systèmes semble donc appropriée afin de tirer parti des avantages de chacun. Les réseaux de neurones sont utilisés en tant qu’estimateurs des probabilités d’émission pour les modèles de Markov qui sont un excellent outil pour segmenter une séquence.

En reconnaissance d’écriture, nous distinguons principalement deux stratégies pour le couplage d’un MMC à des réseaux de neurones :

- Les réseaux de neurones en amonts d’un MMC : dans ce cas, le réseau de neurones est utilisé en tant qu’estimateur local (estimation des probabilités d’observations) et le HMM est utilisé dans son formalisme habituel. De nombreux systèmes ont été développés à partir de ce principe [Schenkel 95, Garcia-Salicitti 96, Cho 97, Senior 98]
- Les réseaux de neurones en aval d’un MMC : dans ce cas, le réseau de neurone est utilisé comme post-traiteur des sorties d’un MMC. Différents cas d’utilisation ont été présentés dans la littérature [Bengio 95, Choisy 02, Koerich 02, Morita 03, Koerich 05].

1.8 Modèles de Markov multi-flux où “multi-stream”

Les modèles de markov cachés (MMC) sont particulièrement appropriés pour modéliser le signal mono-dimensionnel. En revanche l’application de ces mêmes modèles à la reconnaissance d’images d’écriture n’est pas aussi naturelle puisqu’elle oblige à transformer l’information d’entrée bi-dimensionnelle en une représentation mono-dimensionnelle.

De nombreux travaux ont porté sur des extensions de modèles de type Markovien afin de modéliser plus finement la nature bi-dimensionnelle de l’information. Nous avons déjà évoqué les modèles de Markov 2D mais d’autres extensions peuvent être envisagées dans le cadre de la fusion d’informations.

Parmi ces extensions, les modèles de Markov multi-stream ont été ini-

tialement proposés dans le domaine de la reconnaissance automatique de la parole par [Varga 90] pour décomposer le signal audio en signal de parole et bruit. Conçus pour modéliser simultanément plusieurs flux d'information relatifs à un seul processus, ces modèles ont été détournés dans [Bourlard 96] afin de combiner le son de la voix et la vidéo des lèvres en mouvement dans le cadre de la reconnaissance de la parole continue. Ces modèles ont été utilisés récemment par [McCowan 05] dans le cadre de l'identification du locuteur dans les « meetings ».

Le principe de ces modèles MMC multi-flux est de traiter plusieurs sources (flux) d'informations de manière partiellement asynchrone. Les modèles des différents flux sont traités indépendamment jusqu'à certains points d'ancrage appelés états de recombinaison où ils sont contraints à se resynchroniser et à combiner leurs contributions partielles. Les avantages potentiels de ces approches sont multiples [Bourlard 96] :

- Elles offrent un moyen pour combiner différentes sources d'information.
- Cette combinaison peut être adaptative, certaines sources d'informations pouvant être sous pondérées, voir rejetées si elles sont identifiées comme très peu fiables.
- La topologie des modèles de Markov cachés peut être adaptée à chaque source d'information.
- Les différents flux peuvent se désynchroniser jusqu'à certains points lexicaux prédéfinis.

Malgré toutes ces possibilités, l'utilisation des modèles multi-flux dans le domaine de la reconnaissance de l'écriture est restée très limitée. Nous présentons dans le chapitre 2 quelques applications de ces modèles dans le cadre de la reconnaissance de l'écriture.

1.8.1 Formalisme multi-flux

L'approche multi-flux initialement proposée dans [Bourlard 97, Wellekens 98] est une méthode adaptative permettant de combiner différentes sources d'information en utilisant des modèles de Markov coopératifs. Les modèles des différentes sources sont traités indépendamment jusqu'à

certains points d’ancrage¹ (\otimes sur la Figure 1.8) où ils sont contraints à se resynchroniser et à combiner leurs contributions partielles.

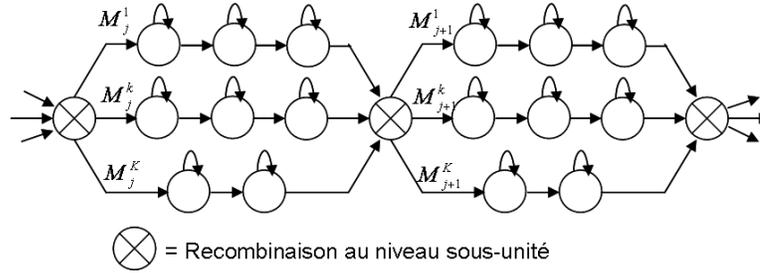


Figure 1.8. Structure générale d’un modèle multi-flux

Soit K le nombre de sources d’informations et soit M le modèle composé d’une séquence de J sous-modèles qui correspondent à des sous-unité lexicales $M_j (j = 1, \dots, J)$ (des lettres par exemple). Chaque sous-modèle M_j est composé de K modèles de Markov cachés (MMC) notés M_j^k . Dans le cadre des modèles multi-flux, les K modèles des K sources d’information d’une même sous-unité sont considérés indépendants entre eux. En revanche la modélisation multi-flux introduit une dépendance entre les sources d’information en imposant un synchronisme entre les K sources lors des transitions entre sous-unités. Ainsi, si X désigne la séquence d’observations multi-flux, celle-ci peut se décomposer en J sous-séquences associées chacune au sous-modèle M_j et composée chacune de K sources d’information synchronisées lors des transitions entre sous-unités. On note alors :

$$X = \left\{ \left(X_1^k, X_2^k, \dots, X_j^k, \dots, X_J^k \right), k = 1 \dots K \right\} \quad (1.1)$$

où X_j^k désigne la séquence d’observation de la source k correspondant au sous modèle j . Si C_j désigne la séquence d’états multi-flux associée à la séquence d’observations multi-flux, alors par définition du modèle multi-flux, la vraisemblance $P(X_j, C_j | M_j)$ est calculée sur la base des vraisemblances associées à chacune des sources d’information selon la relation suivante :

$$P(X_j, C_j | M_j) = f \left(\left\{ P(X_j^k, C_j^k | M_j^k), k = 1, \dots, K \right\} \right) \quad (1.2)$$

1. États de combinaison correspondants aux frontières des caractères ou des syllabes par exemple.

Par exemple, pour une fonction de combinaison f linéaire, la vraisemblance de la sous-séquence X_j par rapport au modèle de sous-unité M_j et pour le chemin C_j s'écrit :

$$\log P(X_j, C_j | M_j) = \sum_{k=1}^K \alpha_j^k \log P(X_j^k, C_j^k | M_j^k) \quad (1.3)$$

où α_j^k représente la fiabilité du flux k . On déduit alors la vraisemblance de la séquence X en sommant sur toutes les sous-unités.

$$\log P(X|M) = \sum_{j=1}^J \sum_{k=1}^K \alpha_j^k \log P(X_j^k, C_j^k | M_j^k) \quad (1.4)$$

1.8.2 Stratégies de combinaison

Comme nous l'avons déjà évoqué, le formalisme multi-flux consiste à combiner des décisions locales en certains points correspondant aux états de recombinaison. En pratique différentes stratégies de combinaison sont possibles. On peut distinguer dans ce cadre des règles de combinaison linéaires comme la règle de la somme et du produit et d'autres règles de combinaison non linéaire utilisant généralement un perceptron multi-couche (MLP). Une description détaillée des différentes stratégies est donnée dans [Hagen 01].

Certaines de ces stratégies de combinaison nécessitent une estimation et/ou une optimisation des poids relatifs à chaque flux appelé aussi fiabilité du flux. Grâce à ces poids, certaines sources d'informations peuvent être sous pondérées, voire rejetées si elles sont identifiées comme très peu fiables. Généralement ces poids peuvent dépendre de l'index du flux, de la sous-unité lexicale correspondante (caractère, syllabe . . .), des données d'observation ou d'une combinaison de ces trois derniers.

Différentes stratégies de pondération ont été proposées. Dans [Hagen 01] l'auteur distingue deux catégories, les stratégies à poids fixes où les poids sont estimés lors de l'apprentissage et les stratégies à poids adaptatifs dont les poids sont adaptés pendant la phase de reconnaissance. Dans le cadre de ce travail, nous avons testé deux stratégies de pondération que nous détaillons dans la section 1.8.4.

1.8.3 Aspects de décodage multi-flux

Le processus de décodage des modèles multi-flux revient à calculer la vraisemblance du meilleur chemin de la séquence de vecteurs d’observations des différentes sources d’informations. Il s’agit donc d’un décodage selon le principe du calcul du meilleur chemin, semblable à l’algorithme de Viterbi [Forney 73]. Pour ce faire, différentes solutions sont envisageables :

- La combinaison au niveau état : bien qu’elle ne permette pas l’asynchronie entre les différents flux, l’implémentation de cette méthode est relativement simple et revient à appliquer le décodage standard de Viterbi pour lequel les probabilités locales sont obtenues en combinant les probabilités relatives de chaque flux à tous les instants. Cette stratégie n’est valable que pour les modèles multi-flux synchrones.
- La combinaison au niveau sous-unité lexicale : cette méthode de combinaison permet de contrôler le degré d’asynchronie entre les différents flux de données. Le décodage nécessite dans ce cas une adaptation sérieuse de l’algorithme de décodage classique. Dans la littérature, deux algorithmes ont été proposés pour résoudre ce problème :
 1. Algorithme de programmation dynamique sur deux niveaux, dit “Two-Level” [H.Sakoe 79] : le décodage est séparé en deux niveaux imposant deux phases de traitement. Un premier processus de programmation dynamique opère au niveau des sous-unités et permet de déterminer un score pour chaque sous-modèle pour toutes les positions possibles de début et de fin de trames de données. Un deuxième processus de programmation dynamique détermine, en se servant des résultats du premier processus, la meilleure séquence de sous unités.
 2. “HMM-recombination” [Bourlard 96] : cet algorithme est une adaptation de l’algorithme “HMM-decomposition” [Varga 90] utilisé dans le domaine de la reconnaissance de la parole permettant de décomposer le signal audio en deux composantes indépendantes (son et bruit). C’est cet algorithme que nous utilisons dans notre système de reconnaissance multi-flux et dont nous détaillons le principe dans la suite.

1.8.3.1 Algorithme HMM-recombination

Le principe de l'algorithme est de construire un MMC composite à partir des MMC parallèles du modèle multi-flux. Le MMC composite est considéré comme le produit des MMC associé à chaque flux de données. Chaque état du MMC composite correspond à un K-uple d'états des K modèles coopératifs. La topologie de ces modèles composites possède donc K dimensions et est définie de façon à pouvoir représenter tous les schémas possibles étant données les topologies des modèles coopératifs initiaux. Les contributions locales d'un état composite sont définies comme une combinaison des contributions locales des états intervenant dans cet état composite. La combinaison peut donc se faire à chaque trame et le décodage est tout à fait classique en opérant selon l'algorithme de Viterbi.

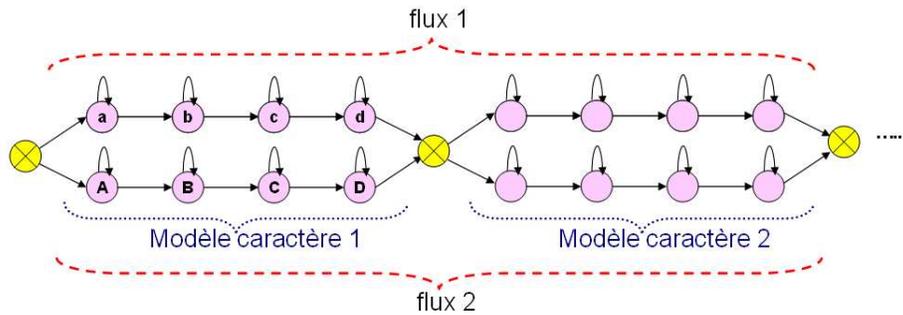


Figure 1.9. Topologie d'un modèle parallèle à 2 flux.

Dans le cas de deux modèles coopératifs de topologies identiques comme représenté à la figure 1.9, on construit le MMC composite symétrique présenté par la figure 1.10. Les paramètres du MMC produit résultant sont déterminés à partir des MMC mono-flux comme suit :

La log-vraisemblance conditionnelle des données pour le méta-état i est donnée par :

$$\log P(X_1(t), X_2(t)|i) = \alpha_i^1 \log P(X_i^1(t)|S_1(i)) + \alpha_i^2 \log P(X_i^2(t)|S_2(i))$$

$X_1(t)$ (resp. $X_2(t)$) est le vecteur de données sur le premier flux (resp. deuxième flux) à t et $S_1(i)$ (resp. $S_2(i)$) est une fonction qui désigne l'état correspondant à l'état produit dans le MMC du flux 1 (resp. flux 2).

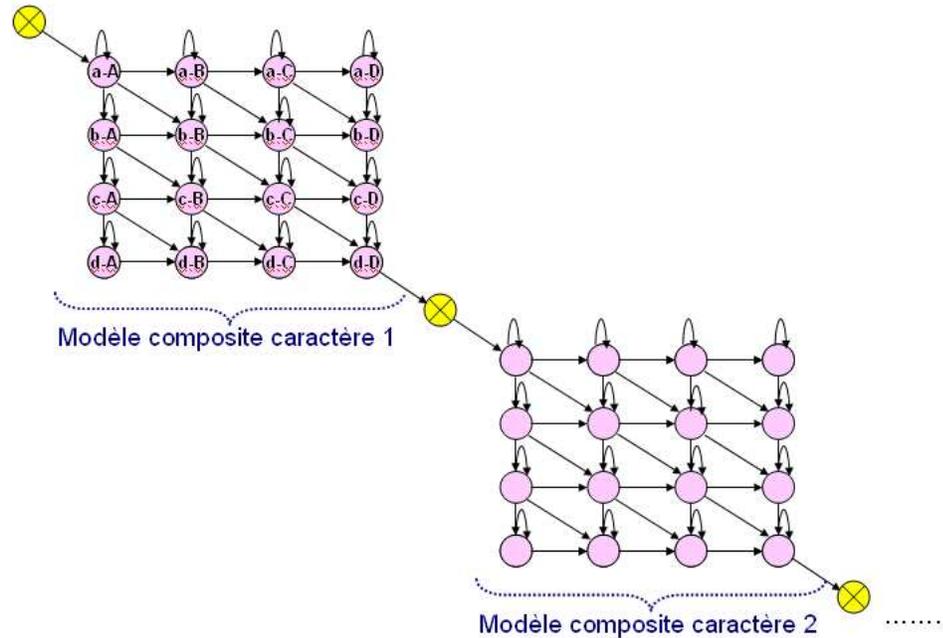


Figure 1.10. Topologie du modèle MMC résultant du modèle à 2 flux de la Figure 1.9.

Les probabilités de transitions de chaque état du modèle produit sont déterminées à partir des probabilités de transitions des états correspondants dans le modèle parallèle. Par exemple, la transition de l'état $a - A$ vers l'état $a - B$ dans le MMC composite correspond à une transition de l'état a vers lui-même sur le MMC du flux 1 dans le modèle parallèle et de l'état A vers l'état B dans celui du MMC du deuxième flux.

Par exemple, la probabilité de transition $P(a - A \rightarrow a - B)$ se calcule alors comme étant le produit des probabilités de transition $P(a \rightarrow a)$ et $P(A \rightarrow B)$ en faisant l'hypothèse d'indépendance.

Notons que sur la Figure 1.10, les mouvements gauche-droite correspondent à des transitions dans la chaîne A, B, C, D , les mouvements haut-bas correspondent à des transitions dans la chaîne a, b, c, d , et les mouvements diagonaux à des transitions simultanées pour les deux canaux.

1.8.4 Apprentissage

Les paramètres des MMC multi-flux peuvent être estimés soit indépendamment soit conjointement. Dans la première approche, les paramètres des

MMC sont estimés indépendamment dans chaque flux avant d'effectuer le produit. Dans ce cas, l'apprentissage des modèles multi-flux se fait en deux étapes : Apprendre les paramètres des MMC des différents flux (probabilités de transitions, moyenne, variance et poids des gaussiennes) et estimer leurs poids de combinaison.

La deuxième approche consiste à profiter de la représentation sous forme de MMC produit pour estimer conjointement les paramètres du modèle.

Dans les deux cas, l'apprentissage des paramètres des MMC d'un modèle N-flux est réalisé grâce aux techniques classiques d'apprentissage des MMC décrites dans la section 1.6.2.3.

D'autres techniques sont proposées pour l'apprentissage des poids de combinaison que nous présentons dans ce qui suit.

Apprentissage des poids de combinaison

Différentes stratégies de pondération ont été proposées dans [Hagen 01] dans le cadre de la reconnaissance de la parole par des modèles multi-flux. L'auteur distingue deux catégories, les stratégies à poids fixes où les poids sont estimés lors de l'apprentissage et les stratégies à poids adaptatifs dont les poids sont adaptés pendant la phase de reconnaissance.

Notons que dans le cadre de l'approche multi-flux, dans le cas d'une fonction de combinaison linéaire en logarithme de vraisemblance (c'est-à-dire par moyenne géométrique pondérée de vraisemblances), l'estimation par maximisation de vraisemblances des poids de combinaison échoue. En effet, la vraisemblance globale des données d'entraînement s'écrit :

$$\begin{aligned} P(X, C|M) &= \prod_{k=1}^K \prod_{j=1}^J P(X^k, C_j^k | M_j^k)^{\alpha^k} \\ &= \prod_{k=1}^K \prod_{j=1}^J (P_k)^{\alpha^k} \end{aligned}$$

Maximiser $P(X, C|M)$ sous la contrainte $\sum_k \alpha^k = 1$ conduit à la solution suivante :

$$\alpha^k = \begin{cases} 1 & \text{si } k = \operatorname{argmax}_k P_k \\ 0 & \text{sinon} \end{cases}$$

Dans [Potamianos 98], les auteurs introduisent des contraintes supplé-

mentaires pour remédier à ce problème.

Nous présentons deux stratégies qui peuvent être appliquées dans le cadre de ce travail.

- Approche de combinaison par poids égaux :

Cette stratégie consiste à attribuer des poids égaux pour les différents flux. Les poids sont indépendants des experts et fixés a priori à :

$$\alpha_k = cte = \frac{1}{K}$$

K étant le nombre de flux dans le modèle.

L’avantage de cette stratégie est qu’elle ne nécessite ni de données ni de temps additionnel pour estimer les poids.

- Approche par fréquence relative : Les poids sont estimés à partir de la segmentation forcée obtenue par l’algorithme de décodage multi-flux où on calcule le rapport entre le nombre de fois où l’expert améliore la reconnaissance et le nombre d’occurrence du caractère concerné dans la base d’apprentissage.

$$\alpha_j^k = \frac{n_k^j}{n_j}$$

où n_k^j est le nombre de trames dans les données d’apprentissage où l’expert k a la meilleure probabilité pour l’état j et n_j le nombre d’occurrence de l’état j dans la base d’apprentissage.

A ce niveau, on détermine un poids pour chaque modèle de caractère. Cette approche peut être généralisée pour affecter un poids à chaque état dans le modèle de caractère.

Nous évaluons dans le chapitre 3 ces deux stratégies d’estimation des poids de combinaison dans le cadre de la reconnaissance de mots manuscrits.

Nous avons présenté tout au long de ce chapitre différentes techniques de classification. Malgré leur nature différente, ces diverses approches de reconnaissance ont été appliquées de différentes façons sur le problème de la reconnaissance de l’écriture manuscrite. Par ailleurs, afin d’améliorer les performances des systèmes de reconnaissance, la tendance actuelle est de construire des systèmes hybrides combinant plusieurs classifieurs à des niveaux différents du traitement, ce qui permet d’améliorer les performances des classifieurs individuels [Zouari 04]. Une étude de synthèse de méthodes

hybrides et leur application à la reconnaissance de l'écriture cursive est présentée dans [Xu 92]. Dans le cadre de la combinaison HMM et réseaux de neurones, de nombreux travaux ont été proposés [Augustin 01, Kim 00], inspirés des systèmes neuro-markovien proposés dans le domaine de la reconnaissance de la parole [Morgan 95].

Nous présentons dans ce qui suit un aperçu sur les stratégies de combinaison proposées dans la littérature.

1.9 Les techniques de combinaison de classifieurs

La multiplication des travaux sur la combinaison a entraîné la mise au point de nombreux schémas traitant les données de manières différentes [Heutte 94]. Trois approches pour la combinaison de classifieurs peuvent être envisagées : parallèle, séquentielle et hybride.

1.9.1 La combinaison séquentielle

La combinaison séquentielle, appelée également combinaison série, est organisée en niveaux successifs de décision permettant de réduire progressivement le nombre de classes possibles. Dans chaque niveau, il existe un seul classifieur qui prend en compte la réponse fournie par le classifieur placé en amont afin de traiter les rejets ou confirmer la décision obtenue sur la forme qui lui est présentée (figure 1.11). Une telle approche peut être vue comme un filtrage progressif des décisions dans la mesure où elle permet de diminuer au fur et à mesure l'ambiguïté sur les classes à distinguer. Cela permet généralement de diminuer le taux d'erreur globale de la chaîne de reconnaissance. Néanmoins, une combinaison de ce type demeure particulièrement sensible à l'ordre dans lequel sont placés les classifieurs. En effet, même s'ils ne nécessitent pas d'être les plus performants, les premiers classifieurs invoqués doivent être robustes, c'est-à-dire que la solution réelle de la forme à identifier doit apparaître dans les listes successives quelle que soit leurs tailles. En cas de mauvaise décision du premier classifieur, placé en amont de la série des classifieurs utilisés, l'erreur va se propager de façon irrévocable. Il faudra donc choisir judicieusement le premier classifieur afin d'éviter - autant que possible - l'apparition d'une telle situation. La combinaison séquentielle suppose donc une certaine connaissance a priori du comportement de chacun des classifieurs. Notons que dans cette approche,

chaque classifieur est réglé en fonction du classifieur placé en amont de la chaîne. Une simple modification du premier classifieur peut provoquer un ré-paramétrage (ré-apprentissage) des classifieurs suivants [Zouari 04].

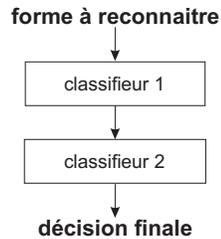


Figure 1.11. Combinaison séquentielle de classifieurs

1.9.2 La combinaison parallèle

La combinaison parallèle consiste à combiner les résultats de deux ou plusieurs classifieurs indépendants les uns des autres, en fusionnant leurs réponses respectives. Cette fusion est soit dirigée, dans ce cas, on attribue à la réponse de chaque classifieur un poids en fonction de ses performances, soit au contraire, ne favorise aucun classifieur par rapport à un autre, (figure 1.12). L'ordre d'exécution des classifieurs n'intervient pas dans cette approche.

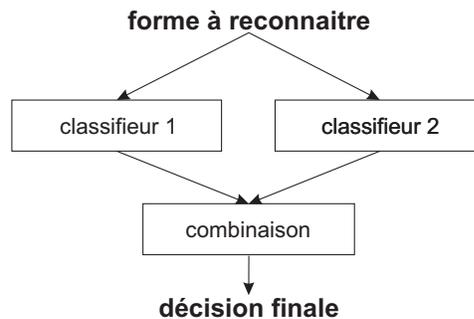


Figure 1.12. Combinaison parallèle de classifieurs

L'inconvénient majeur de l'approche parallèle est qu'elle nécessite l'activation de tous les classifieurs du système qui doivent participer de manière concurrente et indépendante. Par contre, la décision finale est prise avec le maximum de connaissances mises à disposition par chaque classifieur. Dès lors se posent les problèmes de précision des informations fournies par les classifieurs et de la confiance qu'on peut accorder à chacun d'eux.

1.9.3 Les architectures hybrides

Les architectures hybrides consistent à combiner des architectures séquentielles et parallèles (figure 1.13). L'idée principale, est de combiner les avantages de ces deux architectures et d'empêcher les inconvénients. Ce type d'approche permet de générer de nombreux schémas de coopération qui sont les plus souvent complètement dédiés à un problème donné et sont par conséquent très difficilement généralisables à d'autres problèmes de classification. Notons, que dans le même contexte, plus particulièrement dans le cadre de

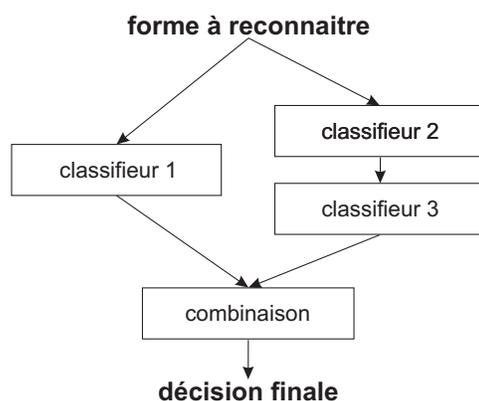


Figure 1.13. Combinaison hybride de classificateurs

la fusion d'informations, d'autres outils ont été proposés dans le cadre de la théorie des fonction de croyance que nous détaillerons dans le chapitre 4.

1.10 Conclusion

Nous avons dressé tout au long de ce chapitre un panorama des principales méthodes de classification dans le domaine de la reconnaissance des formes et plus particulièrement de la reconnaissance de l'écriture manuscrite. Le choix d'une approche plutôt qu'une autre est lié à certaines contraintes comme la représentation d'entrée, la taille de la base d'apprentissage disponible, le temps de calcul requis...

Nous avons mis l'accent sur les méthodes Markoviennes qui sont utilisées dans ce travail. Leur application en reconnaissance automatique de l'écriture hors-ligne a permis d'obtenir des résultats intéressants pour certaines applications malgré la nature 2D de l'information véhiculée par les images de l'écriture. Le succès de ces modèles revient essentiellement à leur capacité

d'intégration du contexte et d'absorption du bruit, ainsi qu'à leurs fortes bases théoriques et leurs algorithmes efficaces que ce soit pour l'estimation des probabilités (Forward-Backward), l'apprentissage (Baum Welch) où le décodage (viterbi).

Nous avons aussi présenté dans ce chapitre une autre variante de ces modèles markoviens, les modèles multi-flux ou multi-stream. Ces modèles permettent de combiner différentes sources d'information en utilisant des modèles de Markov coopératifs. Peu utilisées dans le domaine de la reconnaissance de l'écriture, ces approches présentent de nombreux avantages et ont été exploitées dans ce travail dans le cadre de la reconnaissance de mots manuscrits isolés.

Nous présentons dans le chapitre suivant l'application des approches de classification les plus répandues dans le domaine de la reconnaissance de l'écriture manuscrite, après avoir dressé un état de l'art sur les caractéristiques de l'écriture manuscrite, l'architecture d'un système de reconnaissance de mots isolés, ainsi que les différentes stratégies de reconnaissance proposées dans la littérature.

Chapitre 2

La reconnaissance de l'écriture manuscrite hors-ligne

2.1 Introduction

La problématique de reconnaissance de l'écriture manuscrite a été abordée depuis plus de quatre décennies. On distingue communément deux secteurs d'application en reconnaissance de l'écriture manuscrite suivant le mode d'acquisition :

- La reconnaissance d'écriture dite en-ligne pour laquelle le texte est saisi avec un stylet sur une surface sensible, fournissant ainsi une information temporelle représentée par un ensemble de coordonnées ordonnées dans le temps. Il est ainsi possible de suivre le tracé, de connaître les posés et levés de stylo et éventuellement l'inclinaison et la vitesse.
- La reconnaissance d'écriture dite hors-ligne pour laquelle aucune information temporelle n'est disponible : la page de texte est scannée afin d'obtenir une image numérique.

Ces dernières années, la reconnaissance de l'écriture manuscrite a connu de grands progrès, et les succès des travaux de recherches ont donné lieu à de nombreuses applications industrielles hors-ligne, notamment dans le domaine de la lecture automatique de formulaires [Ramdane 03, Cracknell 98, Milewski 06], de chèques [Knerr 97, Heutte 97, Leroux 97, Duzba 97, Kim 97] ou d'adresses postales [Cohen 94, El-Yacoubi 02, Kim 98]. Les ap-

plications de reconnaissance de l'écriture en-ligne ont connu un essor similaire à travers les PDA, tablet-PC ou stylo caméra [Seni 96, Connell 02].

Ces progrès restent, cependant, insuffisants pour appréhender des textes à grande couverture lexicale dans des conditions non contraintes. En effet, toutes les applications citées exploitent très fortement les contraintes du domaine d'application. Pour les enveloppes postales, le nom de la ville n'est lu que si le code postal n'a pu être reconnu convenablement. Dans ce cas, le lexique utilisé pour la reconnaissance est directement issu de la lecture partielle du code postal. De même, la lecture de chèques bancaires est réalisée en effectuant une lecture du montant littéral et du montant chiffre. Cette double lecture permet de bénéficier d'une vérification des résultats de reconnaissance, et donc de fiabiliser les réponses du système.

Généralement, le script le plus utilisé dans ces systèmes est le Latin (français, anglais, ...). Cependant, ces dernières années, avec la croissance des moyens de communication, d'autres alphabets non latins ont intégré les systèmes d'informations. Ce qui a entraîné l'apparition de documents multi-lingues ou multi-script où l'écriture figure sous différentes langues dans le même document. La reconnaissance de tels documents nécessite des techniques de traitement spécifiques.

Nous présentons dans la partie suivante les caractéristiques d'un système de reconnaissance de l'écriture manuscrite multi-scripts en mettant l'accent sur les deux scripts qui nous intéressent dans ce travail qui sont le Latin et l'Arabe.

2.2 Traitement de l'écriture manuscrite multi-script

Afin de pouvoir proposer un système de reconnaissance d'écriture manuscrite multi-scripts, il est impératif de bien étudier les caractéristiques des scripts considérés. Généralement, la reconnaissance de l'écriture manuscrite dépend de plusieurs critères qui peuvent avoir un impact direct sur la complexité des traitements. Nous les développons dans le paragraphe suivant.

2.2.1 Complexité du traitement de l'écriture manuscrite

D'une manière générale, la complexité de la reconnaissance d'information manuscrite dépend de plusieurs critères (voir figure 2.1) [Crettez 98, Koerich 03] que nous essayons de résumer dans les points suivants :

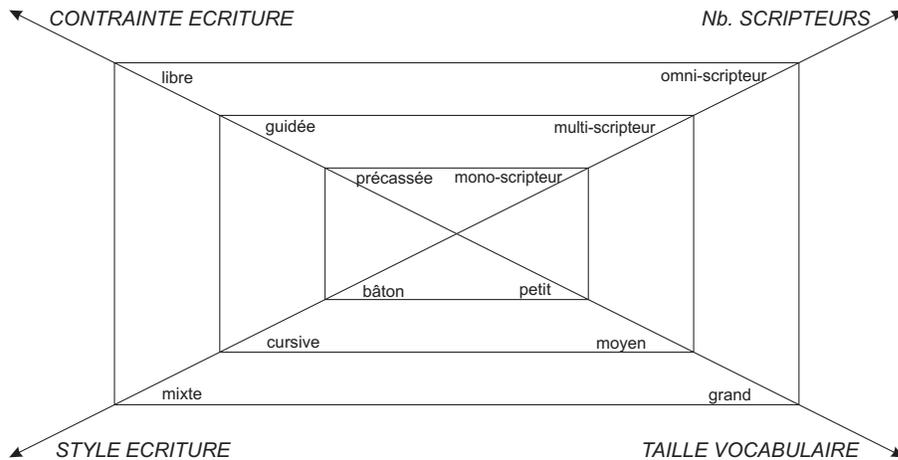


Figure 2.1. Difficultés de l'écriture manuscrite

- Les contraintes d'écriture : l'écriture peut être plus ou moins conditionnée par la présence de précasé (cas des formulaires, code postal d'une adresse), des cadres (montants de chèques) ou lignes (lignes d'un bloc Adresse, montant littéral d'un chèque).
- Le style d'écriture (figure 2.2) : selon [Tappert 84] et en ce qui concerne l'alphabet Latin, la difficulté de la reconnaissance de l'écriture augmente avec les 5 styles d'écriture suivants : écriture scripte précassée, écriture scripte avec caractères espacés, écriture scripte libre, écriture cursive, écriture mixte cursive et scripte. Notons également la présence d'alphabets non latins plus délicats à reconnaître car possédant un grand nombre de symboles : caractères chinois [Park 96b, Liu 00], Kanji [S.Omachi 00], arabes [Amin 98, Al-Hajj 05], japonais [Srihari 97], Devanâgarî [Keeni 96], coréen [Jung 00], etc.
- Le nombre de scripteurs : la réduction du nombre de scripteurs potentiels permet éventuellement de réduire la variabilité et d'apprendre les différents styles d'écriture [Nosary 02]. La difficulté s'accroît en contexte omni-scripteur en raison des styles d'écriture très différents.
- La taille du vocabulaire : les systèmes de reconnaissance de textes sont souvent basés sur un lexique qui facilite grandement la lecture

[Kimura 94], surtout si celui ci possède un faible nombre de mots (cas des montants littéraux de chèques qui contient une trentaine de mots). Il est habituel de considérer qu'un petit vocabulaire est constitué de quelques dizaines de mots, qu'un vocabulaire de taille moyenne comporte quelques centaines de mots, qu'un grand vocabulaire rassemble des milliers de mots et enfin, un très grand vocabulaire renvoie à plus d'une dizaine de milliers de mots.

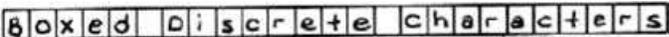
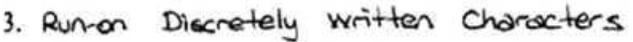
1. 
2. 
3. 
4. 
5. 

Figure 2.2. Classification de l'écriture selon Tappert

La variabilité de l'écriture manuscrite complique son traitement et rend plus difficile de proposer un système de reconnaissance multi-script d'autant plus que les performances des systèmes de reconnaissance travaillant sur l'alphabet Latin ne sont pour l'instant pas directement transposables sur d'autres alphabets. Afin de comparer les deux scripts Latin et Arabes, nous présentons dans la suite les principales caractéristiques de la langue Arabe.

2.2.2 Caractéristiques de l'écriture Arabe

L'écriture Arabe est une écriture alphabétique qui a une origine commune avec les écritures hébraïques, latines et grecques. La première attestation d'un texte en alphabet arabe remonte à 512 après J.-C. C'est au VIIe siècle qu'on a ajouté des points sur ou sous certaines lettres afin de les différencier. Lors de ces modifications, l'ordre des lettres a été changé : de fait, l'alphabet arabe ne suit plus l'ordre traditionnel des autres alphabets sémitiques.

L'alphabet arabe comprend vingt-neuf lettres fondamentales comme le montre la figure 2.5 (vingt-huit si l'on exclut la "hamza", qui se comporte soit comme une lettre à part entière soit comme un diacritique) et s'écrit de droite à gauche. Il n'y a pas de différence entre les lettres manuscrites et les lettres imprimées, les notions de lettre capitale et lettre minuscule n'existent pas (l'écriture est donc monocamérale). En revanche, la plupart des lettres

niveau du sens de l'écriture : l'Arabe s'écrit de droite à gauche. De plus, les caractères arabes sont majoritairement des consonnes, une bonne partie de la vocalisation étant produite par les signes diacritiques (voir figure 2.4). Certains mots se différencient seulement par des signes ou/et des points diacritiques d'où leur importance pour la reconnaissance de mots. Ces signes peuvent se situer au-dessus ou au-dessous du caractère, mais jamais en haut et en bas simultanément. Les signes diacritiques dans la langue arabe sont donc plus nombreux et variés que pour les caractères latins.

Les figures 2.6 et 2.7 présentent des exemples d'écritures arabes imprimées et manuscrites. Ces exemples montrent que la cursivité de l'écriture arabe, la complexité de la morphologie des caractères, les élongations des ligatures horizontales ainsi que les combinaisons verticales de certains caractères, constituent les problèmes majeurs liés au traitement de cette écriture. Tous ces problèmes et bien d'autres, se trouvent accentués dans le cas du manuscrit où d'autres facteurs interviennent (variabilité intra et inter-scripteurs, conditions de l'écriture, fusion de points diacritiques, chevauchement de pseudo-mots, graphismes inégalement proportionnés...).

A travers cette description du script Arabe, nous remarquons que malgré les différences avec le script Latin, ces deux écritures présentent certaines similarités, du fait que ce sont deux langues alphabétiques. Il sera alors possible d'envisager un unique système de reconnaissance traitant ces deux scripts.

Dans la littérature, il y a peu de travaux traitant la problématique de la reconnaissance de l'écriture multi-script, et à notre connaissance, il n'y a pratiquement pas de travaux pour la reconnaissance de l'Arabe et Latin. Nous décrivons dans le paragraphe suivant le principe d'un système de reconnaissance de l'écriture multi-script.

2.2.3 Traitement de l'écriture multi-script

Généralement, il y a deux possibilités pour le traitement de l'écriture multi-script. Dans la première méthode, une étape d'identification de la langue [Elgammal 01] précède la reconnaissance servant à l'orienter vers le système adéquat [Lee 98, Nagy 06]. Il faut disposer dans ce cas d'autant de système de reconnaissance que de nombre de scripts. Lorsqu'on sait les efforts nécessaires pour mettre en oeuvre un seul système de reconnaissance,

Nom	Isolé	Initiale	Médiane	Finale
alif	ا	-		ا
baa	ب	ب	ب	ب
taa	ت	ت	ت	ت
thaa	ث	ث	ث	ث
jiim	ج	ج	ج	ج
Haa	ح	ح	ح	ح
khaa	خ	خ	خ	خ
daal	د	-		د
dhaal	ذ	-		ذ
raa	ر	-		ر
zaay	ز	-		ز
siin	س	س	س	س
shiin	ش	ش	ش	ش
Saad	ص	ص	ص	ص
Daad	ض	ض	ض	ض
Taa	ط	ط	ط	ط
Dhaa	ظ	ظ	ظ	ظ
ayn	ع	ع	ع	ع
ghayn	غ	غ	غ	غ
faa	ف	ف	ف	ف
qaaf	ق	ق	ق	ق
kaaf	ك	ك	ك	ك
laam	ل	ل	ل	ل
miim	م	م	م	م
nuun	ن	ن	ن	ن
haa	ه	ه	ه	ه
waaw	و	-		و
yaa	ي	ي	ي	ي

Figure 2.5. Alphabet arabe

يعبرون على اهل الاندلس فبضرون بهم كل الاضرار
واهل الاندلس ايضا يكابدونهم ويحاربونهم جهده
الطاقة الي ان كان زمان الاسكندر و وصل الي اهل
الاندلس فاعلوه بما هم عليه من التناكد مع اهل السوس
فاحضروا الفعلة والمهندسين وقصدوا مكان الرقاق وكان
ارضا جافه فامر المهندسين بوزن الارض ووزن سطوح
مال البحرين فوجدوا البحر الكبير يشف علوه على البحر
الشامي يشي يسير فرفعوا البلاد التي على الساحل من
بحر الشام ونقلها من اخفض الي ارفع ثم امر ان تحفر
الارض التي بين بلاد طنجة وبلاد الاندلس فحفرت حتى
وصل الحفر الي الجبال التي في اسفل الارض وبني عليها

طبعت هذه النسخة لقراء

اللغة الفرنسية ، نغذر للقارئ

العربي لانه سيكون مضطرا

لقراءة الكتاب من جهة اليسار

Figure 2.6. Exemple d'écriture manuscrite Arabe selon [Amara 00].

حبر بماء التوت : أخلط مقداراً من عصير
التوت مع نفس المقدار من مسحوق الصمغ
العربي ومقدار من ماء العفص ، اتركه في
الظل والقلى نصف رطل من كل هذا
الخليط وزن درهم صمغ في كل يوم وذلك
خلال خمسة أيام ، وبعد ذلك يصبح
الحبر جاهزاً للاستعمال .

الَّذِينَ اتَّخَذُوا دِينَهُمْ لَهْوًا وَلَعِبًا

وَعَرَّتْهُمْ الْحَيَاةُ الدُّنْيَا فَالْيَوْمَ نَنْسَهُمْ

كَمَا نَسُوا لِقَاءَ يَوْمِهِمْ هَذَا وَمَا كَانُوا

بِتَائِبِينَ بِمَجْدُونِ ﴿٥١﴾

Figure 2.7. Exemples d'écritures imprimées Arabes selon [Amara 00].

on imagine alors la difficulté à en mettre en oeuvre un second, d'où la difficulté à mettre en oeuvre de telles approches. D'autre part, l'efficacité du système de reconnaissance dépend dans ce cas de la précision des résultats d'identification du script.

D'autres alternatives consistent à proposer un unique système de reconnaissance capable de traiter directement l'écriture indépendamment du script. De tels systèmes opèrent d'une façon indépendante de la nature du script et sont plus adaptés aux problèmes de la reconnaissance de l'écriture manuscrite multi-scripts. Cependant, dans la littérature, peu de travaux traitent cette problématique. On peut citer les travaux de [Malaviya 96] qui propose un système de reconnaissance pour trois scripts différents : le Latin, le Devanagari, et le Kanji. Dans [Chaudhuri 97], il a été proposé un système de reconnaissance permettant la lecture de deux scripts indiens : Bangla and Devanagari. Dans le même contexte, dans [Zhu 97], les auteurs traitent la reconnaissance de mots clés dans des documents imprimés écrits avec des scripts orientaux (Chinois, Japonais et Coréen).

Quelque soit le script, un système de reconnaissance de l'écriture manuscrite passe par trois étapes principales que nous décrivons dans le paragraphe suivant. Nous présentons ensuite les différentes stratégies de reconnaissance de mots isolés en analysant leurs avantages et leurs inconvénients ainsi que leurs applicabilité dans le cadre d'une approche multi-script. Nous terminons ce chapitre par une étude des principaux systèmes de reconnaissance présentés dans la littérature.

2.3 Architecture d'un système de reconnaissance de mots manuscrits isolés

Généralement, l'architecture d'un système de reconnaissance de l'écriture manuscrite est composée de trois étapes (figure 2.8) : les pré-traitements, l'extraction des caractéristiques et la classification.

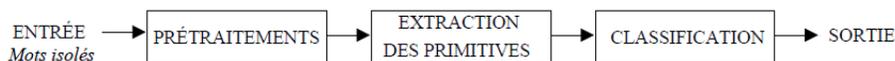


Figure 2.8. schéma général d'un système de reconnaissance de l'écriture manuscrite

2.3.1 Pré-traitement

L'étape de pré-traitements regroupe l'ensemble des opérations visant au bon conditionnement du message écrit et qui sont indispensables à son identification. Ces opérations sont appliquées afin de réduire le volume de données à traiter et à éliminer autant que possible les bruits et les variabilités.

Tout système de reconnaissance d'écriture manuscrite doit impérativement localiser l'information avant de la reconnaître. La localisation consiste à isoler les entités que l'on cherche à identifier tels que les lignes et les mots. Nous présentons dans ce qui suit le principe de la localisation des lignes et des mots.

- La localisation des lignes : De nombreuses méthodes ont été développées pour réaliser une segmentation d'un document en lignes d'écriture. Une des plus faciles et anciennes méthodes de segmentation en ligne est basée sur les histogrammes de projection horizontaux des pixels noirs, les minima locaux de l'histogramme étant considérés comme un espace inter-ligne. Cette méthode a été appliquée aussi bien pour le Latin [Marti 01, Vinciarelli 04] que pour l'Arabe [Romeo-Pakker 95, Syiam 06], son principal inconvénient est qu'elle ne convient pas aux documents dont les lignes sont trop fortement inclinées. Dans le cas de l'Arabe, l'apparition fréquente des marques diacritiques génère des fausses lignes en créant des faux minima qui correspondent aux espaces entre les caractères et leurs marques diacritiques.

D'autres méthodes ont été proposées dans la littérature, nous citons les techniques par accroissement de groupes d'entités connexes [Srihari 05, Ball 06, Farooq 05], les méthodes par transformée de Hough [Likforman-Sulem 95], les méthodes à ombrage (shading) [Cohen 91]... On peut se référer à l'état de l'art de Likforman-Sulem [Likforman-Sulem 07] pour une revue des algorithmes de segmentation en lignes de documents textes.

- La localisation des mots : Une fois les lignes de textes extraites, il faut isoler les mots entre eux afin de les reconnaître. Il s'agit d'un problème de classification à deux classes dans lequel on essaie de différencier les espaces inter-mots et intra-mots. Ceci est plus compliqué dans le cas de l'Arabe vu que l'hypothèse de départ qui consiste à considérer que les espaces inter-mots sont plus importants que les espaces inter-pseudo-

mots est loin d'être vérifiée. Deux types d'approches sont envisageables pour la segmentation des lignes en mots [Heutte 03] : les méthodes de segmentation-reconnaissance qui procèdent par validation de la segmentation en mots par la reconnaissance de la ligne voire même de la phrase [Ball 06, Farah 04a] ; et les méthodes qui procèdent à une segmentation préalablement à la reconnaissance [Shahabi 06].

Après localisation, les processus de normalisation permettent de standardiser les dimensions des données écrites en procédant à des corrections de taille sur le texte et l'épaisseur du trait. Ils permettent également d'absorber partiellement les variations de style des scripteurs (inclinaison, dimensions variables...) pour tendre à ne garder que les informations pertinentes. Parmi les traitements les plus utilisés :

- La suppression de la pente des mots : Un défaut d'orientation du support pendant l'acquisition, ou une écriture imprécise peuvent conduire à une inclinaison et une déformation de la ligne de base. La détection de l'inclinaison moyenne d'un mot selon la direction horizontale est effectuée soit globalement sur le mot [Dehghan 01, Alma'adeed 04], soit par morceaux [Farooq 05] en se basant sur la détection de la ligne de référence de l'écriture. Une fois que l'angle d'inclinaison déterminé, il ne reste qu'à redresser, par rotation, l'image de mot.
- La correction de l'inclinaison du texte : On remarque que très souvent, les caractères sont inclinés : leur direction générale forme un angle avec la verticale variable selon les scripteurs. Les systèmes de reconnaissance corrigent toujours cette inclinaison pour diminuer la variabilité intra-classe et pour rendre la segmentation en caractères plus facile, notamment dans le cas d'une segmentation par des lignes verticales (notion de fenêtres glissantes). Les techniques de correction sont en général fondées sur la détection des traits quasi verticaux. On calcule leur inclinaison moyenne par rapport à la verticale et on corrige l'inclinaison de tous les mots avec ce même angle moyen [Vinciarelli 00].
- la normalisation : elle ramène les images de mots à des tailles standards. Certaines méthodes cherchent même à normaliser localement les différentes parties d'un mot : par exemple, on veut que la partie centrale, les hampes et les jambes occupent chacun un tiers de la hauteur [Saon 97].

2.3.2 Extraction des primitives

La phase d'extraction des primitives (ou des caractéristiques) est une étape très importante pour un système de reconnaissance de l'écriture manuscrite. L'objectif de cette étape est la sélection de l'information pertinente pour les étapes ultérieures de classification. Dans la littérature, deux catégories principales de caractéristiques sont distinguées : les primitives structurelles et topologiques et les caractéristiques statistiques.

- Les primitives structurelles décrivent les propriétés topologiques et géométriques de l'écriture. Il s'agit de compter dans l'image brute, ou à partir du squelette ou du contour de la forme :
 - Des segments de droite et leurs attributs (position, orientation, ...)
 - Des positions relatives entre segments
 - Des arcs, boucles et concavités, mesures de pentes et autres paramètres de courbures pour évaluer des orientations principales
 - La détection des croisements et des jonctions des traits
 - Des angularités, points extremum et points terminaux.
- Les primitives statistiques véhiculent une information distribuée sur toute l'image. L'histogramme, qui représente le nombre de pixels sur chaque ligne ou colonne de l'image, en est un exemple classique et simple à calculer. D'autres primitives sont souvent utilisées :
 - Les directions des contours dans une fenêtre locale.
 - Le profil de projection des densités de pixels noir/blanc qui représente le nombre de pixels sur chaque ligne ou chaque colonne de l'image.
 - L'histogramme des transitions qui, comme son nom l'indique, permet de retenir le nombre des transitions blanc/noir et noir/blanc entre pixels.
 - Les moyennes locales de pixels de l'image (principe du "zoning")
 - Les moments invariants sont des caractéristiques intéressantes car elles sont invariantes en translation, taille et rotation. Ce sont des mesures statistiques de la distribution des pixels autour du centre de gravité du caractère.

Ces caractéristiques ont été initialement appliquées à la reconnaissance de l'écriture Latine et ont été par la suite réutilisées dans les systèmes de reconnaissances de l'écriture Arabe vu que la topologie de ces deux scripts présente des ressemblances se traduisant par exemple par la présence de

boucles, de concavités, d'ascendants et de descendants...

Nous présentons dans le tableau 2.1 les principales caractéristiques utilisées dans la littérature dans le cadre de la reconnaissance de l'écriture manuscrite arabe.

Auteurs	caractéristiques
[Al-Hajj 05]	densité de pixel, densité des transitions, configuration des concavités dans une fenêtre et par rapport à la ligne de base, centre de gravité par rapport à la ligne de base
[Mozaffari 05]	moyenne et variance des changements en X et Y dans des portions du squelette.
[Alma'adeed 04]	Ascendant, descendant, caractéristiques structurales sur des fenêtres.
[Haraty 02]	boucles, points finaux/d'inflexion/de rebroussement, largeur hauteur, transitions et densité sur les lignes et les colonnes, hauteurs du contour
[Farah 04b]	boucles, points, composantes connexes, Ascendant, descendant
[Amin 03]	boucles, points, lignes, courbatures ouvertes
[Clocksin 03]	des fonctions de moment appliquées à l'image et transformation polaire de l'image.
[Khorsheed 03]	points, boucles, points finaux, lignes, points d'inflexion, de rebroussement
[Pechwitz 03]	densité de pixel sur les images pré-traitées
[Maddouri 02]	Ascendant, descendant, boucles, points, position des parties de mots, descripteurs de Fourier normalisés
[Miled 01]	concavités et inclinaison des ascendants et descendants, densité et forme des diacritiques.
[Amin 96]	types et positions des boucles, positions et directions des lignes
[Al-Yousefi 92]	statistiques des moments des projections horizontales et verticales.

Table 2.1. Les caractéristiques utilisées pour la reconnaissance de l'écriture manuscrite arabe

2.3.3 Classification

Le rôle du classifieur est de se prononcer sur l'appartenance d'une forme à une classe à partir des vecteurs de primitives extraits de cette forme. De nombreux classifieurs existent et sont plus ou moins bien adaptés à la

reconnaissance de l'écriture (voir chapitre 1). L'utilisation d'un classifieur plutôt qu'un autre dépend essentiellement de la représentation d'entrée. Trois représentations sont généralement possibles :

- Représentation vectorielle ou matricielle : un vecteur de caractéristiques est généralement utilisé pour représenter les caractéristiques de niveaux bas et intermédiaire. L'image du mot est divisée en sections de dimension fixe ou variable à partir desquelles on extrait un certain nombre de caractéristiques qui vont être représentées par des valeurs booléennes, entières ou réelles dans une matrice ou un vecteur. Pour ce type de représentation, les classifieurs de type réseaux de neurone et SVM sont généralement les plus adaptées.
- Représentation par séquence : dans ce cas, l'image du mot est segmentée implicitement en plusieurs fenêtres successives sur lesquelles on extrait un vecteur de caractéristiques appelé trame. Pour ce type de représentation, les approches markoviennes sont les plus utilisées et les plus performantes puisqu'elles sont bien adaptées pour modéliser les séquences.
- Représentation par structure de graphe : l'image du mot est représentée par un graphe où les noeuds correspondent aux différentes caractéristiques, et leurs interrelations sont représentées par des arcs. Les approches structurelles sont généralement utilisées pour ce type de représentation.

Le tableau 2.2 présente quelques approches de reconnaissance de l'écriture arabe (moteurs de reconnaissance) proposées dans la littérature.

Après avoir présenté l'architecture d'un système de reconnaissance de l'écriture manuscrite, nous nous intéressons dans le paragraphe suivant aux approches de reconnaissance de mots isolés et nous présentons les différentes stratégies proposées dans la littérature.

2.4 Stratégies de reconnaissance de mots isolés

Les systèmes de reconnaissance hors-ligne de mots manuscrits isolés peuvent être classifiés selon trois grandes familles : les approches globales, les approches analytiques et les approches hybrides.

Auteurs	Système de reconnaissance
[Al-Hajj 05]	modèles de caractères (MMC) à quatre états, sauts d'au plus un état, mixture de 3 gaussiennes par état
[Mozaffari 05]	plus proches voisins
[Farah 04b]	combinaison parallèle de réseau de neurones, KPPV, KPPV floue
[Haraty 02]	réseaux de neurones multiples
[Alimi 97]	Réseau de neurones RBF, logique flou et algorithme génétique
[Clocksin 03]	SVM
[Pechwitz 03]	des modèles de mots (MC) construits à partir de 160 MMC semi-continus de caractères
[Dehghan 01]	MMC discrets avec k-mean pour la QV
[Miled 01]	MMC planaires
[Amin 96]	réseau de neurones à cinq couches
[Al-Yousefi 92]	classifieur bayésien quadratique
[Goraine 92]	régles de classifications

Table 2.2. Quelques systèmes de reconnaissance de l'écriture manuscrite arabe

2.4.1 Les approches globales

Dans ces approches, dites aussi holistiques, l'image du mot est considérée comme une entité indivisible. Des caractéristiques de haut niveau, comme par exemple la présence de hampes et de jambages, de boucles hautes, de boucles médianes, de différentes tailles..., sont généralement utilisées afin de représenter de manière grossière l'apparence d'un mot [Powalka 97, Knerr 97]. Une fois extraites, les caractéristiques sont regroupées dans un vecteur et soumises à un classifieur qui se prononcera alors en faveur d'une hypothèse de reconnaissance mot.

On peut également représenter le mot à l'aide d'une séquence de symboles, auquel cas on a recours aux modèles de Markov cachés pour modéliser les mots. La reconnaissance consiste alors à mettre en concurrence les différents modèles (un pour chaque mot à reconnaître).

Ce type d'approche est utilisé pour de petits vocabulaires ou pour réduire le lexique pour des applications à vocabulaire étendu. Le principal avantage est de ne pas avoir à segmenter les mots en lettres, étape extrêmement dé-

licate comme nous allons le voir par la suite. En revanche, elles comportent un grand nombre d'inconvénients :

- Travaillant sur les silhouettes globales des mots, ces approches ont par définition des difficultés à différencier des mots lexicalement proches.
- L'apprentissage des modèles devient délicat quand le nombre de mots à reconnaître augmente. L'apprentissage du classifieur nécessitant un grand nombre d'exemples de mots de chaque classe, la constitution d'une base d'apprentissage devient rapidement difficile. De plus, les caractéristiques utilisées ont tendance à devenir insuffisamment discriminantes lorsque la taille du lexique augmente.
- Enfin, changer le lexique de l'application nécessite un ré-apprentissage du système.

2.4.2 Les approches analytiques

A l'inverse, la deuxième famille, qui regroupe les approches dites analytiques, reconnaît les mots en s'appuyant sur la reconnaissance de ses lettres. Cette famille peut être scindée en deux : les approches à segmentation implicite [Tay 01, Vinciarelli 00, Cho 95], où la segmentation du mot est une segmentation aveugle dans le sens où elle ne dépend en aucun cas d'une analyse de l'image à segmenter et les approches à segmentation explicite [Kimura 94, El-Yacoubi 99, Nosary 02] qui segmentent l'image du mot en des formes élémentaires appelées graphèmes après une analyse de celui-ci.

2.4.2.1 Les approches à segmentation explicite

Les approches à segmentation explicite segmentent l'image du mot après une analyse de celui-ci. La technique de segmentation dépend alors de la nature du script ainsi que de ses caractéristiques morphologiques. Pour le Latin, la méthode la plus répandue se base sur une analyse du contour afin de détecter les ligatures entre les lettres. Elle se déroule en deux temps [Kimura 94, El-Yacoubi 99, Nosary 02].

Premièrement, une liste de points de segmentation potentiels (PSP) est générée. Les cas de sous-segmentation, c'est-à-dire de non détection d'un point de segmentation, sont souvent fatals pour la reconnaissance. Ainsi, la sur-segmentation du mot est favorisée. Les PSP correspondent généralement

aux minima locaux du contour supérieur du mot. Cette liste de PSP contenant beaucoup de fausses alarmes, de nombreux auteurs utilisent des jeux de règles plus ou moins complexes afin d'éliminer un maximum de points correspondant à des cas de sur-segmentation, ceci afin de limiter le nombre de combinaisons à explorer lors de la reconnaissance. Un exemple de segmentation est donné dans la figure 2.9.

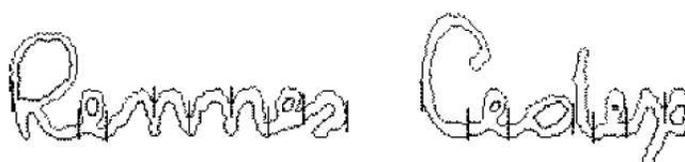


Figure 2.9. Exemple de segmentation explicite réalisée dans [El-Yacoubi 99]

Pratiquement, la réutilisation directe des techniques de segmentation du manuscrit latin pour segmenter les mots arabes manuscrits n'est pas suffisante et aboutit dans la majorité des cas à des erreurs fatales. Ceci est dû à la nature semi-cursive de l'écriture arabe, à la présence trop fréquente des marques diacritiques dans le mot, à la diversité des formes des caractères arabes, à la courte liaison qui existe entre caractères successifs, à l'allongement des ligatures horizontales et à la présence de ligature verticales. Les travaux sur la segmentation de l'écriture introduisent alors des heuristiques qui dépendent généralement de la nature du script Arabe. Dans [Sari 02, Sari 05] un PSP (correspondant à un minima dans le contour inférieur) est retenu si l'une des règles suivantes est vérifiée : il suit un ascendant ou il est suivi par un descendant ; il est le dernier PSP dans le pseudo-mot et suivi d'un ascendant ou un descendant ; il se situe à gauche d'une marque diacritique, mais s'il est le dernier PSP dans le pseudo-mot, il doit être suivi d'un ascendant ou un descendant ; il suit deux PSP rejetés ; il suit une boucle ; il suit un point d'inflexion. En revanche, un PSP est éliminé si l'une des règles suivantes est vérifiée : il passe par une boucle ; il se trouve comme le premier ou le deuxième PSP dans une séquence de trois PSP ; il est le dernier PSP dans le pseudo-mot mais il n'est pas suivi ni d'un ascendant ni d'un descendant. Les PSP qui ne sont ni retenus ni éliminés sont stockés dans une liste de points de segmentation ambigus. Dans [Olivier 96, Miled 98] un PSP (correspondant à un minima dans le contour supérieur) est éliminé si l'une des règles suivantes est vérifiée : il se situe

au-dessus d'une boucle ; l'épaisseur du trait d'écriture dans ce PSP dépasse une certaine proportion de l'épaisseur moyenne du mot ; Il est situé dans une région contenant plusieurs PSPs et n'est pas le PSP le plus proche de la ligne de base. Dans [Ball 06, Srihari 06] les auteurs localisent des ligatures potentielles entre lettres en cherchant les portions de trait d'une épaisseur inférieure à l'épaisseur moyenne du trait dans l'image, en complétant l'analyse par détection des concavité sur les contours supérieurs et inférieurs afin de proposer des points de segmentation supplémentaires.

2.4.2.2 Les approches à segmentation implicite

Les approches à segmentation implicite s'inspirent des méthodes utilisées dans le domaine de la reconnaissance de la parole. La segmentation du mot est une segmentation aveugle dans le sens où elle ne dépend en aucun cas d'une analyse de l'image à segmenter. Elle peut s'illustrer par une fenêtre, dont la hauteur correspond à la hauteur de l'image, glissant horizontalement de gauche à droite. A chaque position, l'image contenue dans la fenêtre est extraite.

Deux paramètres sont à prendre en compte lors d'une telle segmentation : la largeur de la fenêtre et le taux de recouvrement entre deux positions successives. Ces deux paramètres conditionnent le bon fonctionnement du système de reconnaissance et doivent être choisis avec soin. La largeur de la fenêtre doit être telle qu'elle ne puisse contenir qu'une lettre au plus. Elle est généralement de l'ordre de quelques pixels. Un exemple de segmentation est donné dans la figure 2.10. De cette segmentation résulte une séquence d'images



Figure 2.10. Exemple de segmentation implicite réalisée dans [Tay 01]

à partir des quelles peuvent être extraites différentes caractéristiques. Dans [Senior 98], une grille est superposée sur chaque imagerie. Dans chaque case de cette grille, les directions du squelette du mot sont considérées. Dans [Vinciarelli 00], c'est la répartition des pixels noirs dans chaque case de la grille qui est prise en compte. Dans tous les cas, l'image du mot est décrite par une séquence d'observations. Elles peuvent être de deux natures :

- symboles : on parle alors d'observations discrètes

– vecteurs de caractéristiques : on parle alors d’observations continues. La reconnaissance d’un mot consiste donc à faire correspondre chaque observation à une lettre ou un morceau de lettre. Dans ce cas, les HMM sont généralement utilisés et ont permis d’obtenir des résultats intéressants. Leur utilisation est cependant légèrement différente que pour les approches globales. Il ne s’agit plus ici d’apprendre des modèles de mots mais d’apprendre des modèles de lettres. Ces modèles pourront alors être concaténés de manière à obtenir le modèle correspondant à chaque mot du lexique. De même, l’apprentissage est sensiblement différent de celui réalisé pour les approches globales. En effet, il ne s’agit plus d’apprendre des modèles de mots, mais des modèles de lettres, on utilise pour cela un apprentissage embarqué comme décrit en chapitre 1.

2.4.2.3 Les approches hybrides

La troisième stratégie est dite hybride [Snoussi-Maddouri 02], elle est le fruit de la combinaison des précédentes. Elle consiste soit à combiner les listes de mots fournies par différents classifieurs le plus souvent basés sur des primitives et des stratégies différentes, soit à se baser sur les résultats d’une méthode pour en guider une autre.

2.4.3 Conclusion

Nous venons de présenter les différentes approches utilisées pour la reconnaissance de mots manuscrits préalablement isolés. Différentes stratégies ont été mises en place, chacune ayant ses avantages et ses inconvénients. Dans le cadre d’applications où le lexique est relativement faible (une cinquantaine de mots) et statique, les approches globales offrent une solution simple et rapide à mettre en oeuvre. En effet, elles basent leur décision sur une analyse globale du mot contournant ainsi le délicat problème de segmentation.

Si l’application doit gérer un lexique plus large, dynamique, ou inconnu lors de la conception du système, les approches analytiques sont incontournables. Cependant, les méthodes de segmentation nécessitent une connaissance a priori du script concerné et les techniques appliquées à un script donné, ne sont pas directement opérantes sur d’autres scripts.

Tenant compte de ces contraintes, dans le cas d’un système de reconnais-

sance de l'écriture multi-script opérant sur un large lexique, une approche analytique à segmentation implicite nous semble la plus adéquate.

Nous présentons dans ce qui suit un état de l'art sur les méthodes de reconnaissance de l'écriture.

2.5 État de l'art sur les systèmes de reconnaissance de l'écriture manuscrite

Plusieurs systèmes de reconnaissance de l'écriture manuscrite ont été proposés dans la littérature. Ces méthodes peuvent être catégorisées suivant le script étudié (Arabe ou Latin), selon l'approche de segmentation (explicite, implicite, sans segmentation ou holistique), ou aussi selon la technique de modélisation (MMC $1D$, MMC $1D^{1/2}$, MMC $2D$, hybride ou sans MMC).

2.5.1 Application des modèles $1D$

De nombreux travaux ont été proposés dans la littérature utilisant des HMM $1D$ pour la reconnaissance de l'écriture manuscrite. Bien que ces systèmes utilisent tous la même modélisation basée sur des HMMs $1D$, l'approche peut être différente d'un système à un autre. Nous trouvons alors des approches avec ou sans segmentation, modélisant des entités de nature différente (un mot, une lettre, un graphème ...), utilisant une topologie de modèles gauche-droite ou ergodiques, émettant des observations discrètes, continues ou semi-continues.

2.5.1.1 Approche avec segmentation explicite

Le système de [El-Yacoubi 99]

L'approche proposée dans [El-Yacoubi 99] est une approche analytique basée sur les HMM discrets de type modèles d'arcs (les observations sont émises le long des transitions) pour la reconnaissance hors-ligne des mots latins non-contrainte dans un vocabulaire large.

Suite à l'étape de segmentation où une liste de points de segmentation potentiels (PSP) est générée, chaque graphème est décrit par une paire de symboles discrets parmi trois jeux de caractéristiques.

Le premier jeu de caractéristiques est basé sur les caractéristiques globales

des graphèmes (boucles, hampes, jambages...). Les boucles sont représentées par différents symboles suivant leur appartenance à chacune des trois zones d'écriture et leurs tailles relatives par rapport à ces zones, ainsi, une boucle dans la zone supérieure est désignée par un « L », une petite boucle située dans la zone médiane est désignée par le symbole « o ». Afin de discerner correctement les « b », « d », « p » et « q », lorsqu'une ligne verticale est détectée à proximité d'une boucle située dans la zone médiane, différents symboles seront invoqués suivant que la ligne est une hampe, un jambage, et qu'elle est située en amont ou en aval de la boucle. Les graphèmes qui ne présentent pas de propriété morphologique notable sont désignés par le symbole « - ». On aboutit ainsi pour le premier jeu de caractéristiques, à un alphabet de 27 symboles.

Le second jeu de caractéristiques utilise pour chaque graphème, deux histogrammes des transitions de contours réalisées par projections horizontales et verticales. Une fois quantifiés, ces couples d'histogrammes permettent de distinguer 14 types de formes différentes auxquelles on associe autant de symboles.

Le dernier jeu de paramètres caractérise la nature de la liaison entre les graphèmes. S'il existe un espace entre deux graphèmes, différents symboles lui sont associés suivant qu'il est négligeable, étroit ou large par rapport à la largeur moyenne d'un graphème. S'il existe au contraire une liaison connexe, alors la hauteur du point de segmentation est prise en compte, et la liaison est désignée par différents symboles suivant sa distance à la ligne de base.

Pour chaque mot, deux séquences de caractéristiques sont générées. La première est constituée d'une alternance de caractéristiques du premier et troisième jeux de caractéristiques et la seconde d'une alternance de caractéristiques du deuxième et troisième jeux de caractéristiques (voir figure 2.11).

Les lettres sont modélisées par des HMMs plus complexes que des modèles gauche-droite. Dans ces modèles, présentés sur la figure 2.12, les différentes segmentations d'une lettre sont prises en compte : le chemin 0-6-7 correspond à une lettre segmentée en un seul graphème, le chemin 0-1-2-3-6-7 à 2 graphèmes et le chemin 0-1-2-3-4-5-6-7 à 3 graphèmes. Enfin, le cas de sous-segmentation est également pris en compte et correspond au chemin 0-7 avec une transition nulle (qui ne consomme pas d'observations). En reconnaissance, le modèle global correspondant à un mot du lexique est construit par concaténation des modèles de lettres. Un modèle de lettre

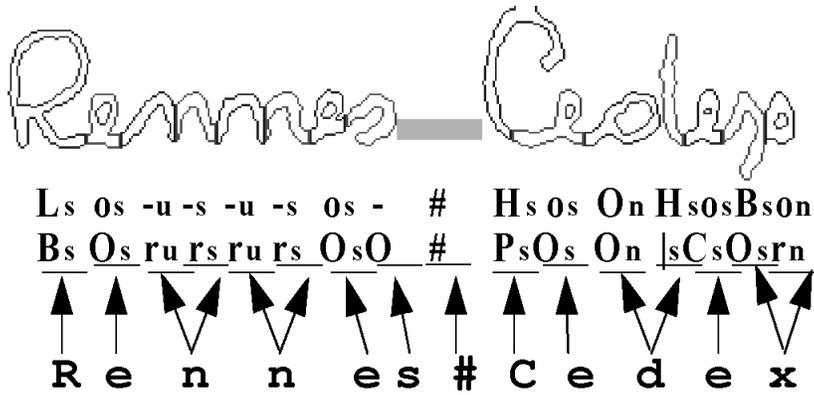


Figure 2.11. Séquence de symboles décrivant un mot dans [El-Yacoubi 99]

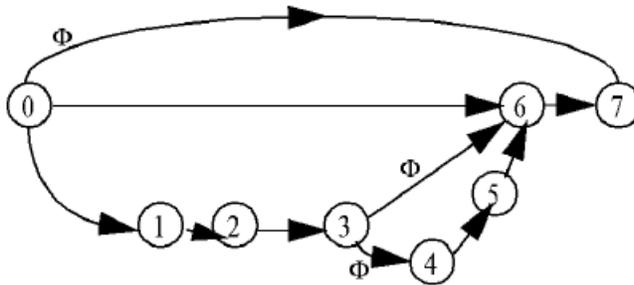


Figure 2.12. HMM modélisant une lettre dans [El-Yacoubi 99]

est ici constitué par la mise en parallèle des deux modèles liés à celle-ci (majuscule et minuscule). De ce fait, les modèles de deux caractères adjacents sont alors connectés par quatre branches ou transitions. Chaque transition correspond à l'une des quatre possibilités d'écrire deux lettres consécutives (majuscule-majuscule (UU), majuscule-minuscule (Ul), minuscule-majuscule (IU), minuscule-minuscule (ll)) (voir figure 2.13).

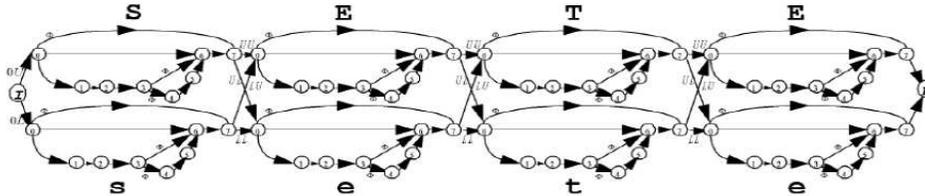


Figure 2.13. Modèle global de reconnaissance du mot "SETE", d'après [El-Yacoubi 99]

2.5.1.2 Approche avec segmentation implicite

Le système de [Al-Hajj 05]

L'approche proposée par [Al-Hajj 05, Al-Hajj 06, Al-Hajj 07] dans le cadre de la reconnaissance de mots manuscrits arabes est de type analytique, sans segmentation explicite des mots en ses caractères constitutants, et est basée sur une modélisation stochastique de type MMC. Le modèle d'un caractère est un MMC à quatre états de type gauche-droite. Les densités de probabilité des observations dans chaque état sont modélisées par un mélange à 3 composantes gaussiennes. Les modèles de mots sont construits par concaténation des modèles de caractères. La méthode adoptée est à deux étapes : une étape de reconnaissance dans laquelle différents types de caractéristiques liées aux densités et aux configurations locales de pixels sont examinés, et une étape pour la combinaison des classifieurs en post-traitement dans laquelle différentes stratégies de combinaison sont appliquées. Les classifieurs combinés en post-traitement prennent en considération les inclinaisons, les positions non précises des marques diacritiques et les chevauchements pouvant exister dans l'écriture manuscrite (voir figure 2.14). Le système de référence basé sur la méthode proposée a gagné la compétition organisée à ICDAR'05 [Märgner 05], où les systèmes à l'état de l'art sont comparés et examinés sur la base de référence IFN/ENIT [Pechwitz 02].

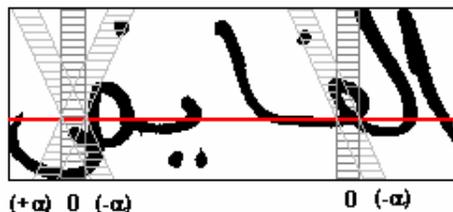


Figure 2.14. Extraction de vecteurs de primitives avec des fenêtres inclinées [Al-Hajj 05]

Le système de Siemens [Schambach 08]

Siemens a dévoilé son système de reconnaissance de mots manuscrits Arabes [Schambach 08] avec lequel la société a remporté la compétition ICDAR'07 [Märgner 07]. Ce système à base de HMMs a été conçu initialement pour la reconnaissance de mots Latins, et a été adapté dans le cadre de la compétition au script arabe. Après une série de prétraitements, l'image du mot est divisée en plusieurs trames en appliquant une fenêtre glissante dont les paramètres sont la largeur et le chevauchement ($2/3$ de la largeur). Chaque fenêtre est divisée en six zones et un vecteur de caractéristiques de dimension 20 est extrait sur chacune. Un caractère est modélisé par un HMM gauche-droite avec self-transition composé de plusieurs chemins (voir figure 2.15).

Tous les chemins dans le modèle sont suivis par un état pause optionnel. Les chemins correspondent aux différents modèles d'allographes d'un caractère. L'apprentissage des modèles est réalisé avec l'algorithme de Baum-Welch. Des adaptations ont été apportées à ce système pour le tester sur le script arabe et qui sont présentées dans [Schambach 08]. A noter qu'à partir de ce système, un deuxième a été proposé combinant trois classifieurs HMMs avec trois jeux de caractéristiques différents. C'est ce dernier système qui a gagné la compétition ICDAR'07 [Märgner 05].

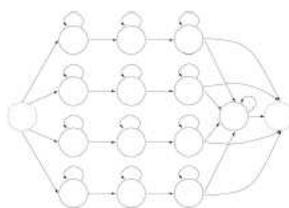


Figure 2.15. Topologie d'un HMM modélisant un caractère selon [Schambach 08]

2.5.1.3 Approche globale

Le système de [Dehghan 01]

Une méthode globale sans segmentation, de type Modèle de Markov Caché (MMC) 1D pour la reconnaissance des manuscrits Arabes/Farsi a été proposée dans [Dehghan 01]. L'approche présentée consiste à modéliser chaque mot, considéré comme une entité indépendante, par un modèle MMC discret. Les caractéristiques utilisées sont basées sur le contour de l'image du mot et sont calculées par zonage qui consiste à découper, de droite à gauche, l'image de mot en un ensemble de bandes verticales de largeur fixe, avec un chevauchement de 50% entre deux bandes consécutives. Chaque bande est ensuite divisée horizontalement en 5 zones de même hauteur. Ainsi, pour chaque bande, un vecteur de caractéristiques de dimension 20 est extrait. Afin de réduire le nombre de symboles d'observation utilisés dans la phase d'entraînement des MMCs, les auteurs ont appliqué une technique de quantification par un réseau de neurones de type cartes auto-organisatrices de Kohonen sur l'ensemble des vecteurs de caractéristiques obtenus. Le système proposé a été examiné sur une base de données (de taille supérieure à 17,000 exemples) de mots manuscrits Farsi/Arabe. Des scores de 65% en premier choix et de 76.1 % en second choix ont été obtenus pour un vocabulaire de 198 mots. Les auteurs ont justifié ces résultats par la simplicité de l'approche globale utilisée, l'absence d'information contextuelle, l'existence d'entrées proches dans le lexique, ainsi que par la nature des scripts Farsi/Arabe (cursivité, positions des points diacritiques, ..) qui rend la reconnaissance automatique de ces scripts difficile.

2.5.2 Application des modèles $1D^{1/2}$

Les applications des modèles $1D^{1/2}$ dans le domaine de la reconnaissance de l'écriture sont basées essentiellement sur les PHMMs. Ces modèles ont été appliqués sur l'imprimé et le manuscrit. Nous exposons dans ce qui suit les principales applications des PHMMs pour la reconnaissance de caractères et des mots en se référant à [Belaid 97].

Le système de Agazzi et Kuo

Agazzi et Kuo [Kuo 94, Agazi 93] ont proposé une architecture PHMM pour la reconnaissance des caractères et des mots imprimés latins, comprenant un modèle vertical de super-états et des modèles horizontaux un par super-état comme le montre la figure 2.16 pour un caractère isolé. Le nombre d'états et de super-états est déterminé manuellement d'après la topologie de la forme considérée, c'est-à-dire en fonction de la présence des zones informatives. L'observation est réalisée directement sur la matrice de pixels et est représentée par un vecteur à quatre composantes, correspondant à la valeur du pixel (0 ou 1), son voisinage horizontal codé binairement (0 à 7), et aux deux positions relatives du pixel, respectivement dans sa colonne et sa ligne (valeur parmi 0...10 tranches (réalisées)).

Pour l'apprentissage, les paramètres du modèle sont estimés en utilisant l'algorithme k-means qui permet de faire des regroupements des observations en classes correspondant aux zones des super-états. La séquence d'observations de chaque échantillon est segmentée en une séquence de super-états, en déterminant l'alignement optimal de l'échantillon avec le modèle courant, à l'aide de l'algorithme de Viterbi. Les paramètres du modèle sont ensuite réestimés, par comptage, en fonction des résultats de la segmentation. Les paramètres initiaux sont donnés manuellement et de manière arbitraire.

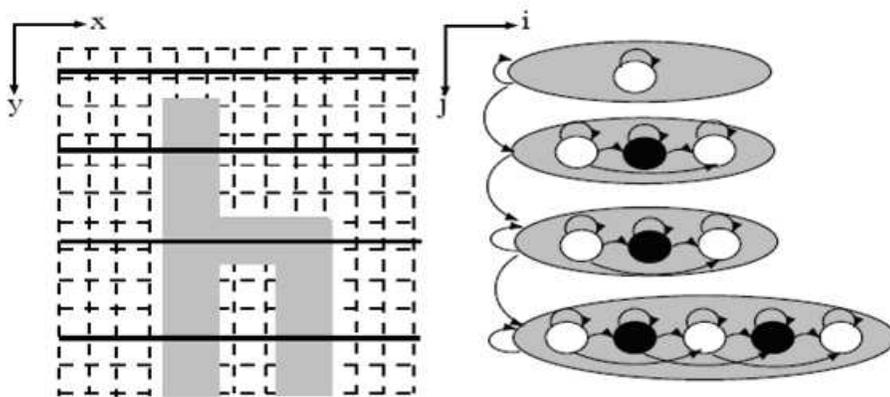


Figure 2.16. PHMM pour un caractère imprimé, d'après [Kuo 94].

Le système de [Amara 99]

Ben Amara a développé un système de reconnaissance de l'écriture arabe imprimée utilisant une architecture de PHMM spécifique à sa morphologie [Amara 99]. La topologie des PHMMs retenue permet de modéliser les différentes variations de l'écriture arabe, en particulier les élongations des ligatures horizontales et la présence des ligatures verticales. Les auteurs ont opté pour une architecture à modèle principal vertical sur un pseudo-mot (voir figure 2.17). L'image d'un pseudo-mot est décomposée verticalement en une ou plusieurs zones selon la topologie de la forme considérée. Les zones correspondent aux bandes de variations de l'écriture arabe, mettant ainsi en évidence des caractéristiques morphologiques des différents caractères qui constituent le pseudo-mot comme les hampes, points diacritiques supérieurs, boucles et/ou lieu de ligature verticale, points diacritiques inférieurs et/ou lieu de ligature verticale et jambage. Chaque zone est modélisée par un modèle secondaire horizontal (HMM 1D) de type gauche-droite (voir figure 2.17). Les observations dans les HMMs horizontaux sont constituées chacune de la couleur du segment courant (noire ou blanche) et de sa longueur en pixels et de son emplacement relatif par rapport au segment qui se trouve au dessus. Le HMM vertical est un modèle à durée explicite pour mieux prendre en compte la hauteur des bandes indiquées. Ainsi la durée dans un super-état est assimilée au nombre de lignes analysées par lui.

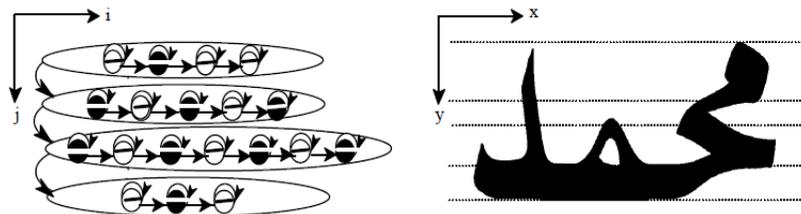


Figure 2.17. Architecture du PHMM, d'après [Amara 99].

2.5.3 Application des modèles 2D

Contrairement aux PHMM, les champs de Markov modélisent une vraie structure 2D. Le problème de ce type d'approches est leur coût calculatoire exponentiel, ce qui les rend inapplicables sans certaines clauses restrictives. Du fait, peu d'approches 2D ont été proposées dans la littérature pour la reconnaissance de l'écriture dont nous présentons le principe.

Le système de Saon

Le système proposé par Saon [Saon 97] est à base de modèles NSHP-HMMs (voir section 2.4.1) pour la reconnaissance de l'écriture manuscrite hors-ligne. Ces modèles sont des champs aléatoires causaux qui présentent des propriétés désirables comme l'applicabilité de procédures récursives pour le calcul de la probabilité de masse ce qui correspond bien au sens (ex. gauche-droite) de production d'écriture manuscrite hors-ligne. La composante NSHP de ces modèles a des capacités de représentation accrues à cause du passé plus étendu qu'elle peut accepter par rapport à d'autres types de champs aléatoires comme les réseaux de Markov. Ainsi, elle permet de modéliser la variabilité de l'écriture suivant les deux directions. Alors que la composante HMM permet de changer dynamiquement les distributions des NSHPs pour les adapter aux primitives rencontrées dans l'image afin de maximiser sa vraisemblance. Ces modèles présentent des capacités d'appariement élastique et leur apprentissage est basé sur le critère MLE. Cet apprentissage, de point de vue champ NSHP, consiste en un comptage ML (Maximum Likelihood) de configurations de pixels, ce comptage étant modulé par la probabilité de se trouver dans l'état associé aux distributions des NSHPs. Il se fait de manière complètement non-supervisée, c'est-à-dire, après une phase d'initialisation aveugle par comptage dans des bandes verticales de même largeur, le modèle NSHP-HMM se charge lui-même de répartir au mieux les colonnes des images dans les états au fur et à mesure des itérations d'apprentissage.

Le système de Park et Lee

Park et Lee [Park 95, Park 96a, Park 96c] ont proposé un système basé sur les réseaux de Markov cachés (Hidden Markov Mesh Random Fields ou HMMRFs) pour une reconnaissance globale de chiffres manuscrits et des caractères coréens manuscrits. La nouveauté de ce travail réside dans l'emploi d'un modèle complètement bidimensionnel. Les auteurs proposent une contrainte permettant de conserver des relations complètement bidimensionnelles entre les observations. Cette contrainte, appelée "look-ahead", consiste à supposer que la maximisation marginale du modèle est optimale. En d'autres termes, considérer les n prochaines observations comme représentatives de l'ensemble des observations restantes. Cela permet de réduire

le calcul de l'optimalité à une petite plage ($n=1$ dans l'étude présentée). La figure 2.18 illustre le modèle HMMRF proposé, et l'hypothèse "look-ahead". Cependant cette hypothèse demeure peu réaliste. En effet, l'une des caractéristiques de base de l'écriture est sa grande variabilité, et le pouvoir d'absorption des variations des modèles markoviens provient de l'optimisation de la répartition des états sur l'ensemble de la forme. En réduisant fortement le contexte futur lors de la répartition, les déformations seront faiblement absorbées et fausseront l'optimisation sur le reste de l'image.

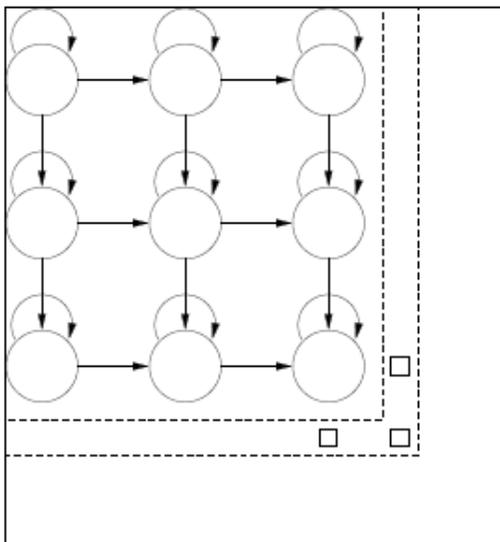


Figure 2.18. Le modèle HMMRF proposé par Park, d'après [Park 95].

téristiques de base de l'écriture est sa grande variabilité, et le pouvoir d'absorption des variations des modèles markoviens provient de l'optimisation de la répartition des états sur l'ensemble de la forme. En réduisant fortement le contexte futur lors de la répartition, les déformations seront faiblement absorbées et fausseront l'optimisation sur le reste de l'image.

Le système de Chevalier

Chevalier [Chevalier 03, Chevalier 04b, Chevalier 04a] a récemment proposé une approche de reconnaissance de l'écriture manuscrite à partir de champs de Markov cachés et fondée sur une analyse entièrement bidimensionnelle de l'écriture. Son originalité réside dans la combinaison d'une analyse fenêtrée de l'image, d'une modélisation markovienne et dans la mise en oeuvre de la programmation dynamique 2D qui permet un décodage rapide et optimal des champs de Markov. L'objectif de l'algorithme proposé est de segmenter les images suivant la position et la direction des traits dans l'image : cela est rendu possible grâce aux primitives spectrales locales. Si cette méthode a été appliquée avec succès à une tâche de reconnaissance

de chiffres manuscrits de la base MNIST, elle est encore à l'état d'ébauche pour la reconnaissance de mots (un taux de reconnaissance de 40% avait été obtenu en ne traitant que les mots de taille inférieure ou égale à 4 lettres).

Dans cette partie, nous avons présenté quelques applications des modèles markoviens avec leurs différentes dimensionnalités. Nous remarquons qu'il y a peu de systèmes qui vont au-delà des modèles 1D, et les systèmes utilisant des PHMMs ou des champs aléatoires de Markov n'ont pas exhibé des performances notoirement meilleures que celles obtenues avec des HMMs 1D, en contre partie d'une complexité prohibitive de calcul.

Quelques approches dites hybrides ont été proposées combinant les avantages des modèles markoviens à ceux des réseaux de neurones. Nous présentons dans ce qui suit quelques systèmes neuro-markoviens.

2.5.4 Application des approches neuro-Markovienne

Le système de [Koerich 03]

Une approche combinant un réseau de neurones et des MMC est présentée dans [Koerich 03, Koerich 05]. Les mots manuscrits sont d'abord traités par un système de reconnaissance de mots basé sur des MMC guidé par un lexique. Une liste des N meilleures hypothèses ainsi que de la segmentation de ces mots en caractères est générée. Un classificateur MLP calcule un score pour chaque caractère en se servant de la segmentation fournie par le premier système. Les résultats des classificateurs MMC et RN sont combinés par une règle de combinaison linéaire des probabilités a posteriori, pour optimiser les performances de reconnaissance (voir figure 2.19).

Le modèle de MMC utilisé pour un caractère est un modèle gauche-droite de 10 états sans self-transition. Des prétraitements dont le but est de diminuer la variabilité intra-scripteur et normaliser l'écriture sont appliqués à l'image du mot. Les caractéristiques extraites sont de nature globale (projections et profils) et locale (histogrammes directionnels). Les premières sont extraites sur la totalité du caractère alors que les dernières sont extraites à travers une grille 3×2 . La juxtaposition de ces caractéristiques donne un vecteur de dimension 108.

L'amélioration de la vitesse de lecture a été mise en oeuvre. Cette stratégie

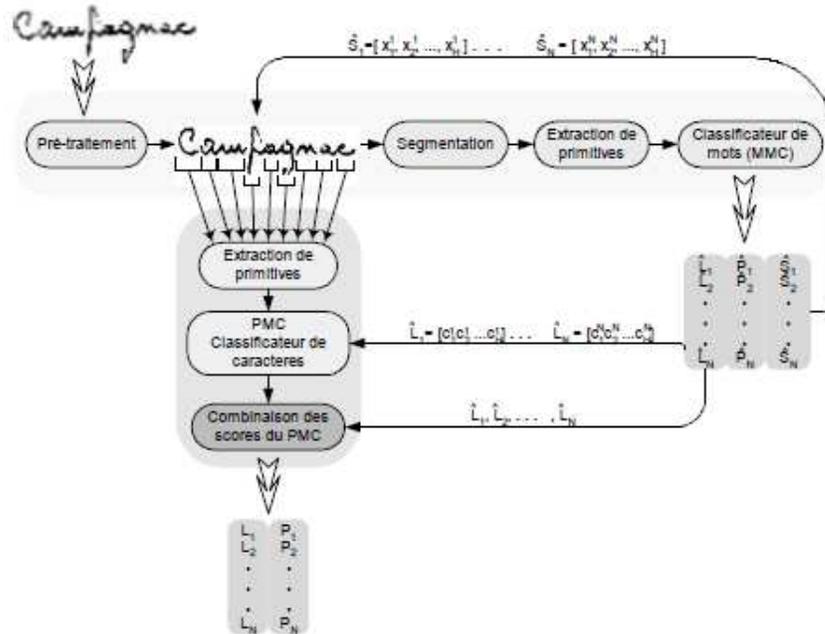


Figure 2.19. Vue d'ensemble du système de reconnaissance hybride MMC/RN [Koerich 03]

consiste à organiser le lexique de reconnaissance sous forme de structure d'arbre, et à découper le calcul des probabilités associées aux mots en un niveau "états" et un niveau "caractères". Ceci permet la réutilisation des probabilités des caractères pour décoder tous les mots dans le vocabulaire et éviter ainsi le calcul répétitif des séquences d'état. La reconnaissance est effectuée par un algorithme "level-building" dirigé par le lexique (Lexicon-Driven Level Building Algorithm ou LDLBA) constituant en quelque sorte une version optimisée de l'algorithme de Viterbi.

2.5.5 Application des modèles multi-flux

L'utilisation des modèles multi-flux dans le domaine de la reconnaissance de l'écriture est restée très limitée. Dans ce cadre, des travaux récents ont été proposés pour la reconnaissance de l'écriture En-Ligne. Dans [Artières 03] les modèles multi-flux sont utilisés pour combiner des flux d'information indicés par deux grandeurs différentes. Le but était de combiner deux représentations de l'écriture, l'écriture en ligne et l'écriture hors ligne. L'écriture en ligne est indiquée par le temps, c'est une séquence temporelle

de points (ou de vecteurs de caractéristiques associés aux points, comme la tangente etc). L'écriture hors-ligne est indiquée par l'abscisse, c'est une image représentée comme une séquence de vecteurs de caractéristiques calculée dans une fenêtre étroite glissante de la gauche vers la droite du tracé. Les auteurs montrent que la combinaison multi-flux permet d'améliorer les résultats de reconnaissance en-ligne de caractères et de mots.

Une méthode similaire à l'approche multi-flux a été proposée dans [Likforman-Sulem 08], qui consiste à utiliser des réseaux bayésiens dynamiques (DBN) pour la reconnaissance de caractères dégradés. Les DBN sont une extension des modèles de Markov cachés $1D$ permettant de gérer plusieurs observations. Dans [Likforman-Sulem 08], les DBN sont utilisés pour coupler deux classifieurs MMC, l'un examinant les lignes de pixels, l'autre les colonnes. Plusieurs types de couplage ont été proposés modélisant les interactions entre parties distantes de l'image. L'architecture du modèle couplé consiste en deux modèles MMC dont les états associés à chaque flux sont reliés par des arcs (voir figure 2.20. Les modèles obtenus ne sont pas forcément symétriques et les dépendances entre les observations des différents flux sont prises en compte par les matrices de transitions entre états. Si l'on considère que les états représentent des configurations similaires d'observations, ces dépendances expriment les probabilités d'avoir des configurations jointes d'observations suivant les deux flux : par exemple une colonne de pixels noirs avec une ligne de pixels noirs. Les résultats obtenus sur la reconnaissance de caractères dégradés montrent que les modèles DBN couplés ont donné de meilleures performances que celles des deux classifieurs MMC mono-flux opérant sur les lignes et les colonnes de pixels, que celles de la combinaison linéaire des scores des deux MMC ainsi que celles des SVM.

Le point commun entre les approches proposées dans les paragraphes précédents est l'utilisation des modèles Markoviens. Ces modèles ont montré un succès considérable dans le domaine de la reconnaissance de l'écriture manuscrite. Cependant, il y a d'autres approches non-markoviennes qui ont été utilisées, nous en décrivons quelques unes dans la partie suivante.

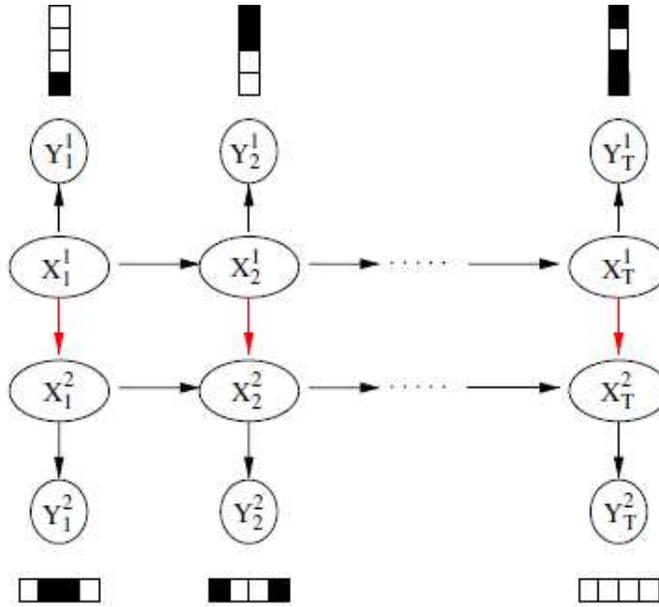


Figure 2.20. DBN : modèle couplé par états

2.5.6 Application des approches non-markoviennes

Le système de [Kimurama 94]

Il est proposé dans [Kimurama 94] un système de reconnaissance de noms de villes en se basant sur un lexique et une base d'apprentissage sur des adresses manuscrites de courriers réels. Après binarisation et correction de l'inclinaison des images de mots, une phase de présegmentation est menée. L'algorithme de présegmentation génère des points de segmentation potentiels sur tous les minima locaux du contour supérieur. Parmi ces candidats, les points d'ordonnée trop proche de l'un des deux maxima les encadrant sont éliminés; la segmentation évite ainsi les vallées trop peu profondes. D'autre part, certains points sont déplacés vers la droite ou la gauche si un faible décalage permet de ne plus couper à nouveau le tracé sur la colonne de pixels passant par ce point. Ceci permet, par exemple, d'éviter de surfragmenter un « c ».

L'image est découpée verticalement selon ces points de segmentation (Figure 2.21), et les composantes connexes issues de ce découpage sont isolées et inscrites dans des sous-images rectangulaires (Figure 2.22). Enfin, on ordonne les graphèmes ainsi obtenus selon une heuristique simple.

Les vecteurs de caractéristiques extraits sont basés sur les propriétés sta-

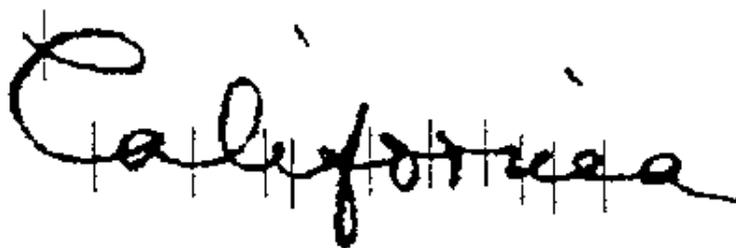


Figure 2.21. Génération des points de segmentation

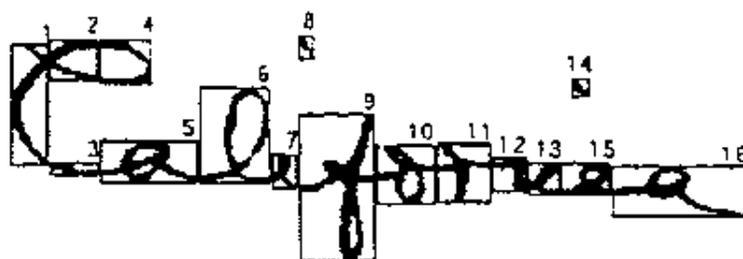


Figure 2.22. Séparation en boîtes englobantes

tistiques du chaincode. Chaque sous image est divisée en $4 * 4$ régions rectangulaires. Sur chacune d'elle le chaincode généré par les contours de la forme est quantifié selon quatre valeurs de pente : 0° , 45° , 90° ou 135° , et un histogramme cumulatif selon ces quatre pentes est produit. Un vecteur de caractéristiques est donc composé de 64 composantes une fois les histogrammes des 16 régions calculés.

Lors de l'étape de segmentation-reconnaissance, est exclu a priori l'ensemble des mots du lexique dont les longueurs en caractères sont incompatibles avec le nombre de graphèmes détectés dans l'image à identifier. Tous les mots restants fournissent des hypothèses pour le mot à identifier, et il s'agit de trouver parmi eux le mot le plus probable, qui correspond à la segmentation optimale en caractères. Pour un candidat du lexique, on s'intéresse à la probabilité d'identifier chacun de ses caractères dans l'image de mot à reconnaître. Pour ce faire, un algorithme de programmation dynamique dégage de tous les agencements de graphèmes envisageables, le plus probable vis à vis du caractère à reconnaître (Figure 2.23).

Les probabilités de chaque caractère sont obtenues par un calcul de distances quadratiques (modified quadratic discriminant function) entre le vecteur de caractéristiques extrait de la forme et le vecteur moyen de la classe considé-



Figure 2.23. Segmentation optimale en caractères

rée. Un facteur correctif dans ce calcul de distance permet de tenir compte de la variance de chaque classe.

Le système de [Koch 04]

La méthode décrite dans [Koch 04] est similaire à la précédente, mais utilise cette fois des MLP (Multi Layer Perceptron). Les agencements de N graphèmes consécutifs sont appelés « graphèmes de niveau N ». Une hypothèse est faite sur le nombre maximum de graphèmes qu'il faut pour reconstituer un caractère, et l'on génère un treillis de tous les graphèmes de niveau N possibles. Le classifieur de caractères se charge d'associer à chaque élément du treillis un score probabiliste pour chacune des 26 classes de caractères. On cherche ensuite dans le treillis la séquence de caractères la plus probable qui pourrait correspondre à chaque mot du lexique. Le mot présentant le meilleur score est supposé être celui qui a généré l'observation inconnue.

2.6 Conclusion

Nous avons présenté dans ce chapitre un descriptif sur la reconnaissance de l'écriture manuscrite ainsi que quelques systèmes de reconnaissance donnés dans la littérature. Bien que ce domaine a été depuis longtemps étudié, les résultats obtenus à ce jour ne sont pas encore satisfaisants surtout pour des grands lexiques, sauf pour quelques applications particulières comme la reconnaissance des montants de chèques ou d'adresses où l'information numérique confrontée à celle textuelle permet d'avoir des systèmes plus robustes. D'où l'intérêt de poursuivre les recherches dans ce domaine d'application.

Nous avons mis l'accent sur la complexité du traitement de l'écriture manuscrite en général, d'autant plus, la reconnaissance de l'écriture manuscrite multi-script qui nécessite des considérations et des traitements supplémentaires. Dans ce contexte, nous avons décrit les principales caractéristiques de la langue Arabe en présentant ses spécificités et en la comparant au script Latin. Notre objectif à travers ce travail étant de proposer un système de reconnaissance opérant indépendamment de la nature du script. Il est primordial donc de procéder sans segmentation explicite vu que les techniques de détermination des points potentiels de segmentation dépendent généralement de la nature du script. D'autre part, nous trouvons qu'une modélisation markovienne convient bien à une approche multi-script vu que ces modèles sont bien adaptés aussi bien pour la modélisation des différentes variations morphologiques des caractères que pour la résolution des problèmes épineux de la segmentation.

Nous présentons dans le chapitre suivant notre système de reconnaissance de mots manuscrits multi-scripts ainsi que les différents résultats obtenus.

Chapitre 3

Un système multi-flux pour la reconnaissance de mots manuscrits multi-script

3.1 Introduction

Dans ce chapitre, nous présentons notre système de reconnaissance hors ligne de l'écriture manuscrite multi-scripts. Le système de reconnaissance proposé opère indépendamment de la nature du script, pour cela, le système procède sans segmentation explicite étant donné que ces techniques de segmentation, comme expliqué dans le chapitre 2, diffèrent d'un script à un autre. De plus, nous avons opté pour une modélisation markovienne qui convient bien à une approche multi-script étant donné que ces modèles sont bien adaptés tant pour la modélisation des différentes variations morphologiques des caractères que pour la résolution des problèmes épineux de segmentation. De même, les phases de pré-traitements et d'extraction de caractéristiques sont indépendantes de la nature du script et des primitives bas niveaux basées sur les densités et directions des pixels ont été utilisées. Afin de remédier au faible pouvoir discriminatif de telles primitives, nous avons fait le choix de combiner plusieurs sources d'informations à travers une approche multi-flux.

Traditionnellement, deux grandes approches sont envisagées en matière de combinaison d'information : la fusion des représentations et la fusion des décisions. La première famille rassemble les techniques utilisant une projection

des représentations des différentes sources d'informations dans un espace de représentation conjoint avant d'utiliser des modèles classiques, comme les modèles de Markov cachés (MMC). Ceci conduit donc à un espace de dimension élevée, d'où une augmentation de la variance lors de l'estimation des paramètres des modèles et donc une diminution des performances du système de reconnaissance. Les méthodes de la deuxième famille s'appuient sur la combinaison de décisions prises indépendamment sur les différentes sources d'informations. Plusieurs travaux ont été réalisés dans le domaine de la combinaison de classifieurs, en particulier les travaux de [Prevost 03, Kim 00] qui ont montré l'importance d'avoir des solutions robustes pour les problèmes de reconnaissance de l'écriture manuscrite. L'inconvénient majeur pour ces deux solutions, c'est que l'on suppose que les différentes sources d'informations sont synchrones, ce qui n'est généralement pas le cas. L'approche multi-flux (cf. chapitre 1 pour le formalisme) est une méthode particulière de combinaison d'informations. L'intérêt de cette approche est qu'elle permet de désynchroniser les caractéristiques. Ce chapitre décrit les principes et les concepts de base de l'approche proposée. Nous détaillons les différentes étapes de notre système de reconnaissance, puis nous présentons les bases de mots que nous avons utilisé dans nos expérimentations. Une présentation et une discussion des résultats viennent pour clôturer ce chapitre.

3.2 Description du système de reconnaissance

Le système de reconnaissance de mots manuscrits isolés que nous avons développé est construit selon l'architecture classique des approches sans segmentation avec une étape de décodage particulière combinant plusieurs sources d'informations à travers une approche multi-flux. Nous présentons tout d'abord la chaîne de pré-traitements et de caractérisation des images d'écriture.

3.2.1 Pré-traitements

Avant de procéder à la reconnaissance de mot, il est nécessaire d'effectuer une série de pré-traitements sur l'image de celui-ci en vue d'éliminer (ou du moins de réduire) les sources de variabilité ou de bruit, et également de simplifier la procédure d'extraction de caractéristiques. La chaîne complète de pré-traitements est illustrée dans la figure 3.1.

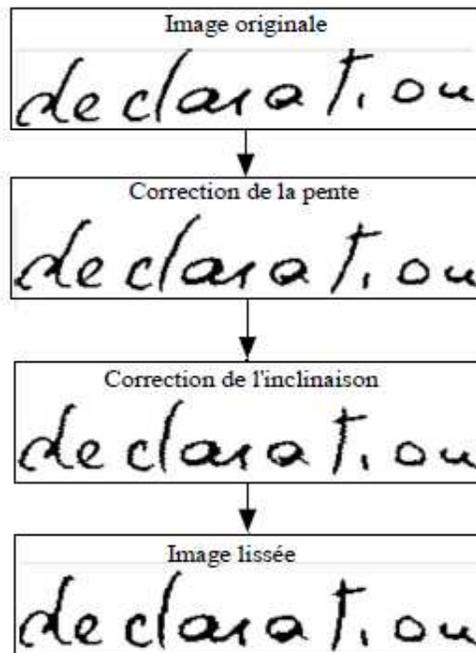


Figure 3.1. Exemples de pré-traitements

3.2.1.1 Correction de la pente "skew"

Nous appliquons une procédure de normalisation dans le but de ramener l'orientation du signal d'écriture à l'horizontale (skew). L'algorithme utilisé procède dans un premier temps à l'extraction de tous les minima locaux du contour inférieur de l'image et calcule, pour chaque point, l'angle qu'il forme avec tous les minima locaux. Il est ensuite nécessaire d'écarter les jambages de ces minima locaux. Ceux-ci étant excentrés de la ligne de base, ils forment des angles importants avec les autres minima. Les points présentant un écart d'angle important par rapport à l'angle moyen sont donc éliminés. Une fois tous les jambages éliminés, on obtient, par la méthode des moindres carrés, la meilleure droite qui relie ces points et l'on détient alors la ligne de base du mot. Il ne reste que redresser, par rotation, l'image du mot.

3.2.1.2 Correction de l'inclinaison "slant"

Une normalisation simple et efficace de l'inclinaison des lettres par rapport à l'axe vertical (slant) est ensuite appliquée [Kimura 94]. Cette norma-

lisation s’effectue en trois étapes : une estimation de l’inclinaison est tout d’abord réalisée d’après le contour du mot. Ensuite l’image est corrigée par décalage des lignes de pixels de l’image (transformation par cisaillement, ou ”shearing transform”). Comme on peut le voir sur la Figure 3.1, cette transformation génère du bruit sur le contour de l’image. On applique alors une dernière étape de lissage du contour par une opération de morphologie mathématique : la fermeture. L’image finalement obtenue est présentée Figure 3.1. L’estimation de l’inclinaison des lettres est réalisée par une analyse du contour du mot. Ce contour est extrait par une méthode de suivi de contour et est encodé sous la forme de «chaincode» selon les 8 directions de freeman [Kimura 94]. Quatre directions sont envisagées pour les segments de contour (0°, 45°, 90° et 135°). L’inclinaison α est alors calculée comme suit :

$$\alpha = \arctan \left(\frac{n_1 + n_2 + n_3}{n_1 - n_3} \right)$$

où n_i est le nombre d’occurrences de la direction i dans le chaincode.

3.2.1.3 Estimation de la ligne de base

La ligne de base porte des informations importantes pour le processus de reconnaissance de mots manuscrits. L’approche appliquée pour l’estimation de la ligne de base utilise le profil de projection vertical des densités des pixels le long de l’axe horizontal (voir figure 3.2). Premièrement, la ligne de base basse correspondante au maximum de la courbe de profil de projection, est cherchée. Après, l’algorithme parcourt l’image du mot de haut au bas pour trouver la ligne de base supérieure (haute) correspondante à la première ligne avec une valeur de projection plus grande ou égale à la densité de rang moyenne. L’approche proposée est sensible à l’inclinaison des mots et

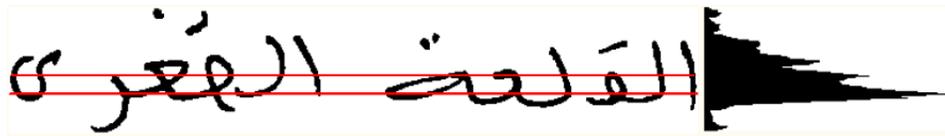


Figure 3.2. Estimation des positions des lignes de base des mots

suppose qu’ils sont écrits horizontalement. Ceci ne pose pas de problème dans notre cas puisque une étape de correction de l’inclinaison de l’écriture est appliquée avant l’extraction de la ligne de base.

3.2.2 Extraction des caractéristiques

Deux ensembles de caractéristiques sont proposés dans ce travail, un premier ensemble basé sur les contours supérieur et inférieur de l'image et un deuxième basé sur les densités.

3.2.2.1 Caractéristiques contours

Ces caractéristiques sont basées sur les points du contour du mot. Ces points sont séparés en un premier ensemble formant le contour supérieur et un second formant le contour inférieur du mot (voir Figure. 3.3). Lors de l'extraction des caractéristiques, nous travaillons sur des fenêtres glissantes positionnées sur les contours. Une fenêtre est caractérisée par une largeur et un taux de recouvrement entre deux positions successives. Pour chaque position de la fenêtre, nous déterminons la direction de Freeman des points du contour supérieur (resp. inférieur), et un histogramme des directions de Freeman est calculé sur tous les points de la fenêtre, constituant ainsi un ensemble de 8 caractéristiques.

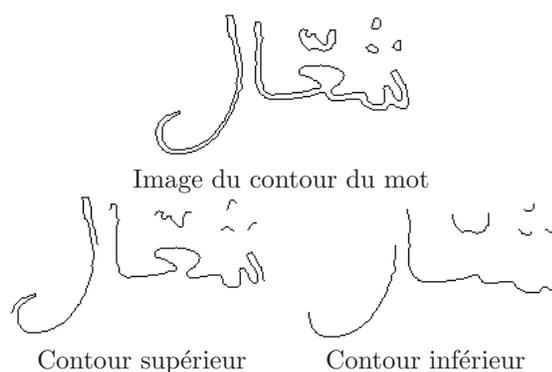


Figure 3.3. Images des contours du mot.

Un deuxième ensemble de caractéristiques consiste à déterminer pour chaque point du contour supérieur (resp. inférieur) la nature du point qui lui correspond verticalement sur le contour en vis à vis.

Quatre cas sont possibles :

- Ce point appartient au contour inférieur (resp. supérieur) (points rouges sur la Figure 3.4).
- Ce point appartient à une occlusion (points bleus sur la Figure 3.4).
- Ce point appartient au contour supérieur (resp. inférieur) (points jaunes sur la Figure 3.4).

- Pas de point, ce qui veut dire que le point du contour supérieur (resp. inférieur) est un point extrême dans la direction horizontale (points verts sur la Figure 3.4).

Les points noirs sur la Figure 3.4 représentent les points du contour inférieur. Ces 4 catégories permettent de caractériser les occlusions, les rebroussements



Figure 3.4. Différents types des points du contour supérieur.

horizontaux (barre de 't' par exemple), les traits simples et les points extrêmes sur l'image du mot. Ces informations locales sont ensuite sommées par catégorie sur toute la largeur de la fenêtre d'analyse pour constituer un second histogramme formant ainsi un deuxième ensemble de 4 caractéristiques.

Un troisième ensemble de caractéristiques indique la position des points du contour supérieur (resp. inférieur) dans la fenêtre. Pour cela, l'image du mot est divisée en trois zones en localisant la ligne de base supérieure et la ligne de base inférieure par la méthode décrite précédemment. Ainsi, nous déterminons pour chaque point du contour supérieur (resp. inférieur) sa position dans la fenêtre, 3 cas sont possibles : 1) le point appartient à la zone médiane, 2) le point appartient à la zone supérieure, 3) le point appartient à la zone inférieure (voir Figure 3.5).

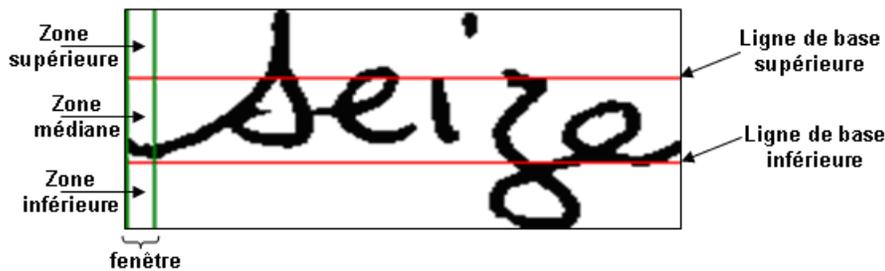


Figure 3.5. Extraction des caractéristiques dans les 3 zones d'écriture

Un histogramme sera calculé pour chaque fenêtre d'où un troisième ensemble de 3 caractéristiques. Ce dernier sous-ensemble fournit une

information supplémentaire sur la présence ou l'absence des ascendants et des descendants dans les mots.

Nous disposons ainsi de deux séquences d'observations constituées chacune d'un vecteur de 15 caractéristiques continues. Chaque séquence est modélisée par un modèle de contour respectivement supérieur et inférieur.

3.2.2.2 Caractéristiques de densités

Un deuxième ensemble de caractéristiques est utilisé dans ce travail basé sur les densités [Al-Hajj 05]. Ce deuxième ensemble permet de détecter la présence des ascendants et des descendants ainsi que les points diacritiques. Des caractéristiques de concavité sont ajoutées pour refléter les concavités locales et les directions des traits. Une caractéristique dérivée communément utilisée dans la reconnaissance de parole est aussi ajoutée. Cette dernière caractéristique permet de surmonter partiellement l'indépendance conditionnelle des observations extraites de l'écriture.

Pour construire une séquence de vecteurs de caractéristiques, l'image d'un mot est balayée par une fenêtre glissante qui s'adapte en hauteur à celle du mot. Chaque fenêtre étant divisée verticalement en 4 cellules (figure 3.6) et les caractéristiques suivantes sont extraites :

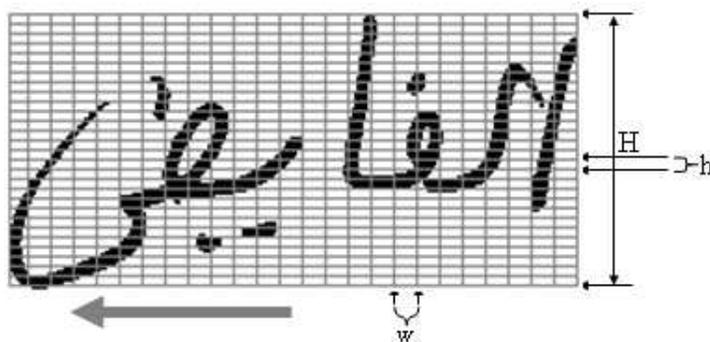


Figure 3.6. Les 5 types de configurations locales autour d'un pixel de fond P

- f_1 : densité des pixels noirs dans la fenêtre.
- f_2 : nombre de transitions Noir/Blanc entre cellules.
- f_3 : différence de position entre les centres de gravité des pixels d'écriture dans deux fenêtres consécutives.

- f_4 à f_{11} : sont les densités de pixels d'écriture dans chaque colonne de la fenêtre (ici une fenêtre est de largeur 8). Les caractéristiques suivantes dépendent de la position estimée des lignes de base.
- f_{12} : position verticale normalisée du centre de gravité des pixels d'écriture, par rapport à la ligne de base basse.
- f_{13} - f_{14} : densités des pixels d'écriture au dessus et au dessous de la ligne de base basse.
- f_{15} : nombre de transitions Noir/Blanc entre les cellules situées au dessus de la ligne de base basse.
- f_{16} : zone à laquelle appartient le centre de gravité de l'écriture dans la fenêtre (zone supérieure $f_{16} = 1$, zone médiane $f_{16} = 2$, zone inférieure $f_{16} = 3$).
- f_{17} - f_{21} : liées aux configurations locales des pixels d'écriture dans chaque fenêtre (5 configurations possibles voir figure 3.7).
- f_{22} à f_{26} sont liées aux configurations locales des pixels d'écriture situés dans la zone médiane de l'écriture.

Cet ensemble est indépendant du langage utilisé. Ces caractéristiques peuvent être utilisées pour tout type d'écriture cursive comportant des hampes et des jambages comme l'écriture Arabe et l'écriture Latine.

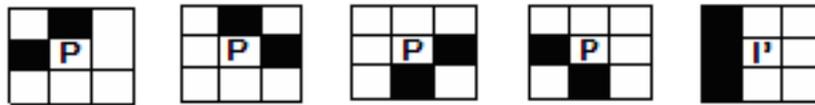


Figure 3.7. Les 5 types de configurations locales autour d'un pixel de fond P

Ces deux ensembles de caractéristiques nous permettent d'extraire 4 flux de données basés respectivement sur des caractéristiques extraites sur les deux contours supérieur et inférieur et des caractéristiques de densités extraites avec deux tailles de fenêtre différentes.

3.2.3 Modélisation des lettres et des mots

Dans l'approche proposée, la modélisation est réalisée au niveau caractère par des MMC. La modélisation des mots du lexique est réalisée par concaténation des modèles de caractères ainsi obtenus. Vu qu'il n'existe pas de moyens théoriques pour déterminer les paramètres des MMC (la topologie du modèle, le nombre d'états nécessaires, le nombre de gaussiennes...) la

solution est alors empirique. De nombreuses expérimentations ont donc été réalisées pour fixer ces paramètres. La topologie retenue est de type gauche droite, n'autorisant que les transitions bouclantes et vers l'état suivant. Le nombre d'états dans le modèle est le même pour chaque caractère et est fixé à 4 états. Les densités de probabilité des observations dans chaque état sont modélisées par un mélange de 3 composantes gaussiennes.

Pour modéliser l'écriture Latine, 52 modèles différents ont été construits correspondant aux caractères minuscules et majuscules, en plus de quelques caractères spéciaux qu'on modélise selon leur présence ou non dans la base sur laquelle on travaille. Dans le cas du script Arabe, le nombre de modèles s'élève à 178 comme décrit dans le paragraphe 3.3.1.

3.2.4 Apprentissage

Les paramètres des MMC multi-flux peuvent être estimés soit indépendamment soit conjointement. Dans la première approche, les paramètres des MMC sont estimés indépendamment dans chaque flux avant d'effectuer le produit. La deuxième approche consiste à profiter de la représentation sous forme de MMC produit pour estimer conjointement tous les paramètres du modèle. Un apprentissage indépendant des flux est utilisé dans ce travail pour deux raisons :

- Ce type d'apprentissage est moins complexe vu qu'il nous évite de lancer un apprentissage sur un MMC-produit dont le nombre d'états est beaucoup plus important.
- Ce type d'apprentissage est plus évolutif puisque l'ajout d'un autre flux ne nécessite pas un ré-apprentissage des autres flux.

Notons qu'une procédure d'adaptation des probabilités des transitions dans le MMC produit a été aussi testée, nous décrivons dans la section 3.3.6 cette procédure ainsi que son influence sur la reconnaissance.

Dans le cas d'un apprentissage indépendant, apprendre un modèle N-flux revient donc, dans un premier temps, à apprendre les paramètres de ses N MMC mono-flux. La deuxième étape d'apprentissage consiste à apprendre les poids de combinaison.

Les MMC de chaque flux sont entraînés par maximisation de la vraisemblance grâce à l'algorithme de Baum-Welch. L'apprentissage est embarqué : tous les modèles de caractères sont entraînés simultanément sur l'ensemble

des mots de la base d'apprentissage (voir section 2.2.4 du chapitre 1). Ce type d'apprentissage a deux atouts majeurs : les caractères sont modélisés en fonction de leurs contextes dans le mot, c'est à dire en tenant compte des différentes formes des caractères existantes dans la base. Le processus de segmentation des mots en caractères n'est pas indispensable pour effectuer l'apprentissage et la reconnaissance.

L'estimation des poids de combinaison est réalisée selon les deux stratégies décrites dans le paragraphe 1.8.4 :

- Approche de combinaison par poids égaux.
- Approche par fréquence relative

Notons que l'évaluation de ces deux approches dans le cadre de la reconnaissance de mots manuscrits a donné des résultats similaires. Les résultats présentés dans ce chapitre sont tous obtenus avec l'approche de combinaison par poids égaux vu qu'elle est plus simple à implémenter et que les tests que nous avons réalisés en utilisant l'approche par fréquence relative n'ont pas amélioré les performances.

L'étape suivante est la reconnaissance, dans laquelle les modèles des différents flux, déjà appris, sont simultanément décodés suivant le formalisme multi-flux (cf. paragraphe 1.8.3 du chapitre 1). Pendant cette phase, la séquence d'observations issue du mot à reconnaître est confrontée au réseau des mots du lexique composés de modèles de caractères. La séquence de caractères correspondant à la vraisemblance maximale identifie le mot reconnu.

3.3 Expérimentations et résultats

Nous avons décrit ci-dessus les différentes étapes de notre système de reconnaissance de mots manuscrits. L'approche proposée est indépendante de la nature du script, elle opère sans segmentation explicite en graphèmes, cette procédure diffère généralement d'un script à un autre. Des caractéristiques bas niveaux sont extraits sur les images du mot indifféremment de leur langue. Afin d'avoir un système plus robuste, différentes sources d'informations sont combinées à travers une approche multi-flux. Pour évaluer notre système de reconnaissance de l'écriture multi-scripts, différentes bases

de données sont utilisés pour les scripts Arabe et Latin. Nous décrivons dans la suite les bases de données utilisées et les expérimentations effectuées.

3.3.1 La base de mots Arabe IFN/ENIT

La base de données de référence IFN-ENIT [Pechwitz 02] est collectée et distribuée par l'IFN (Institute for Communications, de l'université de Braunschweig - Allemagne), et l'ENIT (Ecole Nationale d'Ingénieurs de Tunis). C'est la base la plus utilisée dans le domaine de la reconnaissance de mots manuscrits Arabes et a fait l'objet de deux compétitions internationales au cours de ICDAR'05 [Märgner 05] et ICDAR'07 [Märgner 07] afin de comparer les performances de reconnaissance des différents systèmes présentés. Elle comporte 32,492 images de noms de villes/villages tunisiens arabes écrits par 411 scribes différents formant un lexique de 946 mots. Le nom d'une ville peut contenir entre un et trois mots. Les exemples de la base sont partitionnés en 5 sous-ensembles : a, b, c, d, et e.

Les images sont scannées en 300 dpi et sont disponibles en format binaire. La figure 3.8 montre quelques exemples d'images d'un nom de ville/village de la base IFN-ENIT, écrits par différents auteurs. Pour plus d'informations, une description complète sur la base de référence IFN-ENIT est disponible dans [Pechwitz 02].

أولاد حفوز	أولاد حفوز	أولاد حفوز
أولاد حفوز	أولاد حفوز	أولاد حفوز
أولاد حفوز	أولاد حفوز	أولاد حفوز
أولاد حفوز	أولاد حفوز	أولاد حفوز

Figure 3.8. Exemples de la base de référence IFN-ENIT, d'après [Pechwitz 02]

Durant les expérimentations sur la base IFN/ENIT, 178 modèles de caractères sont construits pour modéliser :

- Les 28 caractères de base de l'alphabet arabe.
- Leurs variations de forme en fonction de leur position dans le mot (au début, au milieu ou à la fin).
- Les caractères avec la présence de marques diacritiques supplémentaires telles que le "Chadda" et le "Hamza".
- Les ligatures.

- Les chiffres figurant dans la base (i.e. 0, 1, 2, 6,7, 8, 9).

3.3.2 Résultats sur la base IFN/ENIT

Pour tester notre moteur de reconnaissance sur la base IFN/ENIT, nous nous sommes mis dans les mêmes conditions que la compétition ICDAR'05. Les sous-ensembles a, b, c et d sont utilisés pour l'apprentissage et le sous-ensemble e pour le test. Le tableau 3.1 présente les résultats de reconnaissance obtenus pour les quatre modèles : contour supérieur, contour inférieur et les densités avec deux tailles de fenêtre différentes (8 pixels pour densité 1 et 14 pixels pour densité 2). Ces résultats sont obtenus en utilisant un décodage classique 1D avec l'algorithme de Viterbi.

Modèles	TOP			
	1	2	5	10
1) Contour supérieur	70.5	78.6	86.3	90.4
2) Contour inférieur	63.5	73.1	82.6	86.4
3) Densité 1	65.1	73	80.6	83.2
4) Densité 2	68.7	78.1	83.3	86.9

Table 3.1. IFN/ENIT : Résultats de reconnaissance avec un seul flux

Les meilleurs résultats sont obtenus avec le modèle du contour supérieur avec 70.5%. Ceci s'explique par le fait que le contour supérieur est plus informatif pour la reconnaissance de mots manuscrits Arabes.

Pour améliorer ces performances, nous avons combiné 2 à 2 ces 4 sources d'informations dans le cadre d'une approche 2-flux. Partant de 4 flux, différentes combinaisons sont alors possibles, les résultats obtenus sont graphiquement illustrés sur la figure 3.9.

Les résultats montrent que l'approche multi-flux améliore les taux de reconnaissance comparée à l'approche 1D. Le meilleur résultat est obtenu en combinant les flux correspondant aux contour supérieur et Densité 2 avec un taux de reconnaissance de 79.6%. Le gain ici est de 9.1% comparé au meilleur taux obtenus par une approche 1D. Notons aussi que la combinaison de deux flux de nature différente (ici contour et densité) améliore mieux les résultats de reconnaissance que de combiner les deux contours ou les deux densités, d'où l'importance de la complémentarité des caractéristiques choisies.

Des expérimentations supplémentaires ont été menées afin de comparer

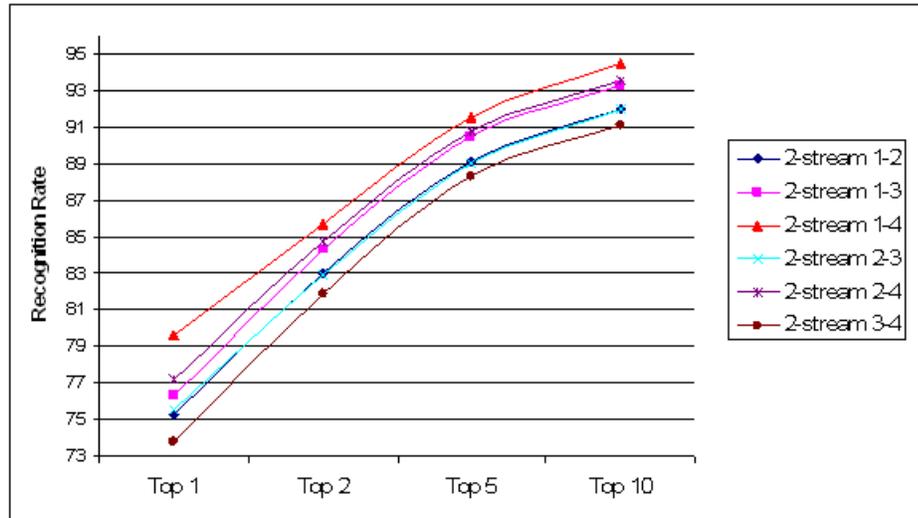


Figure 3.9. IFN/ENIT : Résultats de reconnaissance avec 2 flux

l'apport d'une approche multi-flux par rapport aux approches classiques de combinaison par fusion des représentations et par fusion des scores. Le tableau 3.2 présente les résultats obtenus pour ces différentes approches de combinaison.

Modèles	TOP			
	1	2	5	10
Multi-flux	79.6	85.7	91.6	94.5
Fusion des décisions	75.4	83.2	89.5	92.2
Fusion des représentations	74.1	82.6	88.4	90.8

Table 3.2. IFN/ENIT : Comparaison de l'approche multi-flux aux approches par fusion des décisions et des représentations

Les résultats montrent la supériorité de l'approche multi-flux par rapport aux approches classiques de combinaison. Notons que pour l'approche par combinaison des scores, la même règle de combinaison appliquée dans le formalisme multi-flux est utilisée (i.e somme pondérée).

3.3.3 Comparaison avec d'autres systèmes

Pour comparer nos résultats de reconnaissance aux autres systèmes travaillant sur la même base, nous nous sommes référés aux deux compétitions réalisées dans le cadre de la conférence internationale ICDAR.

3.3.3.1 La compétition ICDAR'2005

Dans le cadre de la conférence ICDAR'05, la première compétition internationale pour la reconnaissance automatique des mots manuscrits arabes [Märgner 05] a été organisée. Cinq groupes ont soumis leurs systèmes pour la compétition. Les systèmes participants ont été entraînés sur les quatre sous-ensembles a, b, c et d de la base de référence. Ces systèmes sont ensuite testés sur un nouveau sous ensemble nommé e contenant environ de 6033 images non connues par les groupes participants. Nous nous sommes placés dans les mêmes conditions de la compétition pour comparer nos résultats aux autres systèmes participants. Le tableau 3.3 montre les résultats obtenus par les cinq systèmes déposés. Les organisateurs de la compétition ont ajouté les résultats obtenus par leur système ARAB-IFN pour la comparaison. Ces résultats montrent bien la supériorité de notre système de reconnaissance qui dépasse le système ayant gagné la compétition [Al-Hajj 05] de 3.67% en Top 1.

Systèmes	Top 1	Top 5	Top 10
ICRA	65.74	83.95	87.75
SHOCRAN	35.70	51.62	51.62
TH-OCR	29.62	43.96	50.14
UOB	75.93	87.99	90.88
REAM	15.36	18.52	19.86
ARAB-IFN	74.69	87.07	89.77
Système proposé	79.6	91.6	94.5

Table 3.3. IFN/ENIT : Comparaison aux systèmes présentés dans la compétition ICDAR'05

3.3.3.2 La compétition ICDAR'2007

A ICDAR'07, et dans le même contexte qu'en 2005, une autre compétition a été organisée [Märgner 07]. Les différents systèmes participants ont été entraînés sur les 5 sous-ensembles a, b, c, d et e et ont été testés sur un nouveau sous-ensemble f. A ce jour, ce dernier ensemble f n'est pas publiquement disponible, c'est pourquoi nous ne pouvons pas comparer nos résultats de reconnaissance à ceux de la compétition ICDAR'07. Par contre, d'autres tests ont été réalisés sur les sous-ensembles d et e sachant qu'ils font déjà partie de la base d'apprentissage sur laquelle les systèmes ont été entraînés. Bien que ceci ne soit pas très significatif, mais c'est notre seul moyen de

se comparer aux résultats de la compétition ICDAR'07. Pour cela on a ré-entraîné notre système sur les 5 sous-ensembles a, b, c, d et e et on l'a testé sur les sous-ensembles d et e. Le tableau 3.4 présente une comparaison de notre résultat de reconnaissance à ceux reportés dans la compétition selon [Märgner 07].

Systemes	set d	set e
01	66.34	64.89
02	40.45	37.73
03	70.62	68.62
04	48.68	44.04
05	68.07	57.37
06	93.63	86.67
07	91.23	84.27
08	94.58	87.77
09	90.02	81.80
10	92.12	83.52
11	92.38	83.92
12	93.32	85.13
13	88.33	83.87
14	89.80	80.24
Systeme proposé	92.82	84.33

Table 3.4. IFN/ENIT : Comparaison aux systèmes présentés dans la compétition ICDAR'07 (ID 01 : MITRE ; IDs 02-04 : CACI ; ID 05 : CEDAR ; ID 06 : MIE ; IDs 07-08 : SIEMENS ; IDs 09-12 : UOB-ENST ; ID 13 : ICRA ; ID 14 : PARIS V)

Le système ayant gagné la compétition est celui de Siemens [Schambach 08] avec un taux de reconnaissance de 94.54% pour le sous-ensemble d et 87.77% pour le sous-ensemble e. Notre système se positionne quatrième parmi les 14 systèmes ayant participé à la compétition.

3.3.4 La base de mots Latin IRONOFF

La base de mots IRONOFF est collectée et distribuée par l'université de Nantes (IRESTE, France), en collaboration avec la société VISION OBJECTS. C'est une base de données duale en-ligne et hors-ligne . Elle a été collectée auprès d'environ 700 scripteurs différents, formée par 31346 mots manuscrits isolés sur un lexique de 197 mots (Français : 28657, Anglais : 2689). Les images sont en niveau de gris avec une résolution de 300 dpi. Pour tester notre système sur la base IRONOFF, nous avons binarisé les

images de la base en utilisant une méthode de seuillage global classique [Otsu 66]. 20898 images ont été utilisées pour l'apprentissage et 10448 pour le test.

Nous présentons dans le tableau 3.5 les résultats de reconnaissance obtenues par les 4 flux utilisant un décodage 1D.

Modèles	TOP			
	1	2	5	10
1) Contour supérieur	81.2	86.8	91.4	94.6
2) Contour inférieur	69.8	78.4	88.6	93.4
3) Densité 1	81.8	87.6	92.6	95.6
4) Densité 2	80	86.2	92.8	95.8

Table 3.5. IRONOFF : Résultats de reconnaissance avec un seul flux

Nous remarquons que les résultats de reconnaissance du contour inférieur sont moins bons comparés à ceux des autres flux, cela s'explique par le fait que le contour inférieur est généralement peu informatif sur les lettres Latines. Comparés aux résultats sur les mots arabes ou le contour supérieur donne le meilleur taux de reconnaissance, pour les mots Latins, il n'y a pas vraiment un flux qui surpasse les autres.

Pour évaluer l'apport de l'approche multi-flux, les quatre sources d'in-

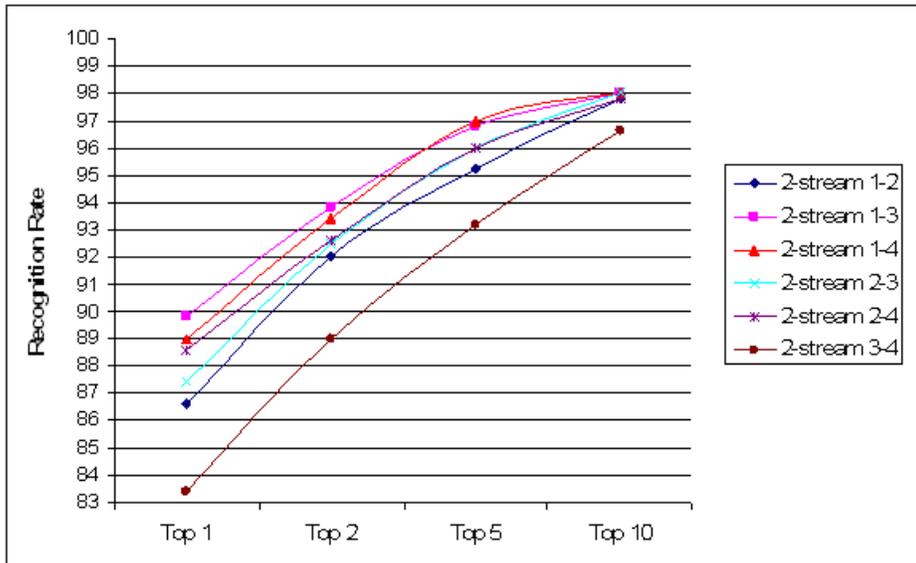


Figure 3.10. IRONOFF : Résultats de reconnaissance avec 2 flux

formations sont combinées 2 à 2 dans le cadre d'une approche 2-flux. Les résultats obtenus sont graphiquement illustrés sur la figure 3.10 présentant les différentes combinaisons possibles. Les résultats confirment l'apport de l'approche multi-flux observé durant les expérimentations réalisées sur la base IFN/ENIT de mots Arabes. Le meilleur taux de reconnaissance est de 89.8% en combinant les deux flux contour supérieur et densité 1. Le gain est de 8% par rapport aux résultats obtenus par une approche 1D. De la même façon que sur l'arabe, les résultats sur le Latin confirment que la complémentarité des flux permet d'améliorer les résultats.

Ces résultats sont comparés dans le tableau 3.6, aux résultats des approches classiques de combinaison par fusion de représentations et de décisions et montrent la supériorité de l'approche multi-flux avec un gain de 3.8% en Top 1.

Modèles	TOP			
	1	2	5	10
Multi-flux	89.8	93.8	96.8	98
Fusion des décisions	86	91.4	95	97.4
Fusion des représentations	85.2	90.6	94.4	97

Table 3.6. IRONOFF : Comparaison de l'approche multi-flux avec les approches par fusion des décisions et des représentations

Pour la base IRONOFF, il n'y a pas eu malheureusement des compétitions à l'échelle internationale. Pour cela, nous avons alors collecté les principaux résultats qui ont été reportés sur cette base afin de pouvoir les comparer à nos résultats de reconnaissance. Les taux de reconnaissance sont ainsi présentés dans le tableau 3.7 et confirment les bons résultats de notre système qui se place en deuxième position.

Systèmes	Top 1	Top 5
[Tay 01]	86.6	94.2
[Viard-Gaudin 05]	87.4	95.8
[Viard-Gaudin 05]	89.8	97
[Tay 01]	96.1	99.1
Système proposé	89.8	96.8

Table 3.7. IRONOFF : Comparaison avec les systèmes présentés dans la littérature

3.3.5 La base de mots Latin RIMES

Le programme Techno-Vision, initiative des ministères en charge de la recherche et de la défense, a pour but d'augmenter le nombre d'applications opérationnelles des techniques d'analyse d'images par ordinateur grâce à la mise en place de moyens d'évaluation quantitative de leurs performances. Ces moyens comprennent des bases de données d'images annotées, des protocoles et des outils d'évaluation ainsi que des métriques de calcul des performances. Dix projets ont ainsi été mis en place et ont débuté en 2004 pour s'achever pour la plupart en 2007. Parmi ces dix projets, le projet RIMES [Augustin 06] (Reconnaissance et Indexation de données Manuscrites et de fac similÉS) est dédié à la reconnaissance et à l'indexation de données manuscrites et de fac-similés.

Différentes tâches sont ainsi envisagées dans le cadre du projet RIMES :

- L'analyse de la structure du document.
- La reconnaissance du contenu des champs manuscrits, des plus élémentaires (chiffre, mot isolé) aux plus complexes (paragraphe, bloc) et plus généralement la reconnaissance de l'écriture connaissant ou non la structuration du document.
- La reconnaissance des logos.
- L'identification des scripteurs.
- L'extraction d'information de haut niveau comme l'identité du destinataire ou l'objet de la lettre

Dans ce cadre, nous avons participé avec notre système de reconnaissance de mots manuscrits à la deuxième passe d'évaluation de la campagne RIMES, et plus précisément à la tâche qui nous concerne qui est la reconnaissance de mots. Pour cela, nous avons disposé de 36,588 images de mots pour l'apprentissage, 7,786 pour la validation et 7542 pour le test. 2 tâches ont été proposées : un premier test avec un lexique de 100 mots aléatoirement choisis dans le dictionnaire complet et un deuxième test avec le lexique complet de 1636 mots. Trois systèmes ont participé à l'évaluation, notre système a eu les meilleurs résultats pour les deux tâches avec un taux de reconnaissance de 91.33% avec un lexique de 100 mots et 72.53% sur tout le lexique. Les résultats complets sont illustrés dans le tableau 3.8. Ces résultats seront publiés prochainement dans le cadre de la conférence internationale ICDAR'2009.

Systèmes	Lexique 100		Lexique 1636	
	Top 1	Top 10	Top 1	Top 10
Systeme 1	84.66	94.33	63.98	86.86
Systeme 2	46.67	77.19	74.81	92.5
Notre système	91.33	99.12	72.53	93.79

Table 3.8. Performances sur la base RIMES

3.3.6 Adaptation des probabilités de transition du MMC composite

Durant toutes les expérimentations précédentes, un apprentissage indépendant des différents flux est réalisé. Dans ce cas, on admet l'hypothèse d'indépendance des flux pour déterminer les paramètres du MMC produit, ce qui n'est pas toujours vrai. En plus, pour ce type d'apprentissage, les contraintes d'asynchronie ne sont utilisées que pour le décodage. Dans ce cadre, nous avons proposé une méthode d'adaptation des probabilités de transition du MMC composite permettant de tenir compte des contraintes sur la synchronie aussi bien pour l'estimation des paramètres que pour le décodage. Ainsi, cette adaptation nous permet de relâcher l'hypothèse d'indépendance des flux.

La procédure d'adaptation des probabilités de transition dans le MMC produit se fait selon l'algorithme suivant :

Algorithme 5: Algorithme d'adaptation

Tantque vraisemblance augmente faire

début

pour chaque mot de la base d'apprentissage faire

début

- construire le MMC produit du mot
- lancer Viterbi sur le MMC produit.
- cumuler la vraisemblance de l'exemple
- stocker la meilleure séquence d'états obtenue par Viterbi.

fin

ré-estimer les probabilités de transitions des modèles composites de caractères.

fin

Pour évaluer l'influence de cette approche d'adaptation, des expérimentations ont été réalisées sur la base IRONOFF dans le cadre d'une approche

2-flux. Les résultats du tableau 3.9 montrent un petit gain ($\leq 1\%$) du à la procédure d'adaptation pour les différentes combinaison des flux .

Modèles	Sans adaptation		Avec adaptation	
	Top 1	Top 5	Top 1	Top 5
2-flux 1)2)	86.6	95.2	87.2	95.3
2-flux 1)3)	89.8	96.8	90.2	96.4
2-flux 1)4)	89	97	89.3	97.2
2-flux 2)3)	87.4	96	87.9	96.2
2-flux 2)4)	88.6	96	89.2	96.1
2-flux 3)4)	83.4	93.2	83.7	93.7

Table 3.9. Performances avec et sans adaptation sur la base IRONOFF

3.3.7 Conclusion

A travers les différentes expérimentations réalisées tout au long de ce chapitre, nous avons pu montrer l'apport de l'approche multi-flux proposée dans le cadre de la reconnaissance de l'écriture manuscrite multi-script. L'approche a été testée sur différentes bases de mots publiques, pour les deux script Latin et Arabe. Ces tests ont montré que l'approche 2-flux améliore considérablement les résultats obtenus avec une approche mono-flux ainsi que les résultats des approches classiques par combinaison de décisions et de représentations.

Pour mieux étudier les possibilités offertes par cette approche, nous avons cherché à étendre le formalisme pour combiner plus de deux flux. Nous étudions alors dans la suite l'apport de cette extension ainsi que la complexité et les limites de l'approche.

3.3.8 Extension au cas N-flux

Combiner N flux de caractéristiques à travers une approche multi-flux revient à construire le MMC produit défini par ses états composites, ses probabilités de transitions et d'émissions.

Soit $S^i = \{s_j^i\}, j = 1 \dots n_i$ l'ensemble des états du modèle mono-flux i . Le MMC composite est défini dans l'espace produit des états notés S^\otimes comme

suit :

$$\begin{aligned} S^\otimes &= S^1 \otimes S^2 \otimes \dots \otimes S^N \\ &= \{s_i^\otimes = (s_j^1, s_k^2, \dots, s_l^N); j = 1 \dots n_1; k = 1 \dots n_2; l = 1 \dots n_N\} \end{aligned}$$

Dans ce modèle, la log-vraisemblance conditionnelle des données pour l'état S_i^\otimes est calculée par une fonction de somme pondérée :

$$\log P(X_1(t), X_2(t), \dots, X_N(t) | S_i^\otimes) = \sum_{\alpha=1}^N \omega_i^\alpha \log P(X_\alpha(t) | f_\alpha(S_i^\otimes))$$

f_α est une fonction qui désigne l'état s_k^α du MMC du flux α et correspondant à l'état produit S_i^\otimes tel que $f_\alpha(S_i^\otimes) = s_k^\alpha$.

$(X_1(t), X_2(t), \dots, X_N(t))$ désigne la séquence des observations des N flux. Les probabilités de transitions de chaque état du modèle produit sont déterminées à partir des probabilités de transitions des états correspondants dans le modèle parallèle par un produit des probabilités de transition grâce à l'hypothèse d'indépendance des flux entre les états de recombinaison.

$$P(S_i^\otimes | S_j^\otimes) = \prod_{\alpha=1}^N P(f[\alpha](S_i^\otimes) | f[\alpha](S_j^\otimes))$$

Il est simple de vérifier que le nombre d'états dans le MMC composite croît exponentiellement avec le nombre de flux N puisqu'on travaille dans l'espace produit des états, ce qui influe sur la complexité de l'approche. En effet, la complexité de l'algorithme de Viterbi est de $O(T.(C.N)^2)$, N étant le nombre d'états par modèle de caractère, C est le nombre de caractères dans le modèle et T le nombre de trames d'observations par état. Dans le MMC produit, le nombre d'états par modèle de caractère croît en fonction du nombre de flux et est égal à $C.N^S$, S étant le nombre de flux. La complexité de l'algorithme de décodage devient alors en $O(T.(C.N^S)^2) = O(T.C^2.N^{2S})$. En conséquence, la complexité d'un algorithme de décodage multi-flux croît exponentiellement avec le nombre de flux.

Vu la complexité de la tâche, nous avons choisi de tester l'approche sur un lexique réduit. Les premières expérimentations sont réalisées sur le sous-ensemble IRONOFF-Chèque de la base IRONOFF, restreint à un petit lexique d'environ 30 mots utilisés pour exprimer les montants de chèques français. On dispose de 7956 images de mots pour l'apprentissage et de 3987

images pour le test. Les résultats obtenus avec 2, 3 et 4 flux sont donnés dans le tableau 3.10.

Modèles	4 e/c		3 e/c	
	Top 1	Top 5	Top 1	Top 5
2-flux 1)2)	86	95.4	87	97.6
2-flux 1)3)	92.8	99.2	91	99.4
2-flux 1)4)	92.2	99	90.4	99
2-flux 2)3)	94	99.6	89.4	97.8
2-flux 3)4)	91	99	86.2	97.2
3-flux1)2)3)	94	99.6	91.6	99.6
3-flux1)3)4)	92.6	99.2	90.8	98.8
3-flux2)3)4)	93.4	99	89.4	98.4
3-flux1)2)4)	93.6	99.8	92	99
4-flux	94.2	99.2	92.8	99.2

Table 3.10. Performances sur la base IRONOFF-Chèque

Dans les mêmes conditions, d'autres expérimentations ont été réalisées sur la base de mots Arabes IFN/ENIT. Les résultats obtenus sur un lexique réduit de 30 mots, puis de 100 mots sont reportés dans le tableau 3.11. La génération des lexiques est réalisée par tirage aléatoire : pour chaque mot à reconnaître, un lexique de taille N est obtenu par un tirage aléatoire de N-1 mots parmi le lexique complet, complété par l'étiquette du mot à reconnaître. Les résultats obtenus montrent que l'ajout d'un flux améliore généralement

Modèles	Taille du lexique			
	30		100	
	Top 1	Top 5	Top 1	Top 5
1) Contour sup.	97.8	99.8	95.4	99
2) Contour inf.	97.8	99.8	93.4	99.2
3) Densité1	92	97.4	88.4	94
4) Densité2	92.6	98	88.8	94.4
2-flux 1)2)	98	99.8	95.6	99.8
2-flux 1)3)	98.6	99.8	98.4	99.4
2-flux 1)4)	98.8	100	98.6	99.8
2-flux 2)3)	98	100	96.8	99.8
2-flux 3)4)	97.6	99.6	96.4	98.6
3-flux1)2)3)	98.6	99.8	98.4	99.8
4-flux	99.2	99.8	98.8	100

Table 3.11. Performances sur la base IFN/ENIT

la performance du système. Cependant, ceci augmente considérablement la

complexité de l'algorithme de décodage. Pour réduire la complexité, et du coup, accélérer le processus de décodage qui dépend fortement du nombre d'états, des solutions sont envisageables :

1. Réduire le nombre d'états par caractère dans les MMCs parallèles : par exemple, réduire le nombre d'états par caractère dans les MMC parallèles de 4 à 3 états permet d'obtenir des MMC composites à 9 états par caractère au lieu de 16 dans le cas de 2 flux, à 27 états par caractère au lieu de 64 dans le cas de 3 flux... Cette réduction a été testée et les résultats avec 3 états par caractère (3e/c) sont donnés dans le tableau 3.10. Ces résultats montrent que la réduction du nombre d'états par caractère accélère le décodage mais décroît les taux de reconnaissance.
2. Limiter le degré d'asynchronie entre les flux de données en restreignant le produit aux seuls états (produits) valides : on mesurera l'asynchronie, dans un état produit comme la valeur absolue de la différence entre les numéros des états des éléments du produit.
3. Accélérer la procédure de décodage en utilisant d'autres algorithmes proposés dans la littérature [[Koerich 02](#), [Rabiner 89](#)] tels que l'algorithme level-building guidé par le lexique, l'algorithme de décodage rapide à deux niveaux, l'algorithme A*...

3.4 Conclusion

A travers les différentes expérimentations réalisées tout au long de ce chapitre, nous avons pu montrer la performance de notre système de reconnaissance de l'écriture multi-script qui a donné des résultats de reconnaissance comparables à ceux des meilleurs systèmes présentés dans la littérature que ce soit sur le script Latin ou sur l'Arabe. L'originalité de l'approche proposée est que le système opère indépendamment de la nature du script, et les différentes étapes de prétraitements, d'extraction des caractéristiques et de reconnaissance sont pratiquement identiques quel que soit le script. Afin de mieux mettre en valeur l'aspect multi-script de notre système, nous l'avons testé sur une base de mots multi-lingue composée de mots Arabes et Latins. Nous présentons dans la section suivante l'architecture complète du système de reconnaissance multi-script ainsi que les résultats obtenus.

3.4.1 Performance du système de reconnaissance sur une base multi-lingue

Lors des expérimentations déjà réalisées dans les sections précédentes, la principale différence entre le traitement des scripts Arabe et Latin est le sens de l'extraction des caractéristiques qui est établi suivant le sens de l'écriture (gauche-droite pour le Latin et droite-gauche pour l'Arabe).

Afin de proposer un système multi-script dont l'image du mot d'entrée est de script inconnu, l'idée est de choisir un sens unique d'extraction des caractéristiques (gauche-droite dans notre cas) et de construire un lexique multi-lingue formé par les mots Latins et les mots Arabes dont on inverse l'ordre des caractères. Ainsi, le système construit les modèles de mots Arabes dans le sens inverse de leur lecture.

L'architecture complète du système de reconnaissance de l'écriture multi-script est donnée par la figure 3.11. Les étapes de prétraitements, d'extraction des caractéristiques et de décodage sont les mêmes que dans les paragraphes précédents.

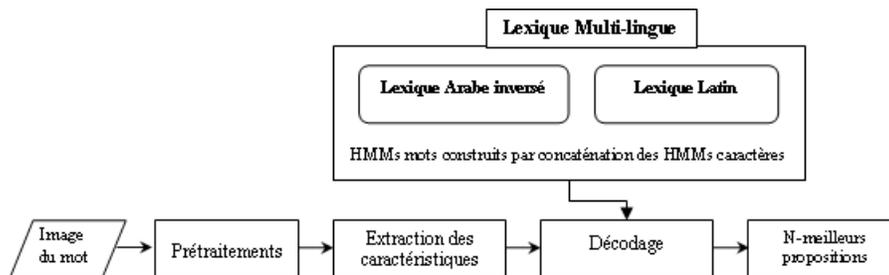


Figure 3.11. Architecture du système de reconnaissance multi-script

Pour tester le système de reconnaissance multi-script proposé, nous avons construit une base de test multi-lingue formée par des mots Latins et Arabes acquis des bases IRONOFF et IFN/ENIT (1000 mots de chaque base). A noter que l'apprentissage des modèles a été effectué séparément pour chaque script. Nous présentons dans le tableau 3.12 les taux de reconnaissance obtenus pour trois tests réalisés sur un lexique multi-lingue formé par 1142 mots dont 946 mots Arabes et 196 mots Latins :

- Test 1 : En considérant tous les mots de la base multi-lingue.
- Test 2 : En ne considérant que les mots Arabes de la base multi-lingue.
- Test 3 : En considérant que les mots Latins de la base multi-lingue.

Nous remarquons que les résultats obtenus en considérant que les mots

Tests	Top			
	1	2	5	10
Test 1	84.5	90.5	94.6	96.3
Test 2	80.2	88	93.4	95.2
Test 3	88.8	93	95.8	97.4

Table 3.12. Résultats de reconnaissance sur la base multi-lingue

Latins (test 3) sont meilleurs de ceux du test 2, ceci peut s'expliquer par le nombre plus petit de mots Latins (196 mots) dans le lexique multi-lingue comparé au nombre de mots Arabes (946 mots). D'autre part, le bon taux de reconnaissance obtenu dans le test 1 en considérant toute la base multi-lingue montre que le système ne tombe pas souvent dans la confusion et arrive souvent à bien identifier les deux scripts.

Afin de mieux analyser ce dernier point, nous définissons un taux de confusion Arabe comme étant le pourcentage de mots Arabes qui ont été reconnus comme des mots Latins par le système de reconnaissance multi-script. Inversement, le taux de confusion Latin est le pourcentage de mots Latins qui ont été reconnus comme des mots Arabes par le système de reconnaissance multi-script. A partir de ces deux taux, on en déduit le taux de confusion multi-script qui n'est que la somme de ces deux derniers taux. Nous reportons dans le tableau 3.12 ces différents taux de confusion.

Taux de confusion	Top			
	1	2	5	10
Multi-script	1.5	3.7	7.4	11.8
Arabe	0.3	0.3	0.6	0.8
Latin	1.2	3.4	6.8	11

Table 3.13. Taux de confusion sur la base multi-lingue

Les résultats du tableau 3.13 montrent que le système arrive à bien identifier les scripts dans la plupart des cas, avec un taux de confusion de 1.5% seulement en rang 1. Nous reportons dans la figure 3.12 quelques exemples de confusion du système de reconnaissance multi-script. Ces confusions peuvent être expliquées par la présence d'un espace inter-mot dans le mot de test et le mot reconnu comme pour les exemples 1 et 3, ou par la présence de boucles comme dans l'exemple 2 ou encore par la mauvaise qualité d'écriture du mot de test comme les exemples 5 et 6, ou par la présence d'un chevauchement

entre les lettres comme dans l'exemple 4, dans ce cas le système n'arrive pas à bien aligner les observations sur les mots du lexique et retourne un mot erroné.

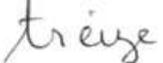
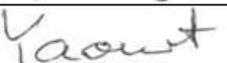
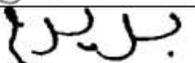
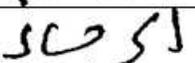
N°	Image de test	Mot reconnu
1		بزيمة
2		حلما
3		شظ السلام
4		tout
5		SOUS
6		SIX

Figure 3.12. Exemples de confusion du système de reconnaissance multi-script

3.5 Conclusion

Dans ce chapitre, nous avons décrit les différentes étapes de notre système de reconnaissance multi-script. Le système proposé opère indépendamment de la nature du script, pour cela il procède sans segmentation explicite ou la détermination des points potentiels de segmentation dépend généralement de la nature du script. D'autre part, nous avons opté pour une modélisation markovienne qui convient bien à une approche multi-script étant donné que ces modèles sont bien adaptés pour la modélisation des langues alphabétiques ainsi qu'aux différentes variations morphologiques des caractères.

Les premiers tests ont été fait en utilisant des modèles mono-flux, ceci a montré que le même système arrive à reconnaître aussi bien des mots Arabes que Latins.

Afin d'améliorer les performances obtenues par le système mono-flux, un deuxième système multi-flux combinant plusieurs sources d'informations à été proposé. L'évaluation de ce système a montré que l'approche multi-flux améliore considérablement les résultats de reconnaissance aussi bien pour le Latin que pour l'Arabe. On parvient à des taux de reconnaissance qui sont comparables à ceux des meilleurs systèmes proposés dans la littérature.

Ces expérimentations ont été complétées par des tests sur une base de mots multi-lingue, et les résultats obtenus ont montré que le système de reconnaissance de l'écriture multi-script proposé arrive à bien identifier et reconnaître des mots multi-lingues avec un taux de confusion très faible.

Ainsi, nous avons pu proposer un système de reconnaissance de l'écriture multi-script à base de modèles markoviens probabilistes. Ces approches offrent de très bonnes performances et ont été bien exploitées dans le cadre de ce travail. Outre la théorie des probabilités largement exploitée depuis des décennies, la théorie des fonctions de croyance, plus récente, offre des outils puissants de gestion de l'incertitude, de l'imprécision et du conflit sur les données ainsi que des méthodes de modélisation et de combinaison crédibilistes facilitant l'interprétation et l'intégration de connaissances. Ces possibilités vont être étudiées dans le chapitre suivant toujours dans le cadre de la reconnaissance de l'écriture multi-script.

Chapitre 4

Application du Modèle des Croyances Transférables à la combinaison d'informations multi-flux

4.1 Introduction

Dans le domaine de la reconnaissance des formes, nous manipulons des informations, qui sont censées donner une image aussi fidèle que possible de la réalité. Or, le plus souvent, ces informations sont imparfaites : imprécises, incertaines, vagues, incomplètes,...

D'autre part, si nous disposons de plusieurs sources d'informations, il est relativement fréquent qu'une source favorise l'événement A , tandis qu'une autre favorise un événement B différent de A . On parle alors de situation de conflit. Dans la théorie des probabilités, on ne dispose pas d'outils spécifiques formalisant la notion de conflit.

La théorie des fonctions de croyances, proposée par Shafer en 1976 est un cadre formel générique pour la représentation des connaissances disposant de règles de combinaison conçues pour la fusion de données.

Dans ce chapitre, nous commençons par présenter les mécanismes de base des fonctions de croyance ainsi que les outils permettant de gérer des problèmes plus complexes de fusion d'information. Ensuite, nous décrivons comment nous avons appliqué les fonctions de croyance pour combiner deux sources

d'informations à travers les modèles de Markov multi-flux dans le cadre de la reconnaissance de mots manuscrits. Nous terminons le chapitre en décrivant le principe des modèles de Markov crédibilistes.

4.2 Outils de base des fonctions de croyance

La théorie des fonctions de croyance a été développée par Shafer en 1976 à la suite des travaux de Dempster sur les probabilités inférieures et supérieures. Philippe Smets [Smets 90, Smets 93, Smets 94] a ensuite énormément contribué au développement de cette théorie grâce à son modèle des croyances transférables MCT, dérivé de celui de Shafer.

Le MCT est basé sur la définition de fonctions de croyance fournies par des sources d'information pouvant être complémentaires, redondantes et éventuellement non-indépendantes. Il propose un ensemble d'opérateurs permettant de combiner ces fonctions. Il est donc naturellement employé dans le cadre de la fusion d'informations pour améliorer l'analyse et l'interprétation de données issues de sources d'informations multiples.

Les mécanismes de raisonnement du MCT peuvent être regroupés en deux niveaux tel qu'illustrés sur la figure 4.1 :

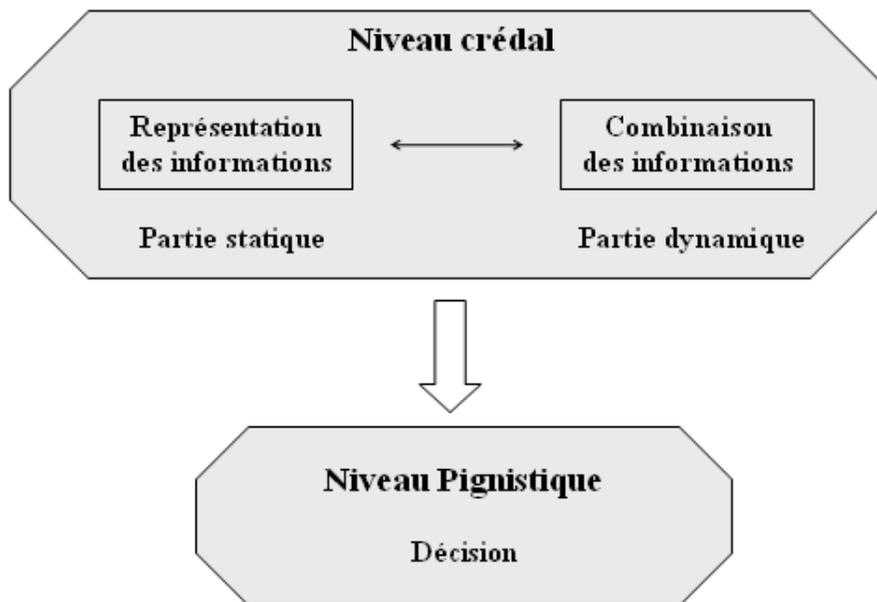


Figure 4.1. Représentation abstraite des mécanismes dans le Modèle des Croyances Transférables.

- Le niveau crédal comporte deux phases. Une phase de représentation des informations (partie statique) et une phase de combinaison et de raisonnement sur ces connaissances (partie dynamique).
- Le niveau pignistique qui sert à prendre une décision en prenant éventuellement en compte le résultat de la combinaison dans le premier niveau et le risque d’effectuer certaines actions suite aux décisions.

Cette décomposition en deux niveaux différencie le MCT des autres théories et méthodes, en particulier la théorie des probabilités au sein de laquelle la représentation des connaissances, la combinaison et la décision sont effectuées sans distinguer les niveaux d’abstraction. Nous décrivons dans la suite ces deux niveaux d’abstraction en nous référant aux présentations faites dans [Ramasso 07b, Klein 08].

4.2.1 Masse de croyance élémentaire

Soit un problème donné pour lequel un agent doit choisir une solution parmi un ensemble fini de solutions. Ces solutions, ou hypothèses ω_i , sont supposées mutuellement exclusives¹ et exhaustives. L’ensemble des hypothèses constituant le cadre de discernement est noté $\Omega = \{\omega_1, \dots, \omega_K\}$. L’ensemble des sous-parties de Ω est noté 2^Ω . Toute l’information qui permet à un agent de construire ses croyances vis à vis du problème est appelé **élément d’évidence** Ev . Un élément d’évidence correspond au fait *qu’une source S déclare une proposition E vraie et que l’agent construisant ses croyances reconnaît la validité des informations fournies par S* . Dans la suite du document, nous ferons souvent l’amalgame entre une source d’information et l’élément d’évidence d’un agent.

L’agent peut affecter une quantité de croyance à chaque élément A de 2^Ω . On appelle **masse de croyance** ou **fonction de masse**(FDM) la fonction représentant cette assignation. Elle est notée $m^\Omega [Ev] (\cdot)$, ainsi, $m^\Omega [Ev] (A)$ symbolise le crédit attribué à l’événement $A \subseteq \Omega$. Pour plus de simplicité, la notation $[Ev]$ sera généralement omise, de même que celle précisant le cadre de discernement sur lequel est définie la fonction.

1. une seule solution est possible

Une fonction de masse respecte par définition les contraintes suivantes :

$$m(A) : 2^\Omega \rightarrow [0, 1] \quad (4.1)$$

$$\sum_{A \subseteq \Omega} m(A) = 1 \quad (4.2)$$

Éléments particuliers

Parmi les éléments de 2^Ω , certains jouent un rôle particulier :

- Un ensemble A tel que $m(A) > 0$ est appelé **élément focal**. Nous noterons \mathcal{F}_m^Ω l'ensemble des éléments focaux d'une FDM $m(\cdot)$.
- L'ensemble vide \emptyset : Il représente la possibilité que la solution au problème ne soit pas contenue dans Ω . Certains auteurs considèrent qu'aucune masse ne doit être attribuée à cette hypothèse en considérant que, le cas échéant, le problème a été mal posé. On parle alors de **monde fermé**. A l'inverse si cette hypothèse est prise en compte, ce qui revient à autoriser $m(\emptyset) > 0$, on parle de **monde ouvert**. Le MCT préconise l'usage d'un monde ouvert afin de rester le plus générique possible. De plus, quand deux sources sont conflictuelles et combinées de façon conjonctive (cf. section 4.2.3 pour une définition de la conjonction), du crédit est attribué à l'hypothèse \emptyset . Certains auteurs appellent donc cette hypothèse "**conflit**", par léger abus de langage. La gestion du conflit fait partie des justifications de l'utilisation des fonctions de croyance, d'où un second intérêt à travailler en monde ouvert. Les avantages des mondes fermés ou ouverts ont été largement discutés par Smets [Smets 90].
- L'ensemble Ω : Il représente le fait que la solution au problème soit contenue dans Ω , mais qu'il soit impossible de déterminer quelle hypothèse est la bonne. Ainsi $m(\Omega)$ permet de quantifier l'ignorance face au problème. Par la suite, nous ferons référence par abus de langage à Ω en tant qu'**ignorance**, quand Ω est vu en tant qu'hypothèse et non en tant qu'ensemble.

Quelques exemples particuliers de FDM sont à noter 4.2 :

- **Fonction de masse vide** : l'unique élément focal est Ω , i.e. $m(\Omega) = 1$.
- **Fonction de masse simple** : il existe au plus deux éléments focaux dont Ω .

- **Fonction de masse dogmatique** : la proposition Ω n'est pas un élément focal, i.e. $m(\Omega) = 0$.
- **Fonction de masse normale** : la masse sur l'ensemble vide est nulle, i.e. $m(\emptyset) = 0$
- **Fonction de masse consonante** : une fonction de masse est dite consonante si et seulement si l'inclusion est une relation d'ordre totale sur ses éléments focaux. Si ses éléments focaux sont $\{A_i\}_{i=1}^n$, il existe une permutation $perm(\cdot)$ sur les indices telle que : $A_{perm(1)} \subseteq A_{perm(2)} \subseteq \dots \subseteq A_{perm(n)}$.
- **Fonction de masse bayésienne** : une fonction de masse $m(\cdot)$ est bayésienne si et seulement si tous ces éléments focaux sont des singletons $\{\omega_i\}$. En d'autres termes, $\forall A \subseteq \Omega$, tel que $|A| > 1, m(A) = 0$. Les fonctions de croyance bayésiennes sont des distributions de probabilités.

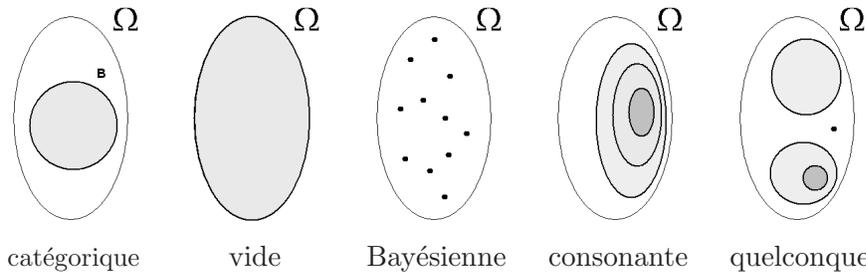


Figure 4.2. Représentation des éléments focaux de certaines classes de fonctions de croyance.

4.2.2 Transformations de la fonction de masse

A partir de la fonction de masses $m(\cdot)$, d'autres fonctions en relation biunivoque peuvent être déduites : elles représentent la même information mais sous une forme différente. Ces fonctions ont diverses interprétations mais en ce qui nous concerne, nous les utiliserons essentiellement pour simplifier les calculs. Elles sont toutes définies de $2^\Omega \rightarrow [0, 1]$ mais la somme des éléments focaux n'est pas forcément unitaire. Les fonctions élémentaires les plus utilisées sont les suivantes :

- **Plausibilité pl** : la plausibilité d'un élément B est la part maximale de croyance qui pourrait soutenir B :

$$pl(B) = \sum_{C \cap B \neq \emptyset} m(C), \forall B \subseteq \Omega$$

La relation inverse la suivante :

$$m(B) = \sum_{C \subseteq B} (-1)^{|B|-|C|+1} pl(\bar{C}), \forall B \subseteq \Omega$$

- **Crédibilité (ou croyance) *bel*** : la crédibilité d'un élément B est la part de croyance spécifiquement (sans le conflit) allouée à B :

$$bel(B) = \sum_{C \subseteq B, C \neq \emptyset} m(C), \forall B \subseteq \Omega$$

et l'on a la relation inverse suivante :

$$m(B) = \sum_{C \subseteq B} (-1)^{|B|-|C|} bel(C), \forall B \subseteq \Omega$$

- **Implicabilité *b*** : l'implicabilité d'un élément B est la somme des masses allouées aux sous-ensembles de B tels que leur véracité implique la véracité de B :

$$b(B) = \sum_{C \subseteq B} m(C) = bel(B) + m(\emptyset), \forall B \subseteq \Omega$$

$$m(B) = \sum_{C \subseteq B} (-1)^{|B|-|C|} b(C), \forall B \subseteq \Omega$$

- **Communalité *q*** : la communalité d'un élément B représente la somme des masses allouées aux sur-ensembles de B et donc qui ont B en commun :

$$q(B) = \sum_{B \subseteq C} m(C), \forall B \subseteq \Omega$$

$$m(B) = \sum_{B \subseteq C} (-1)^{|C|-|B|} q(C), \forall B \subseteq \Omega$$

4.2.3 Règles de combinaison

Le formalisme des fonctions de croyance offre la possibilité de fusionner différentes sources d'information, représentées par différentes fonctions de masses, à l'aide d'opérateurs appelés règles de combinaison. Si l'on note une

règle générique \odot , elle doit être telle que :

$$\odot : \mathfrak{B}^\Omega \times \mathfrak{B}^\Omega \longrightarrow \mathfrak{B}^\Omega \quad (4.3)$$

$$(m_1, m_2) \longrightarrow m_{1\odot 2} \quad (4.4)$$

La première règle de combinaison fut introduite par Dempster dans le cadre d'un monde fermé pour combiner des sources distinctes et fiables. La règle de Dempster est notée \oplus et se définit ainsi :

$$\forall A \neq \emptyset, m_1 \oplus m_2(A) = \frac{1}{1 - \kappa} \sum_{B, C / B \cap C = A} m_1(B) m_2(C) \quad (4.5)$$

$$\kappa = \sum_{B, C / B \cap C = \emptyset} m_1(B) m_2(C) \quad (4.6)$$

$$m_1 \oplus m_2(\emptyset) = 0 \quad (4.7)$$

Le terme κ est en réalité l'expression du conflit. Dans le monde fermé sa valeur est rejetée, et comme il n'y a aucune raison pour attribuer cette valeur à une hypothèse plutôt qu'à une autre, elle est répartie identiquement sur l'ensemble des hypothèses par normalisation. La règle a été adaptée au monde ouvert par Smets. Elle est alors appelée règle conjonctive et notée \odot . Sa définition se simplifie et devient :

$$m_1 \odot m_2(A) = \sum_{B, C / B \cap C = A} m_1(B) m_2(C) \quad (4.8)$$

Notons que l'utilisation des communalités (q) réduit la règle de combinaison conjonctive à un produit membre à membre simplifiant ainsi les calculs.

$$q_1 \odot q_2(A) = q_1(A) q_2(A) \quad (4.9)$$

La règle disjonctive vient en complément de la règle conjonctive, elle est d'une certaine manière l'opération duale, utilisée généralement pour combiner des sources distinctes et non-fiables. La règle disjonctive, notée \oslash , redistribue le crédit en provenance des deux FDM fusionnées en s'appuyant sur l'union des éléments focaux (au lieu de l'intersection pour la règle conjon-

tive) :

$$m_{\odot}(A) = \sum_{B,C|B \cup C = A} m_1(B) m_2(C) \quad (4.10)$$

Notons que l'utilisation des implicabilités (b) permet de réduire la règle de combinaison disjonctive à un produit membre à membre.

$$b_1 \odot b_2(A) = b_1(A) b_2(A) \quad (4.11)$$

Remarque

Les règles de combinaison conjonctive ou disjonctive sont généralement utilisées sous l'hypothèse que les sources de croyance sont distinctes. Une règle de combinaison conjonctive dite prudente (CCRC : Cautious Conjunctive Rule of Combination), et une règle de combinaison disjonctive hardie (BDRC : Bold Disjunctive Rule of Combination), ont été proposées par Denoeux [Denoeux 08] pour combiner des sources non-distinctes. D'autres règles de combinaison sont présentées dans [Klein 08].

4.2.4 Prise de décision à partir de fonctions de croyance

Dans le cadre du MCT, il est intéressant d'utiliser les fonctions de croyance pour gérer des données imprécises et mettre au point des processus complexes de fusion. En revanche pour prendre une décision, les probabilités sont mieux adaptées, car à ce stade l'imprécision ne fait que bruite le traitement et doit être éliminée, d'où l'intérêt de se doter d'une transformation permettant le passage des croyances vers les probabilités.

Le traitement de l'information au niveau crédal doit aboutir à la création d'une fonction de masse $m_{\odot}(\cdot)$ synthétisant l'ensemble des connaissances à l'instant où doit être prise la décision. Pour ramener m_{\odot} vers un espace probabiliste, Smets [Smets 90] se base alors sur le principe de la raison insuffisante. Ce principe stipule qu'en l'absence de raison de privilégier une hypothèse plutôt qu'une autre, les hypothèses sont toutes supposées équiprobables. Ainsi, pour tout élément focal A de $m_{\odot}(\cdot)$ la masse sera redistribuée uniformément sur les éléments de A . La transformation pignistique

$BetP$ se définit alors :

$$BetP(\{\omega_1\}) = \sum_{A \subseteq \Omega} \frac{|\omega_1 \cap A|}{|A|} \frac{m_{\odot}(A)}{1 - m_{\odot}(\emptyset)} \quad (4.12)$$

La formule est donnée pour $\{\omega_1\}$ et $m_{\odot}(\cdot)$ mais est valable pour tout ensemble et toute FDM. La décision est généralement prise en choisissant l'élément ω_k possédant la plus grande probabilité pignistique. Notons qu'à travers ce procédé, une seule distribution de probabilité est associée à une fonction de masse, tandis qu'une infinité de fonctions de masse est associée à une distribution de probabilité. Cela reflète d'une part que l'espace des fonctions de masse est bel et bien plus vaste que celui des distributions de probabilité, mais aussi d'autre part que la transformation pignistique occasionne une perte d'information.

4.3 Opérations avancées sur les fonctions de croyance

Le formalisme des fonctions de croyance contient de nombreux autres outils mathématiques. Nous nous limitons à présenter quelques uns d'entre eux que nous appliquons dans la suite de ce travail.

4.3.1 Combinaison de fonctions de croyance définies sur un cadre de discernement différent

Dans le paragraphe 4.2.3, nous avons présenté quelques méthodes de combinaison de fonctions de croyance dont le rôle est de prendre en compte toutes les informations disponibles afin d'améliorer la prise de décision. Ces mécanismes sont basés sur l'hypothèse que les fonctions de croyance sont définies sur le même espace de discernement, mais cette contrainte limite les applications pratiques.

Le MCT propose des outils performants et adaptés lorsque les cadres de discernement sont différents. Afin d'illustrer l'utilisation de ces outils, nous considérons le cas de la combinaison de deux fonctions de croyance : la première est m_1^{Ω} définie sur un premier cadre Ω , la seconde m_2^{Θ} est définie sur Θ , un autre cadre de discernement.

Le principe général de ces mécanismes consiste en trois étapes :

1. Construire le cadre de discernement commun.
2. Exprimer les croyances sur le cadre de discernement commun obtenu à l'étape 1.
3. Combiner les fonctions de croyance obtenues à l'étape 2 par les règles de combinaison classiques.

Les deux mécanismes de base qui permettent le changement d'espace sont le raffinement, où l'espace devient plus détaillé, et le grossissement, où l'espace devient plus grossier. Ici, nous allons utiliser un raffinement pour définir les deux fonctions de croyance sur un espace commun qui est l'espace produit.

Raffinement

Un raffinement associe à chaque hypothèse $\omega_k \in \Omega$ une ou plusieurs hypothèses $\Theta_i \in \Theta$. L'application $\rho : 2^\Omega \rightarrow 2^\Theta$ est un raffinement de Ω si et seulement si ρ vérifie les contraintes suivantes :

- L'ensemble $\{\rho(\{\omega_k\}), \omega_k \in \Omega\} \subseteq 2^\Theta$ est une partition de Θ .
- $\forall B \subseteq \Omega, \rho(B) = \bigcup_{\omega_k \in B} \rho(\{\omega_k\})$

La figure 4.3 présente un exemple de raffinement où le cadre Θ est la version raffinée de Ω . Les flèches partant de l'ensemble Ω et allant vers Θ décrivent le raffinement.

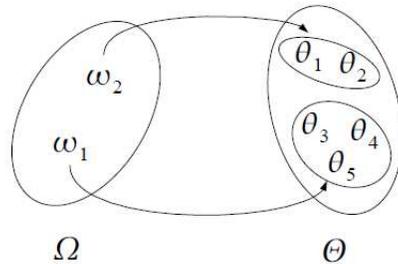


Figure 4.3. Raffinement de cadre de discernement : Θ est un raffinement de Ω

Revenons à notre exemple, nous construisons un nouvel espace produit défini par : $\Omega \times \Theta = \{(\omega_k, \theta_i), \forall k \in \{1 \dots |\Omega|\}, \forall i \in \{1 \dots |\Theta|\}\}$.

Ainsi nous définissons un nouveau cadre de discernement produit qui sera le raffinement des deux cadres de discernement Ω et Θ .

Après le raffinement des cadres de discernement Ω et Θ en $\Omega \times \Theta$, l'étape suivante est de redéfinir sur $\Omega \times \Theta$ la distribution de masses de croyance initialement définie sur Ω et Θ . Pour cela, on utilise l'**extension vide** [Shafer 76, Smets 93] qui est un processus conservatif de réallocation des masses de croyance. L'opérateur associé est noté \uparrow .

Par exemple, m_1^Ω est redéfinie sur $m_1^{\Omega \times \Theta}$ de la manière suivante :

$$m_1^{\Omega \uparrow \Omega \times \Theta}(C) = \begin{cases} m_1^\Omega(B) & \text{si } C = B \times \Theta \\ 0 & \text{sinon} \end{cases}$$

La masse initialement allouée à la proposition $B \subseteq \Omega$ est réallouée à $B \times \Theta$ après extension vide minimisant ainsi tout a priori sur Θ (aucune hypothèse de Θ n'est favorisée car la source de croyance fournissant m_1 ne connaît rien de Θ). De la même manière m_2^Θ est redéfinie sur $m_1^{\Omega \times \Theta}$ et la combinaison des deux fonctions de croyance est maintenant possible vu qu'elles sont définies sur le même cadre de discernement produit.

4.3.2 Construction de modèles de fonctions de croyance à partir des fonctions de vraisemblances

En pratique, construire une fonction de croyance représentant les informations disponibles n'est pas chose aisée, et il ne semble pas exister de solution universelle.

En reconnaissance des formes, différentes méthodes de construction de fonctions de croyance à partir d'un ensemble d'apprentissage ont été proposées [Vannoorenberghe 03]. Ces méthodes d'affectation ont été définies à partir du vecteur forme de l'objet à identifier, par le calcul d'une distance [Denoeux 95, Denoeux 00] ou d'une vraisemblance [Walley 99, Appriou 91]. Notons aussi l'existence de méthodes d'affectation basées sur l'élicitation d'avis d'experts [Yaghlane 06]. Ces différentes familles de modèles diffèrent selon la façon de coder l'incertitude sur la classe d'appartenance des exemples de l'ensemble d'apprentissage.

Dans le cadre de notre application, nous nous intéressons aux approches d'affectation à partir de vraisemblances pour transformer en fonction de masse les densités de probabilité d'émission des observations conditionnellement aux états fournies par le modèle de mélange de gaussiennes, nous en présentons quelques méthodes. Une première règle a été proposée par Smets dans

[Smets 93], dérivée de l'algorithme du GBT.

Le GBT (**Théorème de Bayes Généralisé**) est la généralisation du théorème de Bayes en probabilité aux fonctions de croyance [Smets 93]. Par rapport à sa version probabiliste, le GBT est capable de manipuler des fonctions de croyance permettant ainsi de modéliser la méconnaissance sur *l'a priori* nécessaire au calcul de la distribution a posteriori.

Les expressions analytiques des distributions générées par le GBT ont été dérivées par Smets [Smets 93]. Parmi les plus utilisées, trois sont basées sur les plausibilités $pl^\Omega[\theta_i]$ définies conditionnellement aux singletons $\theta_i \in \Theta$. Les expressions analytiques sont basées sur l'hypothèse que *l'a priori* sur Θ est vide. Le GBT génère alors les plausibilités, communalités et masses a posteriori suivantes :

$$pl^\Theta[\omega](\theta_i) = 1 - \prod_{\theta_i \in \Theta} (1 - pl^\Omega[\theta_i](\omega)) \quad (4.13)$$

$$q^\Theta[\omega](\theta_i) = \prod_{\theta_i \in \Theta} pl^\Omega[\theta_i](\omega) \quad (4.14)$$

$$m^\Theta[\omega](\theta_i) = \prod_{\theta_i \in \Theta} pl^\Omega[\theta_i](\omega) \cdot \prod_{\theta_i \in \bar{\Theta}} (1 - pl^\Omega[\theta_i](\omega)) \quad (4.15)$$

Dans la littérature, il est convenu que $pl^\Omega[\theta_i]$ est assimilée à une vraisemblance $l(\theta_i|\Omega)$ [Smets 93]. On en déduit donc une construction possible des fonctions de croyance à partir des vraisemblances.

Une autre construction des fonctions de croyance à partir de vraisemblances a été proposée dans [Appriou 91].

Soient $f(X|\omega_k)$ les densités de probabilité conditionnelles des classes que nous supposons connues. En ayant observé X , une fonction de vraisemblance est une fonction de Ω dans $[0, +\infty]$ définie par $L(\omega_k|X) = f(X|\omega_k), \forall k \in \{1 \dots K\}$. En utilisant cette fonction de vraisemblance construite à partir des données d'apprentissage, il est possible de définir une fonction de masse m_k associée à la classe ω_k . Les éléments focaux de cette fonction sont le complémentaire $\bar{\omega}_k$ du singleton $\{\omega_k\}$ et l'ensemble Ω lui-même. Cette fonction de masse est définie de la manière suivante :

$$m_k(\{\omega_k\}) = 0 \quad (4.16)$$

$$m_k(\bar{\omega}_k) = \alpha_k (1 - R.L(\omega_k|X)) \quad (4.17)$$

$$m_k(\Omega) = 1 - \alpha_k (1 - R.L(\omega_k|X)) \quad (4.18)$$

Dans ces équations, α_k est un coefficient d'affaiblissement associé à la classe ω_k tandis que R est un coefficient de normalisation positif inférieur ou égal à $[\sup_x \max_k (L(\omega_k|X))]^{-1}$. A partir de ces K fonctions de croyance et en utilisant la règle de combinaison, une fonction de masse unique m peut être obtenue. Elle quantifie le degré d'appartenance du vecteur X à chacune des classes dans Ω .

4.3.3 Le déconditionnement pour la fusion de FDM conditionnelles

L'une des applications du déconditionnement, est de fusionner des FDM conditionnelles sur un espace produit. On suppose disposer d'un jeu de FDM $m^\Omega[\theta_i], \forall \theta_i \in \Theta$ définies sur un cadre de discernement $\Omega = \omega_1, \omega_2 \dots$ conditionnellement à chacun des singletons θ_i d'un deuxième cadre de discernement $\Theta = \theta_1, \theta_2 \dots$.

Les FDM conditionnelles peuvent être combinées sur l'espace produit $\Omega \times \Theta$ obtenue par la combinaison conjonctive des FDM déconditionnées :

$$m^{\Omega \times \Theta} = \bigodot_i m^\Omega[\theta_i]^{\uparrow \Omega \times \Theta}$$

Chaque sous-ensemble S de l'espace produit $\Omega \times \Theta$ se voit allouer la masse :

$$m^{\Omega \times \Theta}(S) = \prod_{\theta_i \in \Theta} m^\Omega[\theta_i](v), \forall S \subseteq \Omega \times \Theta$$

où $v = ((\theta_i \times \Omega) \cap S) \downarrow \Omega$. Cette expression analytique des éléments de la FDM résultante a été dérivée par Smets [Smets 93]. De la même manière, l'expression analytique de la communalité des sous-ensemble $S \subseteq \Omega \times \Theta$ est donnée par :

$$q^{\Omega \times \Theta}(S) = \prod_{\theta_i \in \Theta} q^\Omega[\theta_i](v), \forall S \subseteq \Omega \times \Theta$$

Après avoir présenté le formalisme et quelques outils des modèles de croyances transférables, nous décrivons dans la section suivante comment nous les avons appliqué à la combinaison d'information dans un modèle de Markov multi-flux.

4.4 Approche proposée

Nous avons présenté dans le chapitre 1 les modèles de Markov multi-flux. Ces modèles permettent de combiner d'une façon asynchrone différentes sources d'informations, en construisant un MMC composite ou produit à partir des MMC mono-flux. Les densités d'observations (vraisemblances) conditionnelles aux états du MMC composite sont déterminées en combinant les vraisemblances conditionnelles aux états correspondants dans les MMC mono-flux.

Tout au long des expérimentations réalisées dans le chapitre 3, cette combinaison est assurée par une simple règle de combinaison linéaire des vraisemblances.

Dans cette partie, nous exploitons les possibilités offertes par les fonctions de croyance à travers leurs différentes règles de combinaison conçues pour la fusion d'informations. Nous expliquons dans la partie suivante comment exploiter les fonctions de croyance pour déterminer les vraisemblances dans le MMC composite en combinant les vraisemblances des MMC mono-flux. Nous évaluons l'approche proposée dans le cadre de la reconnaissance de mots manuscrits.

4.4.1 Les différentes étapes de l'approche proposée

L'approche de combinaison proposée prend comme entrée les vraisemblances conditionnelles aux états de chaque MMC mono-flux, soient $P(O^1|s_i^1), 1 \leq i \leq N_1$ pour le MMC du premier flux, et $P(O^2|s_j^2), 1 \leq j \leq N_2$ pour le MMC du deuxième flux. A partir de ces vraisemblances, nous calculons en se basant sur les outils des MCT, la vraisemblance conditionnelle aux états produits du MMC composite $P(O^{1,2}|s_k^{1,2}), 1 \leq k \leq N_1 \times N_2$. Ces dernières vraisemblances sont utilisées par la suite par l'algorithme de Viterbi lors de la phase de décodage multi-flux.

L'approche de combinaison proposée comprend les différentes étapes suivantes (voir figure 4.4) :

1. Transformer les vraisemblances de chaque source d'information en des fonctions de masses.

2. Construire un nouveau cadre de discernement conjoint aux deux sources d'information défini par l'espace produit.
3. Étendre les fonctions de masse des deux sources d'informations sur cet espace produit.
4. Combiner les deux fonctions de masses déjà définis sur le même cadre de discernement produit.
5. Transformer les fonctions de masses en des probabilités des états produit conditionnellement aux observations.

Ces différentes étapes sont réitérées pour chaque modèle de caractère dans un mot du lexique et pour chaque trame d'observation. Nous les détaillons dans les sections suivantes.

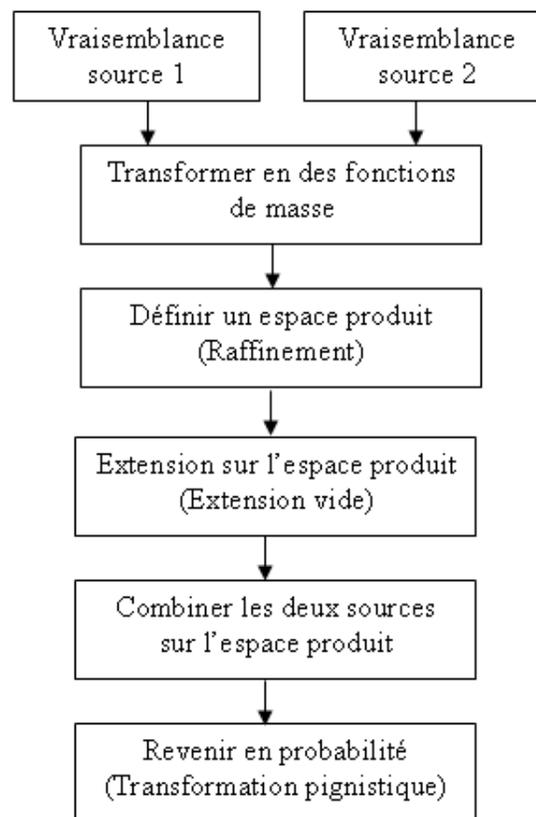


Figure 4.4. Différentes étapes de l'approche de combinaison proposée

4.4.2 Définition des cadres de discernements

Un des problèmes les plus importants dans la mise en place d'une approche crédibiliste est la définition du cadre de discernement dans lequel l'algorithme va évoluer. Le nombre d'hypothèses doit être en général assez limité, car la taille de 2^Ω croît exponentiellement avec celle de Ω .

Nous disposons en entrée de notre système les vraisemblances $P(O_t^k | s_j^k)$ où $t \in \{1 \dots T\}$ désigne le nombre de trames, $j \in \{1 \dots N\}$ désigne le nombre d'états dans le MMC du flux $k \in \{1 \dots K\}$, K étant le nombre de flux. On suppose que les vraisemblances sont calculées par des mélanges Gaussiens comme décrit dans le chapitre 3.

Selon les choix faits au chapitre 3, la combinaison des vraisemblances se fait au niveau caractère. On définit donc un cadre de discernement formé par les quatre états du modèle de caractère pour chaque flux.

Soit $\Omega_t^k = \{s_1^k, s_2^k, s_3^k, s_4^k\}$ le cadre de discernement pour un modèle de caractère du flux k à un instant t .

Pour simplifier la présentation, nous nous plaçons dans le cas de deux flux, le principe peut être généralisé par la suite à N flux.

Pour un caractère donné, et un instant t fixé, on définit donc deux cadres de discernement différents correspondant aux deux flux tels que :

$$\Omega_t^1 = \{s_1^1, s_2^1, s_3^1, s_4^1\} \text{ et } \Omega_t^2 = \{s_1^2, s_2^2, s_3^2, s_4^2\}$$

Après avoir défini le cadre de discernement, l'étape suivante consiste à construire les fonctions de masses.

4.4.3 Construction des fonctions de masse

Les fonctions de masses sont construites à partir des vraisemblances $P(O_t^k | s_j^k)$ données par un modèle de mélanges de gaussiennes. Pour cela, on applique la transformation décrite dans le paragraphe 4.3.2, dérivée du théorème de GBT et on obtient :

Avec les communalités :

$$q^{\Omega_t^k} [O_t^k] (S_k) = \prod_{s_j^k \in S_k} P(O_t^k | s_j^k)$$

Avec les masses, la formule devient :

$$m^{\Omega_t^k} [O_t^k] (S_k) = \prod_{s_j^k \in S_k} P(O_t^k | s_j^k). \prod_{s_j^k \in S_k} (1 - P(O_t^k | s_j^k))$$

Pour simplifier l'écriture, on omettra le conditionnement $[O_t^k]$ et on notera $m^{\Omega_t^k} (S_k)$ (resp. $q^{\Omega_t^k} (S_k)$) la masse (resp. la communalité) des états conditionnelles aux observations.

Ainsi, nous attribuons une masse pour chaque sous ensemble $S_k \in 2^{\Omega_t^k}$ et $S_2 \in 2^{\Omega_t^2}$. Le nombre de sous-ensembles est de $2^{|\Omega_t^k|}$, soit dans notre cas $2^4 = 16$ sous-ensembles différents.

4.4.4 Combinaison des fonctions de masse

Les fonctions de masses déjà construites pour les deux sources d'informations doivent être combinées, mais ces fonctions ne sont pas définies sur le même cadre de discernement. Une combinaison directe n'est donc pas possible.

La solution est de construire un cadre de discernement commun pour les deux sources qui est l'espace produit, et d'étendre les deux fonctions de masse sur cet espace produit comme décrit dans la section 4.3.1.

Nous définissons l'espace produit par :

$$\Omega_t^1 \times \Omega_t^2 = \{(s_i^1, s_j^2), \forall i \in \{1 \dots |\Omega_t^1|\}, \forall j \in \{1 \dots |\Omega_t^2|\}\}$$

Les fonctions de masses sont étendues avec l'opérateur d'extension vide sur l'espace produit $\Omega_t^1 \times \Omega_t^2$ de la façon suivante :

$$m^{\Omega_t^1 \uparrow \Omega_t^1 \times \Omega_t^2} (C) = \begin{cases} m^{\Omega_t^1} (B) & \text{si } C = B \times \Omega_t^2 \\ 0 & \text{sinon} \end{cases}$$

de même :

$$m^{\Omega_t^2 \uparrow \Omega_t^1 \times \Omega_t^2} (C) = \begin{cases} m^{\Omega_t^2} (B) & \text{si } C = \Omega_t^1 \times B \\ 0 & \text{sinon} \end{cases}$$

A ce niveau, les fonctions de masses des deux sources d'informations peuvent être combinées en utilisant la règle conjonctive (Eq 4.8) ce qui nous permet de déterminer des fonctions de masses conditionnelles aux sous-ensembles d'états de l'espace produit.

Or, dans l'approche proposée, la procédure de décodage se fait dans l'espace probabiliste à l'aide de l'algorithme de Viterbi. Parmi les entrées de l'algorithme, il faut définir les probabilités d'observations conditionnelles aux états produit. Nous utilisons donc la transformation pignistique (Eq 4.12) pour transformer les masses des sous ensembles d'états conditionnelles aux observations en des probabilités d'observations conditionnellement aux états produit.

4.4.5 Complexité de l'approche proposée

Un point important que nous soulignons sur l'utilisation des fonctions de croyance en général, et sur notre approche en particulier, est la complexité qui croit considérablement comparée à l'approche probabiliste.

Nous présentons dans l'algorithme 6 les étapes de décodage multi-flux, faisant appel à l'approche de combinaison à base de MCT pour la reconnaissance de mots manuscrits.

A travers cet algorithme, nous remarquons déjà que pour une image de mot de test donnée, les étapes de combinaison à base de MCT sont réitérées, pour chaque mot du lexique, pour chaque caractère dans le mot, et pour chaque trame d'observation. La complexité de l'algorithme provient surtout du fait de travailler sur des cadres de discernement produit ou nous nous trouvons avec un grand nombre de sous-ensembles. En effet, dans l'approche de combinaison proposée, nous partons de deux cadres de discernement de taille 4 chacun, représentant les états du MMC mono-flux d'une source d'information donnée. Nous définissons alors des fonctions de masses pour les $2^4 = 16$ sous ensembles possibles de chaque source d'information.

En passant à l'espace produit, nous construisons un nouveau cadre de discernement de taille $4 \times 4 = 16$ et nous étendons les fonctions de masses initiales sur ce nouvel espace conjoint, soit $2^{16} = 65536$ sous ensembles possibles.

Nous signalons que pour implémenter cette approche, nous avons été amenés

Algorithme 6: Algorithme de décodage avec une combinaison à base de MCT

pour chaque image de mot à reconnaître **faire**
 pour chaque mot du lexique **faire**
 pour chaque modèle de caractère dans le mot du lexique **faire**
 pour chaque trame d'observation **faire**
 - Transformer les vraisemblances en des fonctions de masses.
 - Construire l'espace produit.
 - Étendre les fonctions de masse sur cet espace produit.
 - Combiner les deux fonctions de masses .
 - Transformation pignistique des fonctions de masses.
 fin pour
 Stocker les probabilités du caractère courant.
 fin pour
 Lancer Viterbi sur le mot courant du lexique
 Mémoriser la vraisemblance obtenue.
 fin pour
 Choisir le mot du lexique avec la vraisemblance maximale.
fin pour

à transformer nos algorithmes de façon à ne gérer que les masses non-nulles. Par exemple, la fonction implémentant la règle de combinaison conjonctive présente une complexité de l'ordre de $O(N^3)$, où $N = 65536$ est le nombre de sous ensembles, il nous est donc impossible de travailler avec telle grandeur. Si nous ne considérons que les valeurs non nulles, le nombre de sous ensembles dont la masse est non nulle devient de l'ordre de quelques centaines. Ceci réduit la complexité de l'algorithme à environs 10^6 itérations, ce qui nous a permis de mettre en oeuvre l'approche. Cependant avec un tel nombre d'itérations, l'approche reste malgré tout complexe comparée à l'approche probabiliste.

Remarque

Tout au long de cette description, les différentes étapes de notre approche ont été présentées dans le cadre de deux sources d'informations. Théoriquement, rien n'empêche d'étendre cette approche à plus de deux sources d'informations. En pratique, étant donnés les problèmes de complexité évoqués, il est difficile à notre avis de passer au delà de 2 sources d'informations.

4.4.6 Évaluation de l'approche proposée sur la reconnaissance de mots

L'algorithme 6 intégrant l'approche de combinaison à base de MCT est évalué dans le cadre de la reconnaissance de mots manuscrits Latin. Vu la complexité de l'approche, nous avons choisi de réaliser des tests sur un lexique réduit de 30 mots, celui du sous-ensemble IRONOFF-chèque de la base IRONOFF. Les expérimentations sont effectuées dans le cadre d'une approche 2-flux. La tableau 4.1 présente les résultats de reconnaissance obtenus en combinant chaque fois 2 flux parmi les 4 sources d'informations présentées dans le chapitre 3 : 1) contour supérieur, 2) contour inférieur, 3) densité 1, 4) densité 2. Les résultats de l'approche proposée sont comparés à ceux déjà obtenus avec une combinaison probabiliste linéaire.

Modèles	Probabiliste		MCT	
	Rang1	Rang5	Rang1	Rang5
<i>2-flux 1)2)</i>	86	95.4	47.8	83.3
<i>2-flux 1)3)</i>	92.8	99.2	60.2	86.2
<i>2-flux 1)4)</i>	91.8	99	60.6	86.4
<i>2-flux 2)3)</i>	94	99.6	62.4	87.8
<i>2-flux 2)4)</i>	92.8	99.2	61	87.1
<i>2-flux 3)4)</i>	91	99	49.4	84.4

Table 4.1. Performances sur la base IRONOFF-Chèque

Les résultats obtenus montrent que l'approche de combinaison proposée à base de fonctions de croyance n'améliore pas les taux de reconnaissance déjà obtenus par l'approche de combinaison probabiliste.

Notons que d'autres expérimentations ont été réalisées pour tester la même approche de combinaison mais en utilisant l'autre règle de construction des fonctions de masse à partir des vraisemblance décrite dans la section 4.3.2. Les résultats obtenus sont un peu moins bons que ceux obtenus par la première règle de construction.

Nous expliquons cette dégradation des résultats par le fait que les vraisemblances utilisées dans l'approche probabiliste varient entre 0 et l'infini, tandis que dans l'approche crédibiliste, on force ces vraisemblances à être entre 0 et 1 en utilisant la transformation pignistique. Du coût, les vraisemblances perdent leurs poids dans l'approche crédibiliste ce qui influe par la suite sur les résultats du décodage.

De plus, dans le formalisme des MCT, le niveau pignistique est généralement

utilisé pour prendre des décisions en prenant éventuellement en compte les représentations au niveau crédal. Cependant dans notre approche, nous ne prenons pas de décisions suite à la transformation pignistique, mais nous procédons plutôt à un décodage probabiliste avant de décider. Ces transformation vraisemblance-masse et masse-probabilité sont à notre avis un inconvénient majeur de l'approche proposée. Pour cela nous pensons qu'une procédure de décodage crédibiliste pourrait être une solution intéressante.

Récemment dans [Ramasso 07b, Ramasso 07a], il a été proposé de formaliser les algorithmes des MMC dans le cadre du MCT afin de combiner la puissance des algorithmes des MMC, à la flexibilité, la généralité et les outils tout aussi puissants du MCT. Ces MMC dits crédibilistes, ont été appliqués par [Ramasso 07b, Ramasso 07a] dans le cadre de la reconnaissance d'actions et d'activités humaines dans les vidéos d'athlétisme.

Nous décrivons dans la partie suivante le principe de ces modèles de Markov crédibilistes en se référant à la présentation faite dans [Ramasso 07b, Ramasso 07a].

4.5 MMC crédibiliste

La démarche proposée dans [Ramasso 07b, Ramasso 07a] consiste à transformer les mécanismes de base des MMC probabilistes afin de pouvoir gérer les fonctions de croyance. Une description des algorithmes de Forward et Backward crédibilistes a été présentée en se basant sur la théorie de l'évidence [Shafer 76]. Ces algorithmes n'ont pas été appliqués pour l'estimation des paramètres des MMC crédibilistes et quelques pistes ont été proposées dans [Ramasso 07b] pour développer un algorithme d'apprentissage des MMC crédibilistes.

Dans les expérimentations présentées, la phase d'apprentissage des paramètres des MMC est assurée par les algorithmes probabilistes classiques. Une phase de transformation vient par la suite pour transformer les distributions probabilistes en des fonctions de croyance utilisées par la suite dans l'étape de décodage assurée par un algorithme de Viterbi crédibiliste.

Nous nous intéressons dans cette partie à décrire les différents étapes de l'algorithme de Viterbi crédibiliste proposé dans [Ramasso 07b, Ramasso 07a].

4.5.1 L'algorithme de Viterbi crédibiliste

L'algorithme de Viterbi crédibiliste détermine la séquence d'états singletons optimale (au sens d'un critère prédéfini) $s_1, s_2 \dots s_T \in \Omega_t$ pouvant avoir générée la séquence des observations $O_{1:T}$ par des mécanismes de programmation dynamique comme dans le cas probabiliste.

Pour chaque état $s_j^t \in \Omega_t$, on doit déterminer le meilleur prédécesseur $s_i^{t-1} \in \Omega_{t-1}$. La recherche du prédécesseur fait intervenir à la fois les états à t et $t - 1$, c'est pourquoi on estime préalablement la FDM concernant les états sur l'espace produit $\Omega_{t-1} \times \Omega_t$. Cette FDM est obtenue en combinant les informations disponibles sur ces deux cadres de discernement à savoir :

- Les fonctions de croyance conditionnelles $f_a^{\Omega_t} [S_i^{t-1}]$ décrivant les transitions.
- La fonction de croyances $f_b^{\Omega_t}$ relative aux observations.
- La fonction de croyances $f_\delta^{\Omega_{t-1}}$ sur les états à l'instant précédent.

où $f \in pl, bel, m, q, b$.

La première étape est la transformation des probabilités de transitions et les densités d'observations en des distributions de masses.

Les fonction de croyance relatives aux observations sont déterminées en appliquant la transformation décrite au paragraphe 4.3.2, dérivée du théorème de GBT.

Les matrices de transition des MMC crédibilistes sont construites en partant des matrices probabilistes et en appliquant la méthode de transformation en fonctions de croyance basée sur le Principe du Minimum d'Information (PMI).

Le PMI pour transformer une distribution de probabilités en FDM

Disposons d'une distribution de probabilités P sur les singletons $\omega_1, \omega_2, \dots, \omega_K$ de Ω (avec $K = |\Omega|$) dont les éléments de la distribution sont ordonnés :

$$P(\omega_1) \geq P(\omega_2) \dots \geq P(\omega_k) \dots \geq P(\omega_K)$$

Le PMI permet de construire la FDM la moins engagée \hat{m}_Ω associée à P composée d'éléments focaux emboîtés $A = \omega_1, \omega_2, \dots, \omega_k$ tels que :

$$\hat{m}_\Omega(A) = |A| P(\omega_k) - P(\omega_{k+1})$$

Il a été montré par Smets que cette FDM est consonante.

Synthèse de l'information courante

L'étape suivante consiste à exprimer $f_a^{\Omega_t} [S_i^{t-1}]$, $f_b^{\Omega_t}$, $f_\delta^{\Omega_{t-1}}$ sur l'espace commun $\Omega_{t-1} \times \Omega_t$.

Pour cela on applique l'opérateur d'extension vide pour déterminer $m_b^{\Omega_t \uparrow \Omega_{t-1} \times \Omega_t}$, $q_\delta^{\Omega_{t-1} \uparrow \Omega_{t-1} \times \Omega_t}$.

Les FDM définies sur Ω_t conditionnellement aux singletons $s_i^{t-1} \in \Omega_{t-1}$ sont déconditionnées sur l'espace produit $\Omega_{t-1} \times \Omega_t$ en utilisant l'opérateur de déconditionnement basé sur les communalités décrit dans le paragraphe 4.3.3. Les fonctions de croyance peuvent maintenant être combinées par un opérateur conjonctif tel que la CRC (supposant les FDM distinctes) :

$$q_{a\delta b}^{\Omega_{t-1} \times \Omega_t}(S) = q_\delta^{\Omega_{t-1} \uparrow \Omega_{t-1} \times \Omega_t}(S) \cdot q_a^{\Omega_{t-1} \times \Omega_t}(S) \cdot q_b^{\Omega_t \uparrow \Omega_{t-1} \times \Omega_t}(S)$$

Recherche du prédécesseur

A présent que la connaissance sur $t-1$ et t est connue, on peut déterminer le prédécesseur $\Psi'(s_j^t) \in \Omega_{t-1}$ d'un état s_j^t . Le meilleur prédécesseur $\Psi'(s_j^t)$ d'un état s_j^t est celui qui maximise soit la probabilité pignistique soit la plausibilité (i.e. un critère de décision) de l'état s_j^t après les étapes suivantes :

- Conditionnement de la FDM $q_{a\delta b}^{\Omega_{t-1} \times \Omega_t}$ par le prédécesseur candidat s_i^{t-1} .
- Projection sur Ω_t de la FDM $q_{a\delta b}^{\Omega_{t-1} \times \Omega_t}$.
- Calcul du critère de décision (probabilité pignistique et plausibilité) de chaque état sur Ω_t .

On obtient donc :

$$P_t^1 [s_i^{t-1}] (s_j^t) = Bet P^{\Omega_t} \left\{ m_{a\delta b}^{\Omega_{t-1} \times \Omega_t} [s_i^{t-1} \times \Omega_t]^{\downarrow \Omega_t} \right\} (s_j^t)$$

ou encore, avec les plausibilités :

$$P_t^2 [s_i^{t-1}] (s_j^t) = PIP^{\Omega_t} \left\{ m_{a\delta b}^{\Omega_{t-1} \times \Omega_t} [s_i^{t-1} \times \Omega_t]^{\downarrow \Omega_t} \right\} (s_j^t)$$

Le conditionnement par s_i^{t-1} engendre généralement du conflit. Ce conflit représente la pertinence de l'hypothèse (le prédécesseur de s_j^t est s_i^{t-1}) que l'on mémorise dans la variable $\Gamma_t(s_i^{t-1})$ à valeur dans $[0, 1]$:

$$\Gamma_t(s_i^{t-1}) = 1 - m_{a\delta b}^{\Omega_{t-1} \times \Omega_t} [s_i^{t-1} \times \Omega_t] (\emptyset)$$

Plus le conflit est faible, plus l'hypothèse "le prédécesseur de s_j^t est s_i^{t-1} " est pertinente et on exploitera cette information pour la prise de décision concernant le meilleur prédécesseur.

Pour chaque état s_j^t , on dispose donc d'un jeu de probabilités $P_t^k [s_i^{t-1}] (s_j^t), \forall s_i^{t-1} \in \Omega_{t-1}, k \in 1, 2$ ainsi qu'un ensemble de pertinences $\Gamma_t(s_i^{t-1}) \forall s_i^{t-1} \in \Omega_{t-1}$. Afin de déterminer, parmi tous les états $s_i^{t-1} \in \Omega_{t-1}$ le meilleur prédécesseur de l'état s_j^t , il suffit de chercher l'état s_i^{t-1} maximisant la valeur $P_t^k [s_i^{t-1}] (s_j^t)$ pondérée par la pertinence associée à s_i^{t-1} :

$$\Psi'(s_j^t) = \operatorname{argmax}_{s_i^{t-1} \in \Omega_{t-1}} \left[\Gamma_t(s_i^{t-1}) \cdot P_t^k [s_i^{t-1}] (s_j^t) \right]$$

$$\delta'(s_j^t) = \max_{s_i^{t-1} \in \Omega_{t-1}} \left[\Gamma_t(s_i^{t-1}) \cdot P_t^k [s_i^{t-1}] (s_j^t) \right]$$

où $\delta'(s_j^t), \Psi'(s_j^t)$ mémorisent la valeur maximale de la transition du prédécesseur de s_j^t vers s_j^t ainsi que l'indice du prédécesseur.

Propagation des résultats

Connaissant le prédécesseur de chaque état s_j^t à l'instant t . Il faut maintenant construire la FDM $m_\delta^{\Omega_t}$ qui sera utilisée à l'itération suivante ($t + 1$) et qui a pour rôle de propager les valeurs maximales $\delta'(s_j^t)$ déterminées à t . Dans [Ramasso 07b, Ramasso 07a] les auteurs proposent d'estimer la FDM $m_\delta^{\Omega_t}$ en projetant la FDM $m_\delta^{\Omega_{t-1}}$ sur Ω_t par l'algorithme Forward mais en conditionnant la projection à l'ensemble des états prédécesseurs définis sur

$t - 1$ et noté A_{t-1} tel que :

$$A_{t-1} = \bigcup_{s_j^t \in \Omega_t} \Psi'(s_j^t)$$

Ainsi la FDM résultante obtenue est :

$$m_\delta^{\Omega_t} = m_b^{\Omega_t} \odot \left(\left(m_\delta^{\Omega_{t-1} \uparrow \Omega_{t-1} \times \Omega_t} \odot m_a^{\Omega_{t-1} \times \Omega_t} \right) [\Omega_t \times A_{t-1}] \right) \downarrow^{\Omega_t}$$

Séquence d'états optimale

La séquence d'états optimale, notée $s_*^1, s_*^2, \dots, s_*^T$ est déterminée par *backtracking* en partant de l'état s_*^T maximisant la probabilité pignistique tel que :

$$s_*^T = \operatorname{argmax}_{s_j^T \in \Omega_T} \operatorname{Bet} P^{\Omega_T} \left\{ m_\delta^{\Omega_T} \right\} (s_j^T)$$

Ce processus est similaire à celui utilisé dans le cas des HMM probabilistes.

Classification de séquence d'observations

Disposant de N MMC crédibilistes $\lambda_1, \lambda_2, \dots, \lambda_N$ représentant chacun un type de séquence particulier, on cherche à connaître le modèle λ_* correspondant le mieux aux observations.

Dans [Ramasso 07b, Ramasso 07a], il a été proposé un nouveau critère pour le choix du meilleur modèle λ_* basé sur une mesure globale de conflit $l_c(\lambda)$ entre un modèle λ et des observations $O_{1:T}$ définie par :

$$l_c(\lambda) = \frac{1}{T} \sum_{t=1}^T \log \left(m_\delta^{\Omega_t} [\lambda] (\emptyset) \right)$$

Le modèle λ_* correspondant le mieux aux observations mesurées est alors celui qui minimise le conflit sur la séquence des observations :

$$\lambda_* = \operatorname{argmin}_n l_c(\lambda_n)$$

4.5.2 Complexité de l'approche

Il était clair dans la présentation précédente sur les MMC crédibilistes que l'on est confronté à une complexité qui dépend principalement du nombre d'états dans le MMC et qui augmente en passant à un cadre de

discernement produit. Dans le cadre de la reconnaissance de mots manuscrits isolés, ce problème est plus flagrant étant donné qu'un mot est modélisé par un nombre d'états proportionnel au nombre de caractères qui le composent.

Prenons un exemple concret afin d'évaluer ce problème de complexité. Considérons un mot formé de 5 caractères, sachant que chaque caractère est modélisé par un MMC à 4 états, on construit par concaténation des modèles de caractère un MMC à 20 états correspondant à ce mot.

Si on considère comme cadre de discernement les états du MMC du mot, nous disposons de $2^{20} = 1048576$ sous-ensemble différents. De plus, nous avons vu que dans le cadre du Viterbi crédibiliste, pour combiner des fonctions de masses définies à des instants différents t et $t - 1$, il est nécessaire de construire un cadre de discernement commun défini par l'espace produit. Dans ce cas, le cardinal de l'espace produit est de $20 \times 20 = 400$, et le nombre de sous-ensembles s'élève à 2^{400} ce qui est ingérable par la machine.

A noter que nous avons pu tester le Viterbi crédibiliste en restreignant le nombre d'états par caractère à 1, cependant les résultats de reconnaissance obtenus ont été fortement influencés par cette restriction du nombre d'états.

4.6 Conclusion

Nous avons exploré tout au long de ce chapitre la théorie des fonctions de croyance transférables en décrivant son formalisme et les différents outils qu'elle présente. Nous avons proposé une méthode de combinaison d'information à base de MCT dans le cadre de l'approche multi-flux. L'approche proposée a été testée dans le cadre de la reconnaissance de mots manuscrits isolés en dégradant les résultats déjà obtenus dans l'approche probabiliste. Nous avons décrit dans la suite le principe des MMC crédibilistes ainsi que l'algorithme de Viterbi à base de MCT. Nous avons mis l'accent sur le problème de complexité des fonctions de croyance qui a présenté un handicap pour la mise en oeuvre des MMC crédibilistes dans le cadre de la reconnaissance de mots manuscrits.

Conclusion générale

Le travail présenté dans cette thèse se situe dans le cadre de la reconnaissance hors-ligne de l'écriture manuscrite multi-scripts, problème jugé complexe du fait de la variabilité de l'écriture manuscrite d'une part, et des contraintes supplémentaires provenant de l'aspect multi-script d'autre part. Une étude bibliographique sur les caractéristiques de l'écriture manuscrite en général, et de la langue Arabe et Latine en particulier, ainsi que sur les différentes approches de reconnaissance de mots isolés présentées dans la littérature, a permis d'élaborer et de valider un système de reconnaissance de l'écriture multi-scripts.

Le système de reconnaissance proposé opère indépendamment de la nature du script, pour cela, le système procède sans segmentation explicite de l'écriture afin d'éviter la dépendance de l'approche proposée à un script donné. De même, les phases de pré-traitements et d'extraction de caractéristiques sont indépendantes de la nature du script. De plus, nous avons opté pour une modélisation markovienne qui convient bien à une approche multi-script étant donné que ces modèles sont adaptés à la modélisation des différentes variations morphologiques des caractères et également pour la résolution des problèmes épineux de segmentation.

L'évaluation du système de reconnaissance proposée est effectuée sur différentes bases de données publiques. Les résultats obtenus en utilisant un seul flux de données nous ont permis de valider le système initial sur des bases de mots Arabes et Latins.

Afin d'améliorer les résultats obtenus par ce premier système mono-flux, nous avons proposé un deuxième système de reconnaissance permettant de combiner plusieurs sources d'informations à travers une approche markovienne multi-flux. L'évaluation de ce deuxième système a montré que l'approche multi-flux améliore considérablement les résultats obtenus par

le premier système mono-flux pour les deux scripts Arabe et Latin. Ces résultats dépassent également les performances obtenues par les approches classiques de combinaison par fusion des scores et par fusion des représentations et sont comparables aux meilleurs résultats de reconnaissance rapportés dans la littérature.

Ce dernier système a été adapté pour être testé sur une base de données multi-lingue, composée par un mélange de mots Arabes et Latins. L'évaluation du système de reconnaissance proposé sur cette base de mots multi-lingue montre que le système parvient à bien identifier et reconnaître des mots multi-scripts avec un taux de confusion très faible.

Afin d'exploiter d'autres règles de combinaison dans le cadre de l'approche multi-flux, nous avons proposé une méthode de combinaison à base de modèles de croyances transférables. Cela nous a permis d'explorer un nouveau domaine de recherche et de présenter le formalisme de base de ses outils connus par leur capacité d'appréhender à la fois imprécision et incertitude et proposant des mécanismes intéressants pour la fusion d'information. Cependant, l'approche de combinaison proposée à base de MCT ne nous a pas permis d'améliorer les résultats de reconnaissance déjà obtenus. De plus, la complexité énorme de ces modèles appliqués à la reconnaissance de l'écriture a souvent limité nos travaux.

Malgré la réalisation d'un système de reconnaissance complet efficace, plusieurs améliorations et perspectives sont envisageables. En effet, Les méthodes proposées dans cette thèse sont limitées à la reconnaissance de mots manuscrits supposés préalablement isolés. Or, dans la pratique les mots sont écrits dans le contexte d'un texte, ainsi la première perspective envisagée est l'extension des méthodes proposées à la reconnaissance des textes manuscrits sans contraintes. Cette extension requiert le développement d'un module fiable de détection et d'extraction de mots d'un texte manuscrit. Nous envisageons aussi de comparer la performance de notre système avec les derniers systèmes proposés dans la littérature. Ainsi, une de nos perspectives futures est de participer à la troisième compétition sur la reconnaissance des mots manuscrits arabes sur la base IFN/ENIT, et la deuxième compétition sur la reconnaissance des mots Latins sur la base RIMES, qui auront lieu à l'occasion de la conférence ICDAR 2009. D'autre part, dans le cadre l'ap-

proche multi-flux, nous avons testé deux règles de combinaison différentes, une première règle de combinaison linéaire dans le cadre probabiliste et une deuxième règle à base de modèles de croyance transférable. Nous envisageons dans nos prochains travaux de tester d'autres règles de combinaison, spécialement les règles de combinaison non-linéaire à base de MLP. Les caractéristiques utilisées dans le cadre de ce travail sont des primitives bas niveaux basées sur la densité et les directions des pixels. Nous proposons de tester d'autres jeux de caractéristiques, spécialement des caractéristiques structurales telles que les concavités, les convexités, les occlusions, les ascendants, les descendants, les composantes connexes. Ces caractéristique peuvent améliorer les résultats de reconnaissance grâce à leur caractère discriminant et leur complémentarité avec les sources d'informations existantes. Il peut être intéressant de combiner de tels caractéristiques dans le cadre de l'approche multi-flux. Nous avons vu que la complexité de l'algorithme de décodage multi-flux présente un sérieux handicap pour notre approche. Cet handicap se trouve accentué au delà de deux flux. Afin de réduire cette complexité, l'algorithme de "two-level" peut être une alternative intéressante pour accélérer la procédure de décodage. Enfin, une extension intéressante de ce travail, consisterait à combiner une approche de reconnaissance avec segmentation explicite avec une approche sans segmentation dans le cadre du formalisme multi-flux.

Travaux de l'auteur

- [1] Y. Kessentini, T. Paquet, A. Benhamadou. A Multi-stream Approach to Off-Line Handwritten Word Recognition. *9th International Conference on Document Analysis and Recognition, ICDAR'2007*, Curitiba, Brazil, vol. 1, pp. 317-321, 2007.
- [2] Y. Kessentini, T. Paquet, A. Benhamadou, A Multi-Stream HMM-based Approach for Off-line Multi-Script Handwritten Word Recognition. *11th International Conference on Frontiers in Handwriting Recognition, ICFHR'08*, Montréal, Québec, vol 1, pp. 147-152, 2008.
- [3] Y. Kessentini, T. Paquet, A. Benhamadou. Multi-Script Handwriting Recognition with N-Streams Low Level Features. *International Conference on Pattern Recognition, ICPR'08*. Tampa, Floride, 2008. (Actes sur CDROM).
- [4] Y. Kessentini, T. Paquet, A. Benhamadou. Modèles multi-flux pour la reconnaissance de mots manuscrits. *21e colloque sur le traitement du signal et des images, GRETSI'07*, Troyes, France 2007. (Actes sur CDROM).
- [5] Y. Kessentini, T. Paquet, A. Benhamadou. Combinaison d'Information pour la Reconnaissance de l'Écriture Manuscrite Hors-Ligne. *16e congrès francophone de Reconnaissance des Formes et Intelligence Artificielle, RFIA'2008*, Amiens, France 2008. (Actes sur CDROM)
- [6] Y. Kessentini, T. Paquet, A. Benhamadou. Combinaison de caractéristiques bas niveau à travers des modèles N-flux dans le cadre de la reconnaissance de l'écriture manuscrite multi-scripts. *Colloque International Francophone sur l'Écrit et le Document, CIFED'08*. Rouen, France, pp. 109-114, 2008.
- [7] Y. Kessentini, T. Paquet, A. Benhamadou. Off-Line Handwritten Word Recognition Using Multi-Stream Hidden Markov Models. *En révisions mineures, Pattern Recognition Letters, PRL*, 2009.
- [8] Y. Kessentini, T. Paquet, A. Benhamadou. Reconnaissance de L'écriture Manuscrite Multi-Scripts par des Modèles de Markov Cachés Multi-Flux. soumis à la revue *Traitement de Signal*, sélectionné du colloque CIFED'08.
- [9] Y. Kessentini, T. Paquet, A. Benhamadou. A Multi-Lingual Recognition System for Arabic and Latin Handwriting. *10th International Conference on Document Analysis and Recognition, ICDAR'2009*, p. to appear, Barcelone, Espagne 2009.

Bibliographie

- [Agazi 93] O. Agazi & S. Kuo. *Hidden Markov model based optical character recognition in the presence of deterministic transformations*. Pattern Recognition, vol. 26, no. 12, pages 1813–1826, 1993.
- [Al-Hajj 05] R. Al-Hajj, L. Likforman Sulem & C. Mokbel. *Arabic handwriting recognition using baseline dependant features and hidden Markov modeling*. ICDAR, vol. 2, pages 893–897, 2005.
- [Al-Hajj 06] R. Al-Hajj, C. Mokbel & L. Likeforman-Seulem. *Reconnaissance de l'écriture arabe cursive : combinaison de classifieurs MMCs à fenêtres orientées*. Colloque International Francophone sur l'Écrit et le Doucement CIFED'06, pages 271–276, 2006.
- [Al-Hajj 07] R. Al-Hajj. *Reconnaissance hors ligne de textes manuscrits cursifs par l'utilisation de systèmes hybrides et de techniques d'apprentissage automatique*. Thèse de Doctorat, Ecole Nationale Supérieure de Télécommunications, Paris, 2007.
- [Al-Yousefi 92] H. Al-Yousefi & S.S. Udpa. *Recognition of Arabic Characters*. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 14, pages 853–857, 1992.
- [Alimi 97] M. A. Alimi. *An evolutionary neuro-fuzzy approach to recognize on-line Arabic handwriting*. The Proceedings of ICDAR'97, 4th International Conference on Document Analysis and Recognition, vol. 1, pages 382–386, 1997.

- [Alma'adeed 04] S. Alma'adeed, C. Higgins & D. Elliman. *Off-Line Recognition of Handwritten Arabic Words Using Multiple Hidden Markov Models*. Knowledge-Based Systems, vol. 17, pages 75–79, 2004.
- [Amara 99] N. E. Ben Amara. *Utilisation des modèles de markov cachés planaires en reconnaissance de l'écriture arabe imprimée*. Thèse de Doctorat, Ecole Nationale d'Ingénieur de Tunis, 1999.
- [Amara 00] N. Ben Amara, A. Belaïd & N. Ellouze. *Utilisation des modèles Markoviens en reconnaissance de l'écriture arabe état de l'art*. Colloque International Francophone sur l'Ecrit et le Document (CIFED'00), pages 181–191, 2000.
- [Amin 96] A. Amin, H. Al-Sadoun & S. Fischer. *Hand-Printed Arabic Character Recognition System Using an Artificial Network*. Pattern Recognition, vol. 29, pages 663–675, 1996.
- [Amin 98] A. Amin. *Off-Line Arabic Character-Recognition : The State of the Art*. Pattern Recognition, vol. 31, no. 5, pages 517–530, 1998.
- [Amin 03] A. Amin. *Recognition of Hand-Printed Characters Based on Structural Description and Inductive Logic Programming*. Pattern Recognition Letters, vol. 24, pages 3187–3196, 2003.
- [Appriou 91] A. Appriou. *Probabilités et incertitude en fusion de données multi-senseurs*. Revue Scientifique et Technique de la Défense, vol. 11, pages 27–40, 1991.
- [Artières 03] T. Artières, N. Gauthier, P. Gallinari & B. Dorizzi. *A Hidden Markov Models combination framework for handwriting recognition*. International Journal on Document Analysis and Recognition (IJ DAR), vol. 5, no. 4, pages 233–243, 2003.
- [Augustin 01] E. Augustin. *Reconnaissance de mots manuscrits par systèmes hybrides réseaux de neurones et modèles de*

- Markov cachés*. Thèse de Doctorat, Université de RENE DISCARTE, Paris 5, 2001.
- [Augustin 06] E. Augustin, M. Carré, E. Grosicki, J. Brodin, E. Geoffrois & F. Prêteux. *RIMES evaluation campaign for handwritten mail processing*. International Workshop on Frontiers in Handwriting Recognition, pages 231–235, 2006.
- [Baird 88] H. S. Baird. *Applications of multidimensional search to structural feature identification*. Syntactic and structural pattern recognition, pages 137–149, 1988.
- [Baldi 95] P. Baldi, Y. Chauvin, T. Hunkapiller & M. Mc. Clure. *Hidden Markov models of biological primary sequence information*. Proceedings of Nature Academic Science, vol. 91, no. 3, pages 1059–1063, 1995.
- [Ball 06] G. Ball, S. N. Srihari & H. Srinivasan. *Segmentation-free and segmentation dependent approaches to Arabic word spotting*. IWFHR, pages 53–58, 2006.
- [Baum 70] L. E. Baum, T. Petrie, G. Soules & N. Weiss. *A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains*. Annals of Mathematical Statistics, vol. 41, no. 1, pages 164–171, 1970.
- [Baum 72] L. E. Baum. *An inequality and associated maximization technique for probabilistic functions of Markov processes*. Inequalities, vol. 3, pages 1–8, 1972.
- [Belaïd 92] A. Belaïd & Y. Belaïd. *Reconnaissance des formes, méthodes et applications*. 1992.
- [Belaïd 97] A. Belaïd & G. Saon. *Utilisation des processus markoviens en reconnaissance de l'écriture*. Traitement du signal, vol. 14, no. 2, pages 161–177, 1997.
- [Bengio 95] Y. Bengio, Y. Le Cun, C. Nohl & C. Burges. *Le-Rec : a NNHMM hybrid for online handwriting recognition*. Neural Computation, vol. 7, no. 6, pages 1289–1303, 1995.

- [Bishop 95] C.M. Bishop. *Neural networks for pattern recognition*. Oxford : Oxford University Press, 1995.
- [Bishop 06] C. M. Bishop. *Pattern recognition and machine learning*. 2006.
- [Bouurlard 96] H. Bouurlard, S. Dupont & C. Riss. *Multi-stream speech recognition*. Rapport technique, IDIAP, Juillet 1996.
- [Bouurlard 97] H. Bouurlard & S. Dupont. *Sub-band-based Speech Recognition*. IEEE Int. Conf. on Acoust., Speech, and Signal Processing, pages 1251–1254, 1997.
- [Chaudhuri 97] B.B. Chaudhuri & U. Pal. *An OCR system to read two Indian language scripts : Bangla and Devnagari (Hindi)*. ICDAR 97, vol. 2, pages 1011 –1015, 1997.
- [Chevalier 03] S. Chevalier, E. Geoffrois & F. Prêteux. *A 2D dynamic programming approach for Markov random field-based handwritten character recognition*. International Conference on Image and Signal Processing, pages 617–630, 2003.
- [Chevalier 04a] S. Chevalier. *Reconnaissance d'écriture manuscrite par des techniques markoviennes : une approche bidimensionnelle et générique*. Thèse de Doctorat, Université René Descartes - Paris 5, 2004.
- [Chevalier 04b] S. Chevalier, E. Geoffrois & F. Prêteux. *Programmation dynamique 2D pour la reconnaissance de caractères manuscrits par champs de Markov*. Reconnaissance des Formes et Intelligence Artificielle, pages 1143–1152, 2004.
- [Cho 95] W. Cho, S.W. Lee & H. Kim. *Modeling and Recognition of Cursive Words with Hidden Markov Models*. Pattern Recognition, vol. 28, no. 12, pages 1941–1953, 1995.
- [Cho 97] S. B. Cho. *Neural network classifier for recognizing totally unconstrained handwritten numerals*. IEEE Transactions on Neural Networks, vol. 29, no. 1, pages 43–53, 1997.

- [Choisy 02] C. Choisy. *Utilisation des modèles stochastiques en vision globale et analytique*. Thèse de Doctorat, Université Nancy 2, 2002.
- [Chow 65] C. K. Chow. *Statistical Independence and Threshold Functions*. IEEE Transactions on Electronic Computers, pages 66–68, 1965.
- [Clocksin 03] W.F. Clocksin & P.P.J. Fernando. *Towards Automatic Transcription of Syriac Handwriting*. Proc. Int'l Conf. Image Analysis and Processing, pages 664–669, 2003.
- [Cohen 91] E. Cohen, J. J. Hull & S. N. Srihari. *Understanding handwritten text in a structured environment : determining ZIP codes from addresses*. Character and handwriting recognition expanding frontiers, vol. 30, pages 221–264, 1991.
- [Cohen 94] E. Cohen, J.J. Hull & S.N. Srihari. *Control structure for interpreting handwritten addresses*. IEEE Trans. on PAMI, vol. 16, no. 10, pages 1049–1055, 1994.
- [Connell 02] S.D. Connell & A.K. Jain. *Writer Adaptation for Online Handwriting Recognition*. IEEE Trans. on PAMI, vol. 24, no. 3, pages 329–346, 2002.
- [Cover 67] T.M. Cover & P.E. Hart. *Nearest neighbour pattern classification*. IEEE Trans. on Inform. Theory, vol. 13, no. 1, pages 21–27, 1967.
- [Cracknell 98] C. Cracknell, A.C. Downton & L. Du. *An object-oriented descriptive language to facilitate advanced handwritten form processing*. IVC, vol. 16, no. 12–13, pages 843–853, 1998.
- [Crettez 98] J.P. Crettez & G. Lorette. *Reconnaissance de l'écriture manuscrite*. Techniques de l'Ingénieur, traité Informatique, vol. H1 358, 1998.
- [Dehghan 01] M. Dehghan, K. Faez, M. Ahmadi & M. Shridhar. *Farsi (Arabic) Word Recognition : A Holistic Approach Using Discrete HMM*. Pattern Recognition, vol. 34, pages 1057–1065, 2001.

- [Dempster 77] A.P. Dempster, N.M. Laird & D.B. Rubin. *Maximum likelihood from incomplete data via the em algorithm*. Journal of the Royal Statistical Society, vol. 39, pages 1–39, 1977.
- [Denoeux 95] T. Denoeux. *A k-nearest neighbor classification rule based on dempster-shafer theory*. IEEE Transactions on Systems, Man and Cybernetics, vol. 25, no. 5, pages 804–813, 1995.
- [Denoeux 00] T. Denoeux. *A neural network classifier based on dempster-shafer theory*. IEEE transactions on Systems, Man and Cybernetics A, vol. 30, no. 2, pages 131–150, 2000.
- [Denoeux 08] T. Denoeux. *Conjunctive and disjunctive combination of belief functions induced by non-distinct bodies of evidence*. Artificial Intelligence, vol. 172, no. 2-3, pages 234–264, 2008.
- [Duzba 97] G. Duzba, A. Filatov, D. Gershuny, I. Kil & V. Nikitin. *Check Amount recognition Based on the cross validation of courtesy and legal amount fields*. In Automatic Bankcheck Processing, pages 177–194. World Scientific, 1997.
- [El-Yacoubi 99] A. El-Yacoubi, M. Gilloux, R. Sabourin & C. Y. Suen. *An HMM Based Approach for Off-line Unconstrained Handwritten Word Modeling and Recognition*. IEEE Trans. on PAMI, vol. 21, no. 8, pages 752–760, 1999.
- [El-Yacoubi 02] M.A. El-Yacoubi, M. Gilloux & J.M. Bertille. *A Statistical Approach for Phrase Location and Recognition within a Text Line : An Application to Street Name Recognition*. IEEE Trans. on PAMI, vol. 24, no. 2, pages 172–188, 2002.
- [Elgammal 01] M. Elgammal & M. A. Ismail. *Techniques for Language Identification for Hybrid Arabic-English Document Images*. International Conference on Document

- Analysis and Recognition ICDAR 01, pages 1100–1104, 2001.
- [Farah 04a] N. Farah, L. Souici, L. Farah & M. Sellami. *Arabic Words Recognition with Classifiers Combination : An Application to Literal Amounts*. Proc. Artificial Intelligence : Methodology, Systems, and Applications, pages 420–429, 2004.
- [Farah 04b] N. Farah, L. Souici, L. Farah & M. Sellami. *Arabic Words Recognition with Classifiers Combination : An Application to Literal Amounts*. Proc. Artificial Intelligence : Methodology, Systems, and Applications, pages 420–429, 2004.
- [Farooq 05] F. Farooq, V. Govindaraju & M. Perrone. *Preprocessing methods for handwritten Arabic documents*. ICDAR, vol. 1, pages 267–271, 2005.
- [Forney 73] G. D. Forney. *The Viterbi Algorithm*. Proceedings of the IEEE, vol. 61, pages 268–278, 1973.
- [Garcia-Salicitti 96] S. Garcia-Salicitti. *Une approche neuronale predictive pour la reconnaissance en-ligne de l'écriture cursive*. Thèse de Doctorat, Université Paris 6, 1996.
- [Goraine 92] H. Goraine, M. Usher & S. Al-Emami. *Off-Line Arabic Character Recognition*. Computer, vol. 25, pages 71–74, 1992.
- [Guyon 95] I. Guyon & F. Pereira. *Design of a linguistic postprocessor using variable memory length Markov models*. Proceedings of ICDAR'95, International Conference on Document Analysis and Recognition, vol. 1, pages 454–457, 1995.
- [Hagen 01] A. Hagen. *Robust speech recognition based on multi-stream processing*. Thèse de Doctorat, Ecole polytechnique fédérale de Lausanne, 2001.
- [Haraty 02] R. Haraty & A. Hamid. *Segmenting Handwritten Arabic Text*. Proc. Int'l Conf. Computer Science, Software Eng., Information Technology, e-Business, and Applications, 2002.

- [Heutte 94] L. Heutte. *Reconnaissance de caractères manuscrits : Application à la lecture automatique des chèques et enveloppes postales*. Thèse de doctorat, Université de Rouen, 1994.
- [Heutte 97] L. Heutte, P. Barbosa-Perreira, O. Bougeois, J.V. Moreau, B. Plessis, P. Courtellemont & Y. Lecourtier. *Multi-bank check recognition system : consideration on the numeral amount recognition module*. IJPRAI, vol. 11, pages 595–618, 1997.
- [Heutte 03] L. Heutte. *Analyse et Reconnaissance de l'Écriture : de Nouvelles Perspectives en Traitements Automatique de Documents Manuscrits*. Habilitation à Diriger les Recherches, Université de Rouen, 2003.
- [H.Sakoe 79] H.Sakoe. *Two-level DP matching - A dynamic programming-based pattern matching algorithm for connected word recognition*. IEEE Transactions of the IECE of Japan, vol. 27, pages 588–595, 1979.
- [Huang 90] X. Huang, Y. Ariki & M. Jack. *Hidden Markov models for speech recognition*. Edinburgh Universal Press, 1990.
- [Jain 00] A.K. Jain, R.P.W. Duin & J. Mao. *Statistical pattern recognition : A review*. IEEE Trans. on PAMI, vol. 22, no. 1, pages 4–37, 2000.
- [Juang 91] B. H. Juang & L. R. Rabiner. *Hidden Markov models for speech recognition*. Technometrics, vol. 33, no. 3, pages 251–272, 1991.
- [Jung 00] K. Jung & H.J. Kim. *On-line recognition of cursive Korean characters using graph representation*. Pattern Recognition, vol. 33, no. 3, pages 399–412, 2000.
- [Katz 87] S. M. Katz. *Estimation of probabilities from spare data for language model component of a speech-recognizer*. IEEE Transactions on ASSP, vol. 35, no. 3, 1987.
- [Keeni 96] K. Keeni, H. Shimodaira, T. Nishino & Y. Tan. *Recognition of Devanagari Characters Using Neural*

- Networks*. IEICE, vol. E79-D, no. 5, pages 523–528, 1996.
- [Khorsheed 03] M.S. Khorsheed. *Recognising Handwritten Arabic Manuscripts Using a Single Hidden Markov Model*. *Pattern Recognition Letters*, vol. 24, pages 2235–2242, 2003.
- [Kim 97] G. Kim & V. Govindaraju. *Bankcheck Recognition Using Cross Validation Between Legal and Courtesy Amounts*. In *Automatic Bankcheck Processing*, pages 195–212. World Scientific, 1997.
- [Kim 98] G. Kim & V. Govindaraju. *Handwritten Phrase Recognition as Applied to Street Name Images*. *Pattern Recognition*, vol. 31, no. 1, pages 41–51, 1998.
- [Kim 00] J. Kim, K. Kim, C. Nadal & C. Suen. *A methodology of combining hmm and mlp classifiers for cursive word recognition*. *ICDAR*, vol. 2, pages 319–322, 2000.
- [Kimura 94] F. Kimura, S. Tsuruoka, Y. Miyake & M. Shridhar. *A Lexicon Directed Algorithm for Recognition of Unconstrained Handwritten Words*. *IEICE Trans. Inf. and Syst.*, vol. E77-D, no. 7, pages 785–793, 1994.
- [Kimurama 94] F. Kimurama, S. Tsuruoka, Y. Miyake & M. Shridhar. *A Lexicon Directed Algorithm for Recognition of Unconstrained Handwritten Words*. *IEICE TRANS. INF. and SYST.*, vol. E77 D, no. 7, pages 785–793, 1994.
- [Klein 08] John Klein. *suivi robuste d'objets dans les séquences d'images par fusion de sources, application au suivi de véhicules dans les scènes routières*. PhD thesis, Université de Rouen (France), 2008.
- [Knerr 97] S. Knerr, V. Asimov, O. Baret, N. Gorsky, D. Price & J.C. Simon. *The A2iA intercheque system : Courtesy amount and legal amount recognition for French checks*. *IJPRAI*, vol. 11, no. 4, pages 505–548, 1997.

- [Koch 04] G. Koch, T. Paquet & L. Heutte. *Combination of contextual information for handwritten word recognition*. IWFHR, pages 468–473, 2004.
- [Koerich 02] A. L. Koerich. *Large vocabulary off-line handwritten word recognition*. Thèse de Doctorat, Ecole de Technologie Supérieure, Université du Québec, Canada, 2002.
- [Koerich 03] A.L. Koerich, R. Sabourin & C.Y. Suen. *Large vocabulary off-line handwriting recognition : A survey*. Pattern Analysis and Applications, vol. 6, pages 97–121, 2003.
- [Koerich 05] A. L. Koerich, R. Sabourin & C. Y. Suen. *Recognition and verification of unconstrained handwritten words*. IEEE Transactions on PAMI, vol. 27, no. 10, pages 1509–1522, 2005.
- [Kuo 94] S. Kuo & O. Agazi. *Keyword spotting in poorly printed documents using pseudo 2-D hidden Markov models*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 8, pages 842–848, 1994.
- [Lecun 87] Y. Lecun. *Modèles connexionnistes de l'apprentissage*. Thèse de doctorat, Université P. et M. Curie (Paris 6), 1987.
- [LeCun 98] Y. LeCun, L. Bottou, Y. Bengio & P. Haffner. *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, vol. 86, no. 11, pages 2278–2324, 1998.
- [Lee 98] J.J. Lee, M. Nakajima & J. Kim. *A Hierarchical HMM Network-based Approach for On-line Recognition of Multi-Lingual Cursive Handwritings*. IEICE Transactions on Information and Systems, vol. E81-D, no. 8, pages 881–888, 1998.
- [Leroux 97] M. Leroux, E. Lethelier, M. Gilloux & B. Lemaire. *Automatic Reading of handwritten amounts on*

- french checks*. In Automatic Bankcheck Processing, pages 157–176. World Scientific, 1997.
- [Levin 92] E. Levin & R. Pieraccini. *Dynamic planar warping for optical character recognition*. IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 3, pages 149–152, 1992.
- [Likforman-Sulem 95] L. Likforman-Sulem, A. Hanimyan & C. Faure. *A Hough based algorithm for extracting text lines in handwritten documents*. ICDAR, vol. 2, pages 774–777, 1995.
- [Likforman-Sulem 07] L. Likforman-Sulem, A. Zahour & B. Taconet. *Text line segmentation of historical documents : a survey*. IJDAR, vol. 9, no. 2, pages 123–138, 2007.
- [Likforman-Sulem 08] L. Likforman-Sulem & M. Sigelle. *Recognition of degraded characters using Dynamic Bayesian Networks*. Pattern Recognition, vol. 41, no. 10, pages 3092–3103, 2008.
- [Liu 00] C.L. Liu & M. Nakagawa. *Precise Candidate Selection for Large Character Set Recognition by Confidence Evaluation*. IEEE Trans. on PAMI, vol. 22, no. 6, pages 636–641, 2000.
- [Liu 02] C. L. Liu, K. Nakashima, H. Sako & H. Fujisawa. *Handwriting digit recognition using state-of-the-art techniques*. International Workshop on Frontiers in Handwriting Recognition, pages 320–325, 2002.
- [Maddouri 02] S. Snoussi Maddouri, H. Amiri, A. Belaid & C. Choisy. *Combination of Local and Global Vision Modeling for Arabic Handwritten Words Recognition*. International Workshop on Frontiers in Handwriting Recognition, pages 128–135, 2002.
- [Malaviya 96] A. Malaviya, C. Leja & L. Peters. *Multi-Script Handwriting Recognition with*. In FOHDEL, NAFIPS, pages 147–151, 1996.
- [Marti 01] U.V. Marti & H. Bunke. *Text Line Segmentation and Word Recognition in a System for General Wri-*

- ter Independent Handwriting Recognition*. ICDAR, pages 159–163, 2001.
- [McCowan 05] I. McCowan, D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard & D. Zhang. *Automatic Analysis of Multimodal Group Actions in Meetings*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 3, pages 305–317, 2005.
- [Miled 98] H. Miled. *Stratégies de résolution en reconnaissance de l'écriture semi-cursive : application aux mots manuscrits arabes*. Thèse de doctorat, Université de Rouen, 1998.
- [Miled 01] H. Miled & N.E. Ben Amara. *Planar Markov Modeling for Arabic Writing Recognition : Advancement State*. Proc. Int'l Conf. Document Analysis and Recognition, pages 69–73, 2001.
- [Milewski 06] R. Milewski & V. Govindaraju. *Extraction of Handwritten Text from Carbon Copy Medical Form Images*. DAS06, pages 106–116, 2006.
- [Morgan 95] N. Morgan & H. Bourlard. *Continuous speech recognition : An introduction to the hybrid HMM/connectionist approach*. IEEE Signal Processing Magazine, pages 25–42, 1995.
- [Morita 03] M. Morita, R. Sabourin, F. Bortolozzi & C. Y. Suen. *Segmentation and recognition of handwritten dates : an HMM-MLP hybrid approach*. International Journal on Document Analysis and Recognition, vol. 6, no. 4, pages 248–262, 2003.
- [Mozaffari 05] S. Mozaffari, K. Faez & M. Ziaratban. *Structural Decomposition and Statistical Description of Farsi/Arabic Handwritten Numeric Characters*. ICDAR, pages 237–241, 2005.
- [Märgner 05] V. Märgner, M. Pechwitz & H. ElAbed. *ICDAR 2005 Arabic handwriting recognition competition*. International Conference on Document Analysis and Recognition, vol. 1, pages 70–74, 2005.

- [Märgner 07] V. Märgner & H. ElAbed. *ICDAR 2007 Arabic Handwriting Recognition Competition*. International Conference on Document Analysis and Recognition, pages 1274–1278, 2007.
- [Nagy 06] G. Nagy, S. Seth & X. Zhang. *Multi-Character Field Recognition for Arabic and Chinese Handwriting*. Proceedings of the Summit on Arabic and Chinese Handwriting Recognition, pages 93–100, 2006.
- [Nosary 02] A. Nosary. *Reconnaissance Automatique de Textes Manuscrits pas adaptation du scripteur*. Thèse de doctorat, Université de Rouen, 2002.
- [Olivier 96] C. Olivier, H. Miled, K. Romeo & Y. Lecourtier. *Segmentation and coding of Arabic handwritten words*. Proceedings of ICPR'96, vol. 3, pages 264–268, 1996.
- [Otsu 66] N. Otsu. *A thresholding selection method from gray-level histogram*. IEEE Trans. on Systems, Man, and Cybernetics, vol. 9, page 1979, 62-66.
- [Park 95] H.S. Park & S.W. Lee. *Hidden Markov mesh random field : theory and its application to handwritten character recognition*. International Conference on Document Analysis and Recognition, vol. 1, pages 409–412, 1995.
- [Park 96a] H. Park & S. Lee. *An HMMRF-based statistical approach for off-line handwritten character recognition*. International Conference on Pattern Recognition, vol. 2, pages 320–324, 1996.
- [Park 96b] H.S. Park & S.W. Lee. *Off-Line Recognition of Large-Set Handwritten Characters with Multiple Hidden Markov-Models*. Pattern Recognition, vol. 29, no. 2, pages 231–244, 1996.
- [Park 96c] H.S. Park, J.S. Park & S.W. Lee. *A new methodology for off-line handwritten character recognition using truly 2-d statistical model*. International Workshop on Frontiers in Handwriting Recognition, pages 53–60, 1996.

- [Pechwitz 02] M. Pechwitz, S.S. Maddouri, V. Maergner, N. El-louze & H. Amiri. *IFN/ENIT - database of Handwritten Arabic Words*. Colloque International Francophone sur l'Ecrit et le Doucement, pages 129–136, 2002.
- [Pechwitz 03] M. Pechwitz & V. Margner. *HMM Based Approach for Handwritten Arabic Word Recognition Using the IFN/ENIT-Database*. Proc. Int'l Conf. Document Analysis and Recognition, pages 890–894, 2003.
- [Potamianos 98] G. Potamianos & H.P. Graf. *Discriminative training of HMM stream exponents for audio-visual speech recognition*. Proceeding IEEE Conference Acoustics, Speech, and Signal Processing, vol. 6, pages 3733–3736, 1998.
- [Powalka 97] R. K. Powalka, N. Sherkat & R. J. Withrow. *Word shape analysis for hybrid recognition system*. Pattern Recognition, vol. 30, no. 3, pages 421–445, 1997.
- [Prevost 03] L. Prevost, C. Michel-Sendis, A. Moises, L. Oudot & M. Milgram. *Combining model-based and discriminative classifiers : application to hadwritten character recognition*. International Conference on Document Analysis and Recognition, 2003.
- [Rabiner 89] L. R. Rabiner. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. Proceedings of the IEEE, vol. 77, no. 2, pages 257–286, 1989.
- [Ramasso 07a] E. Ramasso, M. Rombaut & D. Pellerin. *Forward-Backward-Viterbi procedures in the Transferable Belief Model for state sequence analysis using belief functions*. European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, pages 405 – 417, 2007.
- [Ramasso 07b] Emmanuel Ramasso. *Reconnaissance de sequences d'etats par le modele des croyances transferables*. Ap-

- plication a l'analyse de video d'athletisme*. PhD thesis, Université de Grenoble (France), 2007.
- [Ramdane 03] S. Ramdane, B. Taconet & A. Zahour. *Classification of forms with handwritten fields by planar hidden Markov models*. Pattern Recognition, vol. 36, no. 4, pages 1045–1060, 2003.
- [Romeo-Pakker 95] K. Romeo-Pakker, A. Ameer, C. Olivier & Y. Lecourtier. *Structural analysis of Arabic handwriting : segmentation and recognition*. Machine Vision and Applications, vol. 8, pages 232–240, 1995.
- [Saon 97] G. Saon. *Modèles markoviens uni- et bidimensionnels pour la reconnaissance de l'écriture manuscrite hors-ligne*. Thèse de Doctorat, Université Henri Poincaré - Nancy1, 1997.
- [Sari 02] T. Sari, L. Souici & M. Sellami. *Off-line Handwritten Arabic Character Segmentation and Recognition System : ACSA-RECAM*. Proceedings of IWFHR'02, pages 452–457, 2002.
- [Sari 05] T. Sari & M. Sellami. *Cursive Arabic script segmentation and recognition system*. International Journal of Computers and Application, vol. 27, no. 3, pages 161–168, 2005.
- [Sayre 73] K. M. Sayre. *Machine recognition of Handwritten words : A project report*. Pattern Recognition, vol. 5, pages 213–228, 1973.
- [Schambach 08] M. P. Schambach, J. Rottland & T. Alary. *How to Convert a Latin Handwriting Recognition System to Arabic*. In International Conference on Frontiers of Handwriting Recognition, 2008.
- [Schenkel 95] M. Schenkel, I. Guyon & D. Henderson. *On-line cursive script recognition using time delay neural networks and hidden Markov models*. Machine Vision and Applications, vol. 8, no. 4, pages 215–223, 1995.
- [Scott 00] R. E. Scott. *Maximum likelihood estimation : Logic and practice*. 2000.

- [Seni 96] G. Seni, R.K. Srihari & N. Nasrabadi. *Large Vocabulary Recognition Of Online Handwritten Cursive Words*. IEEE Trans. on PAMI, vol. 18, no. 7, pages 757–762, 1996.
- [Senior 98] A. W. Senior & A. J. Robinson. *An Off-Line cursive handwriting recognition system*. IEEE Transactions on PAMI, vol. 20, no. 3, pages 309–321, 1998.
- [Shafer 76] G. Shafer. *A mathematical theory of evidence*. Princeton University Press, 1976.
- [Shahabi 06] S. Shahabi & M. Rahmati. *Comparison of Gabor-based features for writer identification of Farsi/Arabic handwriting*. IWFHR'06, pages 545–550, 2006.
- [Smets 90] P. Smets. *The combination of evidence in the transferable belief model*. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 12, no. 5, pages 447–458, 1990.
- [Smets 93] P. Smets. *Belief functions : the disjunctive rule of combination and the generalized Bayesian theorem*. Int. Journal of Approximate Reasoning, vol. 9, pages 1–35, 1993.
- [Smets 94] P. Smets & R. Kennes. *The transferable Belief Model*. Artificial Intelligence, vol. 66, no. 2, pages 191–234, 1994.
- [Snoussi-Maddouri 02] S. Snoussi-Maddouri, A. Belaid, C. Choisy & H. Amiri. *Modèle perceptif neuronal à vision globale-locale pour la reconnaissance de mots manuscrits arabes*. CIFED'02, pages 11–20, 2002.
- [S.Omachi 00] S.Omachi, F. Sun & H. Aso. *A Noise-Adaptive Discriminant Function and Its Application to Blurred Machine Printed Kanji Recognition*. IEEE Trans. on PAMI, vol. 22, no. 3, pages 314–319, 2000.
- [Srihari 97] S. N. Srihari, T. Hong & G. Srikantan. *Machine-printed Japanese document recognition*. Pattern Recognition, vol. 30, no. 7, pages 1301–1313, 1997.

- [Srihari 05] S. N. Srihari, H. Srinivasan, P. Babu & C. Bhol. *Handwritten Arabic word spotting using CEDARA-BIC document analysis system*. Proceedings of SDIUT, pages 123–132, 2005.
- [Srihari 06] S. N. Srihari, H. Srinivasan, P. Babu & C. Bhol. *Spotting words in handwritten Arabic documents*. Document Recognition and Retrieval XIII, Taghva K., Lin X. (Eds), Proceedings Electronic Imaging, SPIE, pages 606701–1 to 606702–10, 2006.
- [Syiam 06] M. Syiam, T. M. Nazmy, A. E. Fahmy, H. Fathi & K. Ali. *Histogram clustering and hybrid classifier for handwritten arabic characters recognition*. Proceedings of the 24th IASTED International Multi-Conference Signal Processing, Pattern Recognition, and Applications, pages 44–49, 2006.
- [Tappert 84] C. C. Tappert. *Adaptive on-line handwriting recognition*. ICPR, pages 1004–1007, 1984.
- [Tay 01] Y. H. Tay, P.M Lallican, M. Khalid, C. Viard-Gaudin & S. Knerr. *An Offline Cursive Handwritten Word Recognition System*. Proceedings of IEEE Region 10 Conference, vol. 2, pages 519–524, 2001.
- [Vannoorenberghe 03] P. Vannoorenberghe. *Un état de l'art sur les fonctions de croyance appliquées au traitement de l'information*. Information interaction intelligence, vol. 3, no. 2, pages 9–45, 2003.
- [Vapnik 95] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.
- [Varga 90] A. P. Varga & R. K. Moore. *Hidden Markov model decomposition of speech and noise*. Proc. IEEE Internat. Conf. Acoust. Speech and Signal Process, pages 845–848, 1990.
- [Viard-Gaudin 05] C. Viard-Gaudin, P. Lallican & S. Knerr. *Recognition-directed recovering of temporal information from handwriting images*. Pattern Recognition Letters, vol. 26, no. 16, pages 2537–2548, 2005.

- [Vinciarelli 00] A. Vinciarelli & J. Luettin. *Offline cursive script recognition based on continuous density HMM*. IWFHR, vol. 2, pages 493–498, 2000.
- [Vinciarelli 04] A. Vinciarelli, S. Bengio & H. Bunke. *Offline Recognition of Unconstrained Handwritten Texts Using HMMs and Statistical Language Models*. IEEE Trans. on PAMI, vol. 26, no. 6, pages 709–720, 2004.
- [Viterbi 67] A. J. Viterbi. *Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm*. IEEE Transactions on Information Theory, vol. 13, no. 2, pages 260–269, 1967.
- [Walley 99] P. Walley & S. Moral. *Upper probabilities based only on the likelihood function*. Journal of the Royal Statistical Society, vol. B, no. 61, pages 831–847, 1999.
- [Wellekens 98] C. J. Wellekens, J.Kangasharju & C.Milesi. *The use of meta-HMM in multi-stream HMM training for automatic speech recognition*. Intl. Conference on Spoken Language Processing, pages 2991–2994, 1998.
- [Xu 92] L. Xu, A. Krzyzak & C. Suen. *Methods of combining multiple classifiers and their applications to handwriting recognition*. IEEE Trans. Systems Man Cybernet, vol. 22, no. 3, pages 418–435, 1992.
- [Yaghlane 06] A. Ben Yaghlane, T. Denoeux & K. Mellouli. *Elicitation of expert opinions for constructing belief functions*. Dans Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, vol. 2, no. 7, pages 403–411, 2006.
- [Zhang 00] G.P. Zhang. *Neural Networks for Classification : A Survey*. IEEE Trans on Systems, Man and Cybernetics - Part C, vol. 30, no. 4, pages 641–662, 2000.
- [Zheng 04] L. Zheng, A.H. Hassin & X. Tang. *A new algorithm for machine printed Arabic character segmentation*. Pattern Recognition Letters, vol. 25, no. 15, pages 1723–1729, 2004.

- [Zhu 97] J. Zhu, T. Hong & J. J. Hull. *Image-based Keyword Recognition in Oriental Language Document Images*. Pattern Recognition, vol. 30, no. 8, pages 1293–1300, 1997.
- [Zouari 04] H. Zouari. *Contribution à l'évaluation des méthodes de combinaison parallèle de classifieurs par simulation*. Thèse de Doctorat, Université de Rouen, 2004.