

N° d'ordre :



Université Ibn Zohr  
Faculté des Sciences d'Agadir

## THÈSE DE DOCTORAT ÈS-SCIENCES

*Présentée par*

**MUSTAPHA AMROUCH**

**UFR : Mathématiques et Informatique Appliquées**

**Option : Informatique**

---

**Reconnaissance de caractères imprimés et  
manuscrits, textes et documents basée sur les  
modèles de Markov cachés**

---

## **Remerciements**

## Résumé

Les travaux présentés dans cette thèse se situent dans le cadre de la reconnaissance de l'écriture manuscrite et imprimée par les Modèles de Markov cachés. Celle-ci consiste à mieux déchiffrer les caractères présents dans l'image.

Dans un premier temps, nous proposons un système de reconnaissance de caractères manuscrits isolés, basé sur une modélisation markovienne de type modèle discriminant (MD- HMM). L'accumulateur de Hough de chaque image de caractère est partitionné en fenêtres verticales, qui nous utiliserons pour extraire une information directionnelle. L'information obtenue est traduite en séquences d'observations qui sont utilisées pour entraîner le modèle de chaque caractère lors de la phase d'apprentissage. L'évaluation du système de reconnaissance proposé est menée sur une base locale des caractères arabes manuscrits isolés et la base AMHCD des caractères amazighes manuscrits. Les résultats enregistrés montrent l'efficacité de la modélisation proposée, notamment dans le cas continu, où les densités de probabilités sont modélisées par les gaussiennes.

En second temps, nous présentons une nouvelle méthode pour la reconnaissance automatique hors ligne de caractères Tifinaghs imprimés. La méthode proposée est basée sur un chemin discriminant (DP-HMM) opérant sur un vocabulaire de base formé de différents graphèmes fondamentaux. Ce vocabulaire est généré en se basant sur les caractéristiques morphologiques de la graphie Amazighe. Un seul modèle HMM globale construit et entraîné sur les éléments du vocabulaire proposé par des primitives structurelles et géométriques. Chaque chemin au long de ce treillis représente une séquence de segments, qui constitue un caractère de l'alphabet tifinagh. Pour se faire, les caractères d'entrés sont pré classés en deux groupes (forme circulaire et non circulaire) et par la suite sont décrits par leurs points d'intérêts et leurs segments. La reconnaissance s'effectue en décodant dynamiquement le chemin optimal suivant le critère de maximum de vraisemblance. Les scores obtenus montrent la robustesse de l'approche proposée.

## Mots-clés

Modèles de Markov cachés, reconnaissance de l'écriture manuscrite et imprimée, primitives structurelles et directionnelles, transformation de Hough.

## **Abstract**

This thesis deals with offline printed and handwriting recognition using Hidden Markov Models. This task consists in automatically reading characters from an image.

At first, we present a recognition system for isolated handwritten characters, based on Markov modeling with discriminating model strategy (DM-HMM). The Hough accumulator of the character image is partitioned into equal vertical windows that will be used to extract directional information. This information is translated into sequences of observations that are used to train the model for each character during the learning step.

The developed system has been tested on two available databases :a local database of isolated Arabic handwritten characters and the database of isolated Amazigh handwritten characters (AMHCD). The results obtained show the effectiveness of the proposed modeling, especially in the continuous case, where the probability densities are modeled by Gaussian.

Second time, we present a new approach for off-line printed Tifinagh characters recognition. The method is based on the discriminating path (DP-HMM) operating on a basic vocabulary that consists of various fundamental strokes. This vocabulary is generated using the intrinsic morphological characteristics of the amazigh script. Only one model is built and trained on all elements of the proposed vocabulary by structural and geometric features. Each path through this trellis represents a sequence of segments, i.e. the character of the Tifinagh alphabet. The input characters are first pre-classified into one of two character groups (circular, non-circular), and then described with their feature points and segments (arc, horizontal, vertical and diagonal). Finally, the recognition is performed by dynamically decoding the optimal path according to the criterion of maximum likelihood. The developed method has been evaluated on the database of the isolated amazighe printed characters. The obtained scores show the robustness of the proposed approach

## **Keywords**

Hidden Markov Models, printed and handwriting recognition, directional structural primitives, Hough transform.

# Table des matières

<b>Remerciements</b> .....	<b>II</b>
<b>Résumé</b> .....	<b>iii</b>
<b>Abstract</b> .....	<b>iv</b>
<b>Table des matières</b> .....	<b>v</b>
<b>Liste des tableaux</b> .....	<b>viii</b>
<b>Liste des figures</b> .....	<b>ix</b>
<b>Liste des abréviations</b> .....	<b>xiii</b>
<b>Introduction générale</b> .....	<b>1</b>
<b>PARTIE I :</b> .....	<b>1</b>
<b>ÉTUDE BIBLIOGRAPHIQUE</b> .....	<b>1</b>
<b>Chapitre 1 : Généralités sur les systèmes de reconnaissance automatique d'écriture</b> ...2	
1. Introduction.....	2
2. Préliminaires de la reconnaissance d'écriture.....	2
2.1. Types et styles d'écritures.....	3
2.2. Scripteurs et vocabulaires .....	4
2.3. Évolution de la difficulté du traitement de l'écriture .....	5
2.4. Systèmes en ligne.....	6
2.5. Systèmes hors ligne .....	7
2.6. Stratégies de reconnaissance.....	7
2.6.1. Stratégie globale ou holistique.....	8
2.6.2. Stratégie locale ou analytique .....	8
3. Système de reconnaissance de l'écriture hors ligne.....	9
3.1. Architecture .....	9
3.2. Acquisition.....	10
3.3. Prétraitements et normalisation .....	10
3.3.1. Filtrage .....	11
3.3.2. Seuillage.....	12
3.3.2.1. Seuillage global.....	13
3.3.2.2. Seuillage local.....	13
3.3.3. Squelettisation.....	15
3.3.4. Normalisation.....	17
3.3.4.1. Normalisation de la taille.....	17
3.3.4.2. Correction de l'inclinaison des lignes (Skew correction).....	18
3.3.4.3. Correction de l'inclinaison des caractères (Slant correction) .....	19
3.3.4.4. Estimation des lignes de base .....	20
3.4. Segmentation .....	21
3.4.1. Segmentation explicite.....	22
3.4.2. Segmentation implicite .....	22
3.5. Extraction de primitives.....	24
3.5.1. Caractéristiques statistiques .....	24
3.5.2. Caractéristiques structurelles .....	25
3.5.3. Transformations globales.....	25

3.6.	Apprentissage .....	27
3.7.	Classification .....	28
3.7.1.	Méthodes connexionnistes .....	28
3.7.2.	Méthodes structurelles et syntaxiques .....	30
3.7.3.	Machines à vecteurs supports : Support Vector Machine (SVM ) .....	30
3.7.4.	Modèles de Markov cachés .....	31
3.7.5.	Combinaison de classifieurs .....	32
3.8.	Post-traitements .....	33
4.	Mesure des performances .....	34
5.	Domaines d'applications de la reconnaissance de l'écriture .....	34
6.	Conclusion .....	35
<b>Chapitre 2 :</b>	<b>Les Modèles de Markov Cachés .....</b>	<b>36</b>
1.	Introduction .....	36
2.	Définitions préliminaires .....	36
2.1.	Variable aléatoire .....	36
2.2.	Processus stochastique .....	37
2.3.	Chaîne de Markov .....	37
3.	Modèles de Markov cachés unidimensionnels ( HMM 1D) .....	37
3.1.	Topologie des modèles de Markov cachés .....	39
3.2.	Types de densité d'observations .....	40
3.3.	Problèmes fondamentaux associés aux HMMs .....	41
3.3.1.	Évaluation de la probabilité d'une séquence d'observations .....	42
3.3.2.	Analyse et décodage .....	45
3.3.3.	Apprentissage et calcul des paramètres du modèle .....	46
3.4.	Modèle discriminant et chemin discriminant .....	49
4.	Modèles pseudo bidimensionnels : Planar HMM(PHMM ) .....	50
4.1.	Procédure d'apprentissage des PHMMs .....	52
4.2.	Procédure de reconnaissance des PHMMs .....	52
5.	Modèles de Markov bidimensionnels : Champs de Markov (HMM2D) .....	53
6.	Domaines d'application des Mdèles de Markov cachés .....	54
7.	Conclusion .....	55
<b>Chapitre 3 :</b>	<b>La Reconnaissance d'écriture par les Modèles de Markov cachés .....</b>	<b>56</b>
1.	Introduction .....	56
2.	Architecture de référence des systèmes à base HMM .....	56
2.1.	Segmentation .....	57
2.2.	Extraction de primitives .....	58
2.3.	Modélisation Markovienne .....	59
3.	Bases de données de référence (Benchmark) .....	61
3.1.	IFN/ENIT .....	62
3.2.	AMHCD .....	62
3.3.	Base des patterns de la graphie amazighe .....	63
3.4.	MNIST .....	63
3.5.	IAM-DB .....	63
3.6.	Cambridge .....	63
4.	Systèmes et Approches antérieurement proposées .....	64
4.1.	Les modèles de Markov cachés unidimensionnel (HMM 1D) .....	64

4.1.1.	Approches globales.....	64
4.1.2.	Approches Analytiques.....	68
4.2.	Les modèles HMMs planaires .....	83
4.2.1.	Approches globales.....	83
4.2.2.	Approches Analytiques.....	85
4.3.	Modèles HMM bidimensionnels (Champs de Markov ou HMM 2D) .....	88
5.	Conclusion .....	92
<b>PARTIE II:.....</b>		<b>94</b>
<b>CONTRIBUTIONS .....</b>		<b>94</b>
<b>Chapitre 4 :Reconnaissance de caractères arabes et amazighes manuscrits isolés par les Modèles de Markov cachés.....</b>		<b>95</b>
6.	Introduction.....	95
7.	Présentation de l'écriture arabe et amazighe .....	96
7.1.	Caractéristiques morphologiques de l'écriture arabe.....	96
7.2.	Caractéristiques morphologiques de l'écriture amazighe.....	98
8.	Architecture du système proposé.....	99
8.1.	Prétraitements .....	100
8.1.1.	Binarisation.....	100
8.1.2.	Lissage .....	101
8.1.3.	Squelettisation.....	102
8.1.4.	Normalisation de taille.....	103
8.2.	Extraction de primitives.....	104
8.2.1.	Transformation de Hough.....	105
8.2.2.	Génération des primitives directionnelles.....	108
9.	Modélisation markovienne .....	112
9.1.	Choix de modèle .....	112
9.2.	Le modèle de caractère .....	113
9.3.	Apprentissage des modèles HMMs .....	114
9.4.	Classification .....	115
10.	Expérimentations et résultats .....	116
10.1.	Base de donnée utilisée.....	116
10.1.1.	Cas de caractères arabes .....	116
10.1.2.	Cas de caractères amazighes.....	116
10.2.	Tests et résultats.....	120
10.2.1.	Cas des caractères arabes.....	120
10.2.2.	Cas des caractères amazighes .....	120
11.	Conclusion et perspectives.....	122
<b>Chapitre 5 :Reconnaissance de textes amazighes imprimés par les Modèles de Markov cachés .....</b>		<b>124</b>
1.	Introduction.....	124
2.	Graphèmes de base du caractère amazighe.....	125
3.	Synopsis du système développé.....	127
4.	Prétraitements .....	128
4.1.	Binarisation.....	128
4.2.	Lissage ou réduction du bruit.....	129
4.3.	Correction d'inclinaison des lignes de texte (Skew correction) .....	129

4.4.	Segmentation d'un bloc de texte amazighe en lignes .....	130
4.5.	Segmentation d'une ligne de texte amazighe en caractères .....	131
4.6.	Normalisation de la taille et recadrage .....	132
4.7.	Squelettisation .....	132
5.	Pré classification des caractères amazighes .....	133
6.	Extraction de primitives .....	136
7.	Modèles HMMs de segments .....	139
8.	Apprentissage .....	141
9.	Classification .....	142
10.	Expériences et résultats .....	143
10.1.	Base de données utilisée.....	143
10.2.	La plate forme logicielle HTK .....	144
10.3.	Tests et résultats .....	146
11.	Version améliorée du système proposé.....	149
11.1.	Pré classification des caractères amazighes : version améliorée.....	150
11.2.	Extraction de primitives .....	152
11.3.	Analyse des résultats .....	153
12.	Conclusion et perspectives .....	154
	<b>Conclusion générale et perspectives .....</b>	<b>155</b>
	<b>Bibliographie.....</b>	<b>158</b>

## Liste des tableaux

Tableau 1-1 :	Masque filtre moyennneur $3 \times 3$ .....	12
Tableau 1-2 :	Exemples de certaines caractéristiques utilisées en reconnaissance de l'écriture .....	27
Tableau 3-1 :	Un exemple d'image de la base de données IfN/ENIT. Les symboles M, B, A, E présentent la position des caractères dans les mots (au milieu, au début, seul, ou à la fin d'un mot).....	62
Tableau 3-2 :	Caractéristiques et performances de certains systèmes à base des HMMs 1D de reconnaissance de l'écriture manuscrite .....	82
Tableau 3-3 :	Caractéristiques et performances des systèmes à base des PHMM s décrits précédemment pour la reconnaissance de l'écriture .....	88
Tableau 3-4 :	paramètres de la modélisation associée au taux 86,3. ....	91
Tableau 3-5 :	Caractéristiques et performances des systèmes à base champs de Markov précités. ....	91
Tableau 4-1 :	Correspondance entre symboles émis et angles de base.....	110
Tableau 4-2 :	Exemples de caractères amazighes manuscrits issus de la base AMHCD [Essa11].....	118
Tableau 4-3 :	Exemples de caractères amazighes manuscrits issus de la base AMHCD [Essa11].....	119
Tableau 4-4 :	Taux de reconnaissance obtenus sur certains caractères arabes .....	120
Tableau 4-5 :	Pourcentages des erreurs de reconnaissance .....	121



Tableau 4-6 : Taux de reconnaissance en fonction des nombre d'états et de nombre de gaussiennes .....	121
Tableau 5-1 : Liste de graphèmes de base .....	126
Tableau 5-2 : Quelques caractères avec leurs décompositions appropriées .....	126
Tableau 5-3 : Voisins de P (sens horaire de parcours): p1, p2, p3, p4, p5, p6, p7, p8 .....	137
Tableau 5-4 : Correspondance en angle.....	137
Tableau 5-5: Exemples de primitives calculées.....	139
Tableau 5-6 : Exemples de quelques caractères dans la base des patterns de la graphie amazighe .....	144
Tableau 5-7 : Quelques caractéristiques de la librairie HTK.....	145
Tableau 5-8 : Taux de reconnaissance sur BD1 .....	146
Tableau 5-9 : Taux de reconnaissance sur BD2 .....	148
Tableau 5-10: Taux de reconnaissance sur BD1 .....	153

## Liste des figures

Figure 1-1 : Catégories et formes de l'écriture .....	3
Figure 1-2 : Le mot « Rome » écrit par différents scripteurs [Lema07].....	4
Figure 1-3 : variabilités d'écritures selon les scripteurs .....	4
Figure 1-4 : Difficultés de l'écriture.....	5
Figure 1-5 : Exemples des systèmes en-ligne .....	6
Figure 1-6 : Diagramme de référence d'un système de reconnaissance.....	9
Figure 1-7 : Exemple d'égalisation d'histogramme .....	11
Figure 1-8 : Problème de seuillage global .....	13
Figure 1-9 : résultats de seuillage globales sur les manuscrits arabes [Kefa10].....	15
Figure 1-10 : résultats de seuillage locales sur les manuscrits arabes [Kefa10].....	15
Figure 1-11 : Exemples de squelettisation.....	16
Figure 1-12 : Exemples de mots manuscrits avec des tailles différentes.....	18
Figure 1-13 : la technique des histogrammes de projection pour la détection d'inclinaison [Haji05].....	18
Figure 1-14 : (a) image original avec l'angle d'inclinaison $\theta_s$ (b) la correction d'inclinaison par les histogrammes projection [Loul09]. .....	19
Figure 1-15 : Inclinaison de l'écriture et histogrammes de projection verticale correspondants [Mena08] .....	20
Figure 1-16 : (a) texte incliné avec une angle d'inclinaison par rapport à l'axe vertical; (b) texte corrigé [Loul09] .....	20
Figure 1-17 : Topologie d'un mot [Beno07] .....	20
Figure 1-18 : Histogramme de projection horizontale et bande de base [Adam06] .....	21
Figure 1-19 : Différentes segmentation pour le mot quatorze [Augu01]. .....	23
Figure 1-20 : Exemple d'RN : perceptron Multicouche à une couche caché.....	28
Figure 1-21 : (a) : Modèle biologique, (b) : modèle mathématique. ....	29
Figure 1-22 : SVM: Hyperplan séparateur de deux classes $w_1$ et $w_2$ . ....	31
Figure 1-23 : Quelques exemples d'architecture de HMM (a) : linéaire, (b) : parallèle, (c) ergodique .....	32

Figure 1-24 : schéma de combinaison séquentielle.....	32
Figure 1-25 : Schéma de combinaison parallèle .....	33
Figure 2-1 : les composantes cachées et observables de modèle .....	38
Figure 2-2 : Représentation graphique d'un modèle HMM à 3 états.....	39
Figure 2-3 : HMM ergodique à 3 états.....	39
Figure 2-4 : HMM gauche droite à 4 états.....	40
Figure 2-5 : Calcul de la variable Forward.....	43
Figure 2-6 : Calcul de la variable Backward.....	44
Figure 2-7 : Représentation de décodage.....	45
Figure 2-8 : Le retour en arrière.....	46
Figure 2-9 : Exemple d'architecture d'un PHMM, d'après [Bela97].....	51
Figure 2-10 : Exemple de voisinage.....	54
Figure 3-1 : Synopsis de système de reconnaissance à base HMM de l'écriture.....	57
Figure 3-2 : <i>Modèle globale comportant les modèles de caractères</i> .....	58
Figure 3-3: (a) :Techniques de fenêtres glissantes [Tay01], (b) : fenêtres uniformes et non uniformes [Beno07].....	59
Figure 3-4 : Topologies (a) linéaire, (b) Bakis.....	60
Figure 3-5 : Extraction de primitives par le principe de fenêtrage par [Dehg01a].....	67
Figure 3-6 : Extraction de vecteurs de primitives avec des fenêtres inclinées [El-Ha06] ...	69
Figure 3-7 : Extraction de primitives par les fenêtres glissantes à trois colonnes de pixels dans [Pech02][Pech03].....	71
Figure 3-8 : Extraction des primitives par zonage [Pech06].....	72
Figure 3-9 : Exemple de squelettisation et approximation linéaire du mot manuscrit [Khor03].....	72
Figure 3-10 : Caractéristiques perceptuelles avec leurs attributs continus [Xue06].....	73
Figure 3-11 : Topologie d'un HMM modélisant un caractère selon [Scha08] .....	75
Figure 3-12 : Architecture d'un pseudo mot par PHMM, d'après [Bena99].....	85
Figure 3-13 : modélisation de caractère imprimé h par un PHMM [Kuo94].....	86
Figure 3-14 : Extraction de primitives par fenêtre glissante [Bipp97].....	87
Figure 3-15 : (a) : modélisation utilisée, (b) : voisinages utilisés [Saon97].....	90
Figure 4-1 : Alphabet arabe.....	96
Figure 4-2 : Lettres arabes avec signes diacritiques.....	97
Figure 4-3 : Les ascendants et descendants entourés et la ligne de base en rouge.....	97
Figure 4-4 : <i>Le mot technologie en arabe est composé de 3 pseudos mots</i> .....	97
Figure 4-5 : Les caractères arabes dans leurs formes isolées .....	97
Figure 4-6 : Alphabet amazighe .....	98
Figure 4-7 : Exemple du texte amazighe dans un manuel scolaire [Essa12] .....	99
Figure 4-8 : Chaîne de traitement du système proposé .....	100
Figure 4-9 : Image du caractère Mim ( م ) avant et après binarisation .....	101
Figure 4-10 : Image du caractère yae ( ي ) avant et après la binarisation.....	101
Figure 4-11 : Image du caractère kaf avant et après le moyannage .....	102
Figure 4-12: Image du caractère yae ( ي ) avant et après le moyannage .....	102
Figure 4-13: Exemples de caractères arabes manuscrits et leurs squelettes obtenus. ....	103
Figure 4-14: Exemples de caractères amazighes manuscrits et leurs squelettes obtenus. .	103
Figure 4-15 : Exemples de caractères arabes isolés manuscrits et leurs tailles normalisées obtenues.....	104

Figure 4-16: Exemples de caractères amazighes manuscrits et leurs tailles normalisées obtenues. ....	104
Figure 4-17 : Exemple de ligne droite et les paramètre $(\rho, \theta)$ . ....	106
Figure 4-18 : Transformée de Hough de deux points $p_1$ et $p_2$ alignés.....	106
Figure 4-19 : Transformée de Hough d'une ligne droite donnée. ....	107
Figure 4-20 : (a) Transformée de Hough de la lettre kaf (b) : Transformée de Hough de la lettre Ain .....	107
Figure 4-21 : (a) Transformée de Hough de la lettre I (b) : Transformée de Hough de la lettre Y .....	108
Figure 4-22 : Différentes orientations enregistrées par l'accumulateur de Hough.....	108
Figure 4-23 : Division de l'image de l'accumulateur de caractère (Kaf) en bandes verticales et zones de taille $(6 \times 6)$ .....	109
Figure 4-24 : Division de l'image de l'accumulateur de caractère yay (y) en bandes verticales et zones de taille $(6 \times 6)$ . ....	109
Figure 4-25 : (a)accumulateur de Hough ; (b) : affectation de la direction dominante à tous les pixels de la zone ;(c) :alignement des zone de haut en bas et de gauche à droite dans un vecteur linéaire ;(d) séquence d'observation obtenue. ....	111
Figure 4-26 : Exemples de caractères et les séquences d'observations générées (a)séquence représentante de la lettre arabe Kaf (b) séquence représentante de la lettre amazighe(y).....	111
Figure 4-27 : la topologie droite gauche de modèle de caractère arabe .....	113
Figure 4-28 :la topologie gauche droite de modèle de caractère amazighe.....	113
Figure 4-29 : schéma d'apprentissage .....	115
Figure 4-30 : Schéma de classification .....	115
Figure 4-31 : Exemple de modèles de quelques caractères arabes manuscrits isolés sans signes diacritiques.....	116
Figure 4-32 : Le taux de reconnaissance par rapport aux états, cas de seule gaussienne ...	122
Figure 4-33 : Le taux de reconnaissance par rapport aux états, cas de deux gaussiennes ..	122
Figure 5-1 : les graphèmes fondamentaux observés sur les caractères « $\text{ⵏ}$ », « $\text{ⵍ}$ », « $\text{ⵎ}$ », « $\text{ⵏ}$ », « $\text{ⵐ}$ », « $\text{ⵑ}$ » et « $\text{ⵒ}$ » .....	125
Figure 5-2 : Décomposition des caractères « $\text{ⵏ}$ », « $\text{ⵎ}$ » et « $\text{ⵒ}$ » en graphèmes (s :segment) ..	126
Figure 5-3 : Synopsis de système développé .....	127
Figure 5-4 : l'image du texte amazighe et sa binarisation avec la méthode d'Otsu .....	129
Figure 5-5 (a) le caractère « $\text{ⵏ}$ »; (b): binarisation par la méthode d'Otsu (seuil= 0.8196) ..	129
Figure 5-6 : Correction de l'inclinaison des lignes de texte amazighe.....	130
Figure 5-7 : Séparation de ligne du texte par la technique d'histogramme de projection horizontale .....	131
Figure 5-8 : (a) une ligne du texte, (b) Histogramme de projections verticales, (c) le résultat de la segmentation en caractères.....	132
Figure 5-9 : Résultats de l'algorithme de suppression de vide et normalisation de la taille .....	132
Figure 5-10 : Exemples de squelettisation.....	133
Figure 5-11 : Diagramme de l'algorithme sélectif proposé.....	134
Figure 5-12 : Résultat obtenus par notre algorithme .....	134
Figure 5-13 : Algorithme de composantes connexes.....	135
Figure 5-14 : Synopsis de processus de pré-classification .....	135

Figure 5-15 : (a) inflexion L , (b) croisement Y, (c) intersection T, (d) croisement X..	136
Figure 5-16: (a) points d'extrémité, (b) points d'intersection, (c) points d'inflexion.....	137
Figure 5-17: Localisation des points d'intérêts .....	137
Figure 5-18 : Illustration de l'algorithme de suivi de contour sur le caractère yav (v).....	138
Figure 5-19 : Exemples de boîtes englobantes.....	139
Figure 5-20 : Le modèle HMM gauche droite des segments .....	140
Figure 5-21 : le modèle HMM initial correspondant au segment S1 .....	141
Figure 5-22 : Schéma d'apprentissage .....	142
Figure 5-23 : Meilleur modèle HMM enregistré pour S1 .....	142
Figure 5-24 : Exemples des meilleurs chemins par le réseau de Viterbi pour les caractères .....	143
Figure 5-25: Résultats retournés par HTK sur BD1 .....	146
Figure 5-26 : Exemples de caractères da la base, de la fonte « Tassafut » .....	147
Figure 5-27 : Exemples de caractères da la base, de la fonte « Taromeit » .....	147
Figure 5-28 : Exemples de caractères da la base, dont la fonte est déférente de deux fontes Taromeit et Tassafut.....	148
Figure 5-29 : Résultats retournés par HTK sur BD2.....	149
Figure 5-30 : Architecture du système amélioré .....	150
Figure 5-31 : Détection des cercles par la transformation de Hough.....	151
Figure 5-32 : Exemples de localisation des segments par la technique de déviation maximale .....	152
Figure 5-33 : Exemples des boites englobantes .....	153

## Liste des abréviations

<b>PDA:</b>	Personal Digital Assistant
<b>PC:</b>	Personal Computer
<b>1D:</b>	Unidimensionnel
<b>2D:</b>	Bidimensionnel
<b>SVM:</b>	Support Vector Machine
<b>HMM:</b>	Hidden Markov Model
<b>MMC:</b>	Modèle de Markov caché
<b>KL:</b>	Karhunen Loève
<b>RN:</b>	Réseaux de Neurones
<b>RNA:</b>	Réseaux de Neurones Artificiels
<b>MLP:</b>	Multilayer Perceptron (Perceptron Multicouches)
<b>SOM:</b>	Self Organizing Map(Carte Organisatrice)
<b>DBN:</b>	Deep Belief NetWork (les réseaux profonds)
<b>HTK:</b>	Hidden Markov Model Toolkit (Boite à Outils des Modèles de Markov Cachés)
<b>GHMM:</b>	General Hidden Markov Model
<b>SCHMM:</b>	Semi-Continuous Hidden Markov Model
<b>EM:</b>	Expectation Maximisation
<b>ML:</b>	Maximum Likelihood (Maximum de Vraisemblance)
<b>CDHMM:</b>	Chemin Discriminant HMM
<b>DP-HMM:</b>	Discriminating Path HMM
<b>MD-HMM:</b>	Model Discriminant HMM
<b>PHMM:</b>	Pseudo ou Planar Hidden Markov Model
<b>RAE:</b>	Reconnaissance Automatique de l'Écriture
<b>RAEM:</b>	Reconnaissance Automatique de l'Écriture Manuscrite
<b>RAP:</b>	Reconnaissance Automatique de la parole
<b>ASCII :</b>	American Standard Code for Information Interchange
<b>ICDAR:</b>	International Conference on Documents Analysis and Recognition
<b>NSHP:</b>	Non-Symmetric Half-Plane
<b>NSHP-HMM:</b>	Non-Symmetric Half-Plane HMM (HMM à Demi-plan Symétrique)
<b>DCT:</b>	Discrete Cosine Transform(Transformée en Cosinus Discrète)

**FFT:** Fast Fourier Transform (Transformée de Fourier Rapide )  
**MFCC :** Mel-Frequency Cepstral Coefficients  
**MLE:** Maximum Likelihood Estimation  
**FVQ:** Fuzzy Vector Quantization (Quantification Vectorielle Floue)  
**QV:** Quantification Vectorielle  
**HMMRF:** Hidden Markov Model Random Field (Réseau de Markov Caché)  
**HPRF:** Hidden Pickard Random Field (Champ Aléatoire de Pickard Caché)  
**PPV:** Plus Proche Voisin  
**LDA:** A Linear Discriminant Analysis  
**SLLR:** Scaled Likelihood Linear Regression  
**SRTP :** Service de Recherche Technique de la Poste  
**CB:** Classifieur Bayésien  
**CPC:** Continuous Density HMM (HMM à Densité Continue)  
**CVDHMM:** Continuous Variable Duration HMM  
**DPW:** Dynamic Plane Warping (Appariement Planaire)  
**DTW:** Dynamic Time Warping (Appariement Temporal)  
**FDP:** Fonction de Densité de Probabilité  
**LBG:** Algorithme Linde, Buzo, Gray  
**LDBA:** Lexicon-Driven Level Building Algorithm  
**MAP:** Maximum A Posteriori  
**MMI:** Maximum Mutual Information (Maximum d'Information Mutuelle)  
**PCA:** Principal Component Analysis (Analyse en Composante Principale)  
**PRF:** Pickard Random Field  
**RBF:** Radial Basis Function (Fonction de Base Radiale)  
**SDNN:** Space Displacement Neural Network  
**TDNN:** Time Delay Neural Network  
**MRF:** Markov Random Fields  
**MRM:** Markov Random Mesh  
**NIST:** National Institute of Standards and Technology  
**AMHCD:** Amazigh Handwritten Character Database  
**CEDAR :** Center of Excellence for Document and Analysis Recognition  
**CENPARMI:** Center For pattern Recognition and Machine Intelligence  
**AHDB :** Arabic Handwriting Database

**IAM:** Informatics and Applied Mathematics  
**IAM-DB:** Informatics and Applied Mathematics-DataBase  
**IAM-OnDBCam:** Informatics and Applied Mathematics -Online DataBase camera  
**TUDo :** TUDortmund University  
**BBN :** Bolt, Beranek and Newman  
**TUM:** Technische Universität München  
**ETL9B :** Electro-Technical Laboratory (laboratoire d'électro-technique version 9B)  
**PS :** Point de Segmentation  
**TPar /UoB :** Telecom ParisTech/University of Balamand  
**IFN/ENIT :** Intitut fur Nachrichtentechnik/Ecole Nationale d'Ingénieurs de Tunis  
**IRF-SIC:** Reconnaissance de Formes – Systèmes Intelligents et Communicants  
**SEDAL:** System Design and Automation Laboratory at the University of Sydney  
**LOB Corpus :**The Lancaster-Oslo/Bergen Corpus





## Introduction générale

---

L'écriture dans ses différentes formes, imprimée et manuscrite a toujours été un outil essentiel dans la communication humaine, et aussi omniprésent dans la majorité des secteurs de ses activités. Elle est utilisée pour conserver et archiver le savoir. De ce fait, l'homme a toujours développé des techniques visant sa pérennité à travers les générations. En effet, avec l'apparition des nouvelles technologies d'information : l'électronique et l'informatique, et de la nouvelle augmentation de la puissance des machines, l'automatisation des traitements (la lecture, la recherche et l'archivage...) qui y sont attachés apparaît incontournable. Elle préoccupe les chercheurs de domaine de la reconnaissance de forme notamment de la reconnaissance automatique d'écriture. C'est ainsi que les recherches concernant dans ce domaine ont commencé depuis quelques décennies.

La reconnaissance automatique de l'écriture est un processus informatique complexe, qui vise à traduire un texte imprimé ou manuscrit en un texte codé numériquement, donc compréhensible par une machine, en transmettant à celle-ci la capacité de lire. La reconnaissance de l'écriture concerne plus précisément toutes les tâches en relation avec le traitement de masse des documents papiers. Par conséquent, elle porte sur les applications répétitives volumineuses avec des bases de données de taille importantes à savoir : le traitement automatique des dossiers administratifs, le tri automatique du courrier postal, la lecture des montants et des chèques bancaires, le traitement des adresses postales, le traitement des formulaires, les interfaces sans clavier, l'analyse du geste écrit, la lecture des documents patrimoniaux, l'indexation d'archives bibliothécaires et la recherche d'information dans de bases de documents.

L'automatisation de l'un de ces exemples, est un problème extrêmement difficile à mettre en œuvre, vu la grande variabilité liée aux habitudes des scripteurs ainsi aux styles et formes d'écriture (manuscrit, cursif ou imprimé avec de nombreuses fontes). En effet, l'activité de lecture qui est simple pour un humain n'est pas une tâche évidente à faire reproduire à un ordinateur. Réellement l'accomplissement de cette tâche nécessite que la machine se dote d'une base connaissance préalable de domaine et le recours à un formalisme mathématique puissant. La lecture automatique de l'écriture connaît des progrès considérables surtout dans la dernière décennie. Et cela grâce, d'une part aux nombreux travaux réalisés conduisant à une panoplie d'approches différentes et, d'autre part, à la performance des ordinateurs et des systèmes d'acquisition actuels couplées aux techniques statistiques modernes telles que les réseaux de neurones, les machines à vecteurs de support ou les modèles de Markov cachés. En outre, à la disponibilité de bases de données internationales standards relatives à l'écriture manuscrite et imprimée permettaient et permettent encore aux chercheurs de rapporter de façon crédible les performances de leurs approches dans ce domaine, avec la possibilité de les comparer avec d'autres approches.

Malgré les efforts et les progrès réalisés dans le domaine grâce aux nombreuses années d'investigation consacrées au sujet, on est encore loin du rêve d'un monde sans papier. Autrement, il n'existe toujours pas de système fiable capable de traiter l'écriture naturelle dans sa globalité. En effet, les résultats publiés dans la littérature montrent que les taux de reconnaissance obtenus sont restreints à des domaines d'application bien limités (adresses postales, chèques bancaires) ou à des catégories d'écriture très contraintes ne représentant qu'un aspect particulier de l'écriture courante et spontanée.

De ce fait, la reconnaissance automatique de l'écriture manuscrite reste encore un sujet de recherche actif. Récemment, elle fut l'objet de plusieurs recherches intenses. Le nombre de travaux se diffèrent d'un type d'écriture à un autre. Par exemple, plusieurs recherches scientifiques ont été effectuées sur le caractère latin, japonais et arabe. Par contre, le caractère Amazighe, appelé Tifinaghe, est très peu traité, On trouve quelques tentatives qui sont consacrées pour cette écriture. Ces efforts ont permis le développement et l'expérimentation de plusieurs approches de reconnaissance automatique pour chaque type d'écriture qui sont regroupées généralement en grandes classes telles que les approches statistiques, structurelles, géométriques, et syntaxiques.

Les travaux de recherche que nous avons menés s'intéressent au développement des méthodes de reconnaissance de caractères manuscrits et imprimés, de textes et de documents en tenant compte du contexte par combinaison de niveaux d'analyse et de connaissances morphologiques. Nous proposons, ainsi, deux approches de reconnaissance de l'écriture arabes et amazighes.

En effet et en premier temps, nous avons développé une approche de reconnaissance de caractères isolés indépendante de la nature du script, basée sur les primitives directionnelles obtenues à l'aide de la technique des fenêtres glissantes à partir de la transformée de Hough de caractère. L'approche conçue adopte une modélisation markovienne de type modèle discriminant qui consiste à associer un ou plusieurs modèles par classe. De ce fait, la reconnaissance est généralement s'effectue en estimant les probabilités d'émission de la suite d'observations  $\mathcal{O}$  de la forme à reconnaître par les différents modèles préalablement construits. La forme à reconnaître est affectée à la classe dont le modèle qui maximise la probabilité. Cette approche est pratiquement utilisée dans le cas où le nombre de classes à reconnaître est relativement limité (application à vocabulaire limité). Toutefois, elle devient coûteuse en temps de calcul et espace mémoire quand ce nombre dépasse le millier puisque chaque classe possède au moins un modèle qui lui est propre. Nous évaluons le système de reconnaissance proposé sur des bases de données de caractères Arabes et Amazighes. Nous montrons que l'approche proposée donne de bons résultats. Mais, elle ne tient pas en compte des caractéristiques morphologiques de l'écriture étudiée.

Dans le second temps, nous avons proposé une autre méthode pour la reconnaissance automatique hors ligne de caractères Tifinaghs imprimés. La méthode proposée est basée sur un chemin discriminant (DP-HMM) opérant sur un vocabulaire de base formé de différents graphèmes fondamentaux. Ce vocabulaire est généré en se basant sur les

caractéristiques morphologiques de la graphie Amazighe. Un seul modèle HMM globale construit et entraîné sur les éléments du vocabulaire proposé par des primitives structurelles et géométriques. Chaque chemin au long de ce treillis représente une séquence de segments, qui constitue un caractère de l'alphabet tifinagh. Pour se faire, les caractères d'entrés sont pré-classés en deux groupes (forme circulaire et non circulaire). Par la suite, ils sont décrits par leurs points d'intérêts et leurs segments. La reconnaissance s'effectue en décodant dynamiquement le chemin optimal suivant le critère de maximum de vraisemblance. Les scores obtenus montent la robustesse de l'approche proposée.

La suite du mémoire est organisé selon deux parties interdépendantes comprenant en totalité cinq chapitres : dans la première partie, nous passons en revue les principales généralités associées à la chaîne de reconnaissance de l'écriture. Un accent particulier est donné aux approches markoviennes utilisées dans ce travail. Cette partie comporte trois chapitres. La deuxième partie de ce rapport est constituée de deux chapitres qui présentent l'ensemble de nos contributions. En effet,

**Chapitre 1** présente une description générale des différents éléments et aspects concernant le processus de reconnaissance de l'écriture hors ligne. En effet, dans un premier temps, ce chapitre présente un ensemble des concepts préliminaires introduisant le domaine. En seconde temps, il décrit techniquement le système de reconnaissance de l'écriture par un schéma fonctionnel de référence à plusieurs phases.

**Chapitre 2** introduit les concepts théoriques et les outils requis pour la compréhension et l'utilisation des modèles de Markov cachés (HMMs). En effet, et comme formalisme utilisé dans nos contributions, nous y décrivons les différentes définitions nécessaires des HMMs unidimensionnels (1D) qui considèrent la forme comme un signal séquentiel, la notion des HMMs 1D planaires et les champs aléatoires de Markov qui constituent de véritables HMMs 2D.

**Chapitre 3** s'intéresse plus particulièrement aux applications des modèles de Markov cachés à la reconnaissance de l'écriture. Dans ce sens, Il comprend l'état de l'art sur la reconnaissance automatique de l'écriture par les HMMs. En effet, ce chapitre décrit les différents systèmes et approches existantes et utilisées dans ce domaine. Ces techniques seront classées selon le type de modélisation adoptée et selon la stratégie de reconnaissance. Il évoque aussi un aperçu sur les bases de données de références dans le domaine. Ces différentes bases de données sont utilisées pour évaluer et valider les performances des systèmes existants.

**Chapitre 4** présente notre première contribution qui est une approche markovienne de type modèle discriminant DM-HMM, et qui s'intéresse aux problèmes de caractères isolés. Le système développé est évalué et testé sur des images des caractères arabes et amazighes manuscrits isolés. Ce système fait recours à des primitives directionnelles dans la génération des séquences d'observations, qui sont obtenues à l'aide de la techniques des fenêtres glissantes opérant sur la transformée standard de Hough des images de caractères.

Ce chapitre, évoque premièrement les principales caractéristiques morphologiques de l'écriture arabe et amazighe suivi l'architecture de base de notre système. Dans la suite, il décrit l'étape de prétraitements effectuée sur les deux types de caractères arabes et amazighes. Ensuite, il esquisse la phase de l'extraction de primitives directionnelles à partir de l'image de caractère en utilisant la technique des fenêtres glissantes. Les séquences d'observations générées sont utilisées lors des phases de l'apprentissage et de la classification markoviennes pour entraîner les modèles des caractères et pour décider de leurs classes d'appartenances. En fin, nous présentons les expériences effectuées en montrant les résultats obtenus sur un ensemble de caractères arabes manuscrits isolés et une base de données de caractères amazighes manuscrits [ESS11].

**Chapitre 5** décrit un deuxième système de reconnaissance de caractères imprimés, basé sur une nouvelle approche qui exploite les caractéristiques et les spécificités morphologiques de la langue amazighe par une modélisation markovienne. En effet, Il présente la liste des segments de base des caractères tfinaghs déployés par notre système. Après, Il détaille les composantes du système et l'approche utilisée dans ses différentes phases. En effet, ce chapitre passe en revue les étapes de la première version du système : les prétraitements, normalisations, la segmentation du texte en caractères isolés, la pré-classification basée sur un algorithme organisé en série et l'étape d'extraction des primitives structurelles basées sur la détection des points d'intérêts de squelette. Après cela, Il présente la modélisation markovienne adoptée qui comporte les deux phases apprentissage et reconnaissance. Puis, Il donne les premiers résultats expérimentaux obtenus sur la base des patterns de la graphie amazighe [Aito09]. En suite, il propose les modules modifiés pour la version améliorée du notre système, qui comporte, en réalité, deux phases : la pré-calssification basée sur la transformation de hough, et la détection des points d'intérêts qui s'appuie sur la technique de déviation maximale. Il montre aussi leurs résultats obtenus sur la même base de données en les comparants avec ceux obtenus par sa variante antécédente. En fin, il présente la conclusion et les perspectives à envisager pour cette méthode.

Le mémoire se termine par une conclusion sur nos contributions et leurs apports à la reconnaissance de l'écriture manuscrite et des perspectives de recherche à envisager.

---

**PARTIE I :**  
ÉTUDE BIBLIOGRAPHIQUE

---

# Chapitre 1 : Généralités sur les systèmes de reconnaissance automatique d'écriture

---

## 1. Introduction

La reconnaissance automatique de l'écriture est une discipline en plein essor depuis deux décennies, Consistant en développement et la création des systèmes capables de reconnaître toutes sortes écritures dans ses formes manuscrites et imprimées. Grâce à la puissance considérable des performances des ordinateurs actuels, ce domaine connaît encore une évolution extrêmement rapide. Plusieurs approches et méthodes quasi standards de reconnaissance de l'écriture ont été également établies et perfectionnées, notamment pour les écritures latines et arabes. Cependant, la recherche est encore très active dans le cas des documents manuscrits dégradés et historiques qui se caractérisent par une grande variabilité inhérente à la nature de l'écriture manuscrite.

Nous présentons dans ce chapitre, une description générale des différents éléments et aspects concernant le processus de reconnaissance de l'écriture. En effet, dans un premier temps, nous présentons un ensemble des concepts préliminaires introduisant le domaine. En seconde temps, Nous décrivons techniquement le système de reconnaissance de l'écriture par un schéma fonctionnel de référence à plusieurs phases. Ensuite, nous esquissons et nous discutons les différentes approches et techniques proposées et utilisées dans la littérature au niveau de chaque étape.

Cette partie s'appuie en particulier sur les différentes techniques utilisées dans le cadre de la reconnaissance de l'écriture latine et arabe. Un état de l'art plus spécifique à l'utilisation des modèles de Markov caché dans la reconnaissance de l'écriture manuscrite sera donné dans le chapitre 3.

Enfin, nous terminerons ce survol par les différents secteurs d'application de la reconnaissance de l'écriture.

## 2. Préliminaires de la reconnaissance d'écriture

La reconnaissance automatique de l'écriture est un domaine variée qui s'intéresse à l'élaboration de systèmes capables de reconnaître des textes manuscrits ou imprimés. Ces systèmes de reconnaissance dépendent de plusieurs facteurs, à savoir, le nombre de scripteurs, le type ou le style d'écriture, la taille et le type de vocabulaire. Ces facteurs constituent des paramètres fondamentaux aussi bien pour leur mise en pratique que pour leur complexité de conception, ils se différencient aussi selon différents critères, par exemple leur mode de reconnaissance, c'est-à-dire par la stratégie suivie pour la localisation des unités de base de l'écriture dans les mots et de la nature des informations disponibles

pour la reconnaissance. Nous présentons dans ce qui suit une revue des principaux facteurs et critères pilotant un système de l'écriture.

## 2.1. Types et styles d'écritures

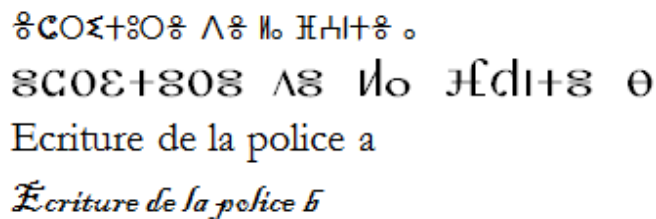
Le style d'écriture représente le type de la fonte ou la police utilisée pour l'imprimé. Cependant, il s'agit de la façon avec laquelle chaque scripteur, produit le signal d'écriture qui lui est propre dans le cas de manuscrit. En effet, son écriture possède des caractéristiques intrinsèques qui sont différentes à celles d'autres scripteurs même si l'alphabet de base est le même. Cette variabilité de l'écriture dans ses différentes formes constitue une contrainte cruciale dans la reconnaissance.

La variété de l'écriture illustrée par Figure 1-1 ci-dessous est due à des contraintes externes et internes.

Contraintes externes : représentent le type d'écriture. Dans certains cas, le texte peut être pré-casé, zoné, guidé ou libre.

Contraintes internes : elles sont les habitudes de chaque scripteur (les spécificités propres au scripteur).

L'écriture peut être scripte c.-à-d. une écriture à lettres séparées, cursive dont les lettres groupées sont liées, ou mixte qui constitue un compromis entre les deux types précédents. Dans [Tapp84]Tappert propose une bonne classification des styles d'écriture du manuscrit latin. Cette classification permet de révéler la complexité de la tâche de reconnaissance en fonction des différentes contraintes.


  
 Ecriture de la police a
   
 Ecriture de la police b

(a) Forme imprimée

Ecriture scripte espacée
   
 Ecriture scripte : lettres détachées
   
 Ecriture cursive pure : lettres liées
   
 Ecriture mixte ou naturelle : lettres liées ou non

(b) Forme manuscrite [Pois05].

Figure 1-1 : Catégories et formes de l'écriture

## 2.2. Scripteurs et vocabulaires

Dans un système de reconnaissance de l'écriture (RAE), lorsque le nombre de scripteurs augmente, la reconnaissance devienne de plus en plus difficile. C.-à-d., plus il existe plusieurs variétés de scripteurs (droitier/gaucher, sexe, différents âges, différentes professions), plus il y a beaucoup de styles d'écritures différents. Les Figure 1-2 et Figure 1-3 ci-dessous montrent quelques écritures selon différents scripteurs. Généralement, on trouve différents systèmes selon ce critère :

- Système mono-scripteur : système destiné à apprendre et reconnaître l'écriture d'un seul utilisateur.
- Système multi-scripteur : système de niveau plus général que le précédent. Il permet d'apprendre et de reconnaître l'écriture d'un nombre de scripteurs bien déterminés. Dans ce cas, chaque classe est représentée par un modèle qui absorbe les variabilités liées aux habitudes de chaque scripteur.



Figure 1-2 : Le mot « Rome » écrit par différents scripteurs [Lema07]

Dans l'attente de votre réponse et en vous  
remerciant par avance, je vous prie d'agréer,  
Madame, Monsieur, l'assurance de toute ma considération.

Je vous prie d'agréer, Monsieur, l'expression  
de mes sentiments distingués.

Je vous prie, Madame, Monsieur, en l'expression de  
mes vœux les plus respectueux.

Je vous prie de croire en  
mon souvenir le meilleur,

Figure 1-3 : variabilités d'écritures selon les scripteurs

- Système omni-scripteur : système aussi bien général et naturel que les précédents. Mais, Il est caractérisé par un degré de complexité maximale. Son but est d'apprendre et de reconnaître l'écriture de n'importe quel scripteur. Ce type d'application exige un



corpus d'apprentissage très important et diversifié, afin d'aboutir à un modèle capable d'absorber cette variabilité abondante des styles d'écriture.

Comme nous l'avons déjà signalé, le type et la taille du vocabulaire sont également des facteurs influant directement sur les performances des systèmes de reconnaissance. De ce fait, on trouve de nombreuses approches de reconnaissance en fonction de la taille du lexique (étendue, restreinte) et de son type (statique ou dynamique). Essentiellement, on distingue entre quatre systèmes principaux ci-dessous :

- Systèmes à vocabulaire petit est constitué d'une dizaine de mots. Dans ce cas, la reconnaissance des mots peut s'effectuer en utilisant des méthodes globales (cf. section 2.6.1) ;
- Systèmes à vocabulaire de taille moyenne comporte une centaine de mots ;
- Systèmes à vocabulaire grand rassemble des milliers de mots ;
- Systèmes à vocabulaire très étendu renvoie à plus d'une dizaine de milliers de mots. Dans ce cas, les seules méthodes efficaces envisageables pour la reconnaissance sont les approches analytiques (cf. section 2.6.2).

Dans le cadre applicatif pour chacun des systèmes précédents, l'encombrement mémoire et le temps d'exécution est proportionnelle à la taille de vocabulaire.

Concernant le type du vocabulaire, un vocabulaire est dit statique lorsque son contenu n'est pas modifiable durant la phase d'exploitation du système. En revanche, un vocabulaire est dit dynamique si son contenu peut être enrichi par le biais d'exploitation.

### 2.3. Évolution de la difficulté du traitement de l'écriture

Le schéma synthétique de la Figure 1-4 ci-dessous résume les degrés de difficultés du traitement de l'écriture en fonction de contraintes externes : type d'écriture, nombre de scripteurs et taille du vocabulaire.

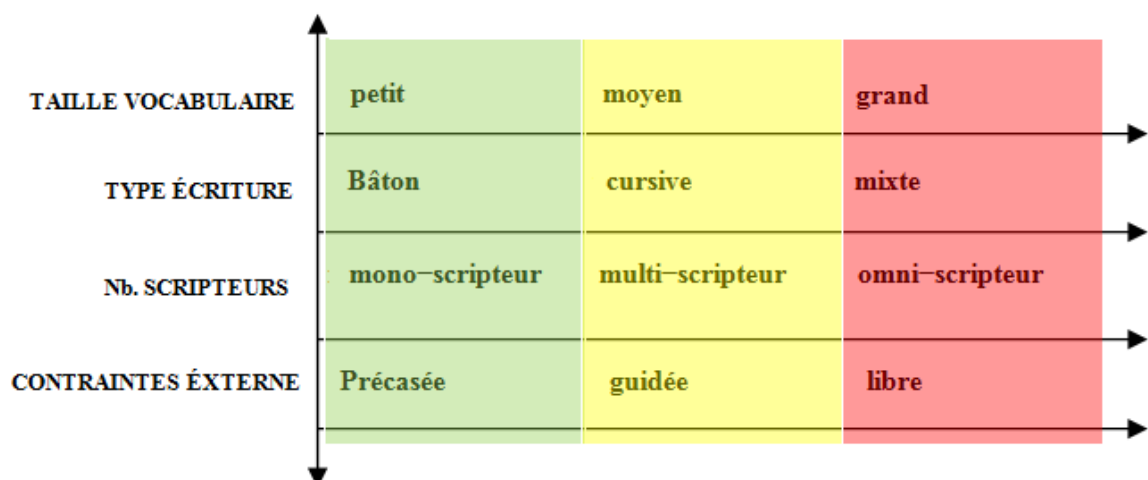


Figure 1-4 : Difficultés de l'écriture

## 2.4. Systèmes en ligne

Dans le cas des systèmes en ligne, dits aussi dynamiques, l'écriture est saisie naturellement à l'aide d'un stylet sur une ardoise ou un écran ou par un stylo digital. Par conséquent, la reconnaissance de l'écriture est réalisée pendant le processus de la saisie. Ces systèmes sont utilisés dans plusieurs équipements électroniques : Smartphone, iPhone, iPad, PDA, ou Tablet PC comme les montrent la Figure 1-5 ci-dessous. Ce type de reconnaissance présente pas mal d'avantages. Par exemple, l'absence de bruit puisque l'écriture se fait sur une table spéciale, une écriture avec 1 pixel d'épaisseur de trait donc tracé squelettique, la disponibilité des informations temporelles, telle que la vitesse, l'accélération, la pression et le levé du stylo et les données se présentent alors sous la forme d'une séquence de points ordonnés dans le temps. Dans ce cas, le signal est de type 1D et le système de reconnaissance peut bénéficier de la représentation temporelle. D'où l'exploitation et l'application de certaines approches qui ont été développées pour la reconnaissance de la parole dans ce domaine. La plupart des systèmes en ligne sont analytiques et se basent sur des techniques telles que : l'appariement élastique (elastic matching) [Bahl01] les k-plus proches voisins [Schw96] les réseaux de neurones et les Support Vecteur Machines (SVM) pour la classification des lettres isolées et des approches hybrides combinant les HMMs avec des réseaux de neurones pour le cas de la reconnaissance de mots ou phrases cursives. Parmi les travaux qui ont été menés dans ce domaine on trouve : plusieurs techniques à base de réseaux de neurones qui sont proposées dans ([Alim97], [Mezg02], [Mezg03], [Miti02]), d'autres, qui traitent la reconnaissance de caractères isolés [Bena00], et qui exploitent des informations temporelles dans la construction des caractères [Beng94]. Plus récemment dans [Amin80], l'auteur propose une méthode syntaxique (appariement des chaînes) des catégories. Une autre approche basée sur l'arbre et le code de Freeman est proposée dans [Al-Em90]. Finalement, un bon état de l'art des principaux systèmes avec l'analyse des différentes techniques de reconnaissance de l'écriture en ligne est présenté dans ([Tapp90], [Plam00]).



Figure 1-5 : Exemples des systèmes en-ligne

Les outils de la Figure 1-5 ci-dessus sont des micro-ordinateurs munis des logiciels commercialisés de reconnaissance de l'écriture en ligne et un matériel d'acquisition beaucoup plus efficace permettant la saisie de mots voire phrase et texte [Pois05].

Aujourd'hui d'autres applications se sont développées à partir de ces assistants personnels liées au domaine de la médecine, management, marketing, l'éducation, l'industrie et la gestion.

## 2.5. Systèmes hors ligne

Dans cette seconde catégorie, les systèmes sont dits aussi statiques. L'écriture qui a été écrite sur un support en papier préalablement est par la suite numérisée par un scanner ou une caméra, qui permet de convertir l'écriture en image numérique. L'information est alors présentée sous l'aspect bidimensionnel. Par rapport à la reconnaissance en ligne précitée, les systèmes hors ligne ne disposent plus de l'information temporelle et dynamique du tracé, de plus l'épaisseur du tracé devient une contrainte supplémentaire à prendre en compte. Par conséquent, ces systèmes disposent seulement des images des caractères ou des mots qui sont indépendantes de l'ordre de leur génération. Ce qui rend ce cas plus difficile. En effet, ces systèmes nécessitent davantage une étape des prétraitements : Seuillage, squelettisation, élimination de bruit, normalisation et segmentation qui peut être beaucoup plus complexe, surtout dans le cas de l'écriture cursive. Cependant, l'utilisation large de ces systèmes nécessite un traitement rapide avec un taux de reconnaissance élevé. Mais généralement, leurs performances sont souvent moins meilleures que celles d'en ligne pour les mêmes conditions d'expérimentations (taille du vocabulaire, nombre de scripteurs, style d'écriture, etc.). En revanche, ils sont très utilisés notamment dans la lecture automatique des montants de chèques bancaires ([Gors99], [Augu00], [Dile00], [Kauf00], [Guil98]), la lecture d'adresses et le tri automatique du courrier [Kim98], la reconnaissance automatique de contenu de formulaires pré-casés [Héro98], la reconnaissance et l'indexation des collections patrimoniales, des archives et des documents anciens: ([Lebo07], [Nico06]), la lecture des images issues des tableaux blancs [Liw05] et la recherche d'information dans une base de documents manuscrits [Bens04]. Nous y reviendrons plus largement dans la section 5. D'ailleurs, notre étude portera uniquement et plus particulièrement sur les systèmes hors ligne, cadre applicatif de cette thèse.

## 2.6. Stratégies de reconnaissance

Les systèmes de reconnaissance hors ligne se distinguent selon différents aspects, par exemple le type de primitives choisi ou le moteur de reconnaissance utilisé ou encore par leur mode de reconnaissance, c'est-à-dire par la stratégie suivie pour la localisation des unités de base de l'écriture dans les mots. Le choix entre ces différentes stratégies est fortement dépendant de la complexité de l'application de reconnaissance d'écriture. Notons qu'une méthode qui peut fonctionner efficacement pour une tâche donnée, peut s'échouer si on l'applique directement sur un autre problème de nature différente. Par exemple, les difficultés qui se posent dans une reconnaissance à grand vocabulaire (étendu) monoscripteur ne sont pas les mêmes que celles dans une reconnaissance omni scripteur dans un vocabulaire très limité. Dans le contexte de la reconnaissance de l'écriture, Il existe deux types de stratégies globale et locale selon la nature et la taille de l'unité considérée dans la modélisation mise en œuvre.

### 2.6.1. Stratégie globale ou holistique

La stratégie globale ([Srig01], [Khor03], [Al-Ma04], [Al-Ra06], [Soui06], [Al-Ma06], [Fara06], [Shah06]), consiste à modéliser les mots comme des entités de bases non divisibles. Elles utilisent des descripteurs globaux sur le mot complet, comme les boucles, ascendants, descendants, profils haut/bas, vallées, longueur, points terminaux, points de croisements et bien d'autres, en évitant le processus de segmentation et ses problèmes. Cette approche est efficace et simple, car elle ne nécessite pas de reconnaissance au niveau des caractères ou des graphèmes qui composent le mot. Donc chaque mot est modélisé par un modèle spécifique, pour une telle application, il faudra générer autant de modèles que de mots présents dans le vocabulaire.

En revanche, le défaut de cette approche est qu'elle n'est applicable que sur de petits vocabulaires. Au-delà de quelques dizaines de classes, comme dans le cas des applications de traitement et de vérification des chèques bancaires. De ce fait, la disposition d'un modèle pour chaque mot (classe) dans le cas d'un grand dictionnaire, rend cette approche plus coûteuse et non fiable, spécialement dans le cas d'un dictionnaire contenant des mots similaires. La capacité discriminante des primitives extraites globalement diminue. Les confusions possibles entre mots du vocabulaire augmentent, ce qui dégrade les performances. En pratique, les approches holistiques se limitent souvent à des vocabulaires de quelques dizaines de classes de mots au maximum.

### 2.6.2. Stratégie locale ou analytique

Contrairement à l'approche globale où la reconnaissance est réalisée par la représentation entière de mot, l'approche analytique ([Pech03], [Choi03], [El-Ha05], [Beno07]) décompose le mot en une séquence de caractères ou de graphèmes intermédiaires, qui font partie d'un alphabet prédéfini. La reconnaissance du mot complet sera obtenue par la combinaison des reconnaissances de ces caractères intermédiaires. Il est donc nécessaire de découper le mot à reconnaître en une séquence de symboles. Ce qui n'est pas toujours trivial. Une étape de segmentation est donc nécessaire pour déterminer les limites entre les entités. Or, cette tâche est particulièrement délicate (dilemme de Sayre section 3.4) du fait de l'absence de segmentation idéale : les limites entre caractères sont parfois difficiles à déterminer même pour un être humain. Par conséquent, le processus de reconnaissance selon cette approche doit nécessairement se concevoir comme un processus alternatif entre les phases de segmentation et identification des segments. L'avantage de cette approche est qu'elle convienne pour des applications à vocabulaire ouverts ou de grande taille, tel que le tri automatique du courrier nécessitant l'identification de nombreux noms de villes.

Dans cette approche, chaque lettre est modélisée par un modèle spécifique et la modélisation d'un mot est faite par la concaténation des modèles des lettres constituantes.

Finalement le choix de l'approche globale ou analytique est généralement dicté par la taille du vocabulaire. Dans le cas de petits lexiques, il est envisageable de créer autant de modèles que de mots possibles. Par contre pour un vocabulaire de plusieurs milliers de mots, il n'est pas concevable de créer un modèle pour chaque classe ; il est plus pertinent de

modéliser au niveau d'entités plus petites comme les caractères. Dans ce cas, le système doit segmenter l'écriture en entités de base.

### 3. Système de reconnaissance de l'écriture hors ligne

Dans cette section, nous intéressons essentiellement aux systèmes de la reconnaissance automatique de l'écriture hors ligne. C'est le type qui va nous intéresser dans nos contributions. En effet, pour mieux cerner notre étude, nous allons les introduire par un modèle général à plusieurs étapes, qui reflète globalement le processus de reconnaissance de l'écriture. Les différentes techniques et méthodes développées et utilisées au niveau de chaque étape de ce modèle sont évoquées et analysées.

#### 3.1. Architecture

La tâche essentielle de système de reconnaissance d'écriture hors-ligne réside dans la transformation de signal d'écriture sous ses différentes formes, imprimée ou manuscrite en représentation symbolique, en tenant compte de ses différentes natures spatiales. Pour se faire, il nécessite un certain nombre d'étapes à mettre en œuvre. L'ensemble de ces phases forment généralement la structure de système de reconnaissance d'écriture, qui peut se résumer par le schéma de la Figure 1-6 ci-dessous.

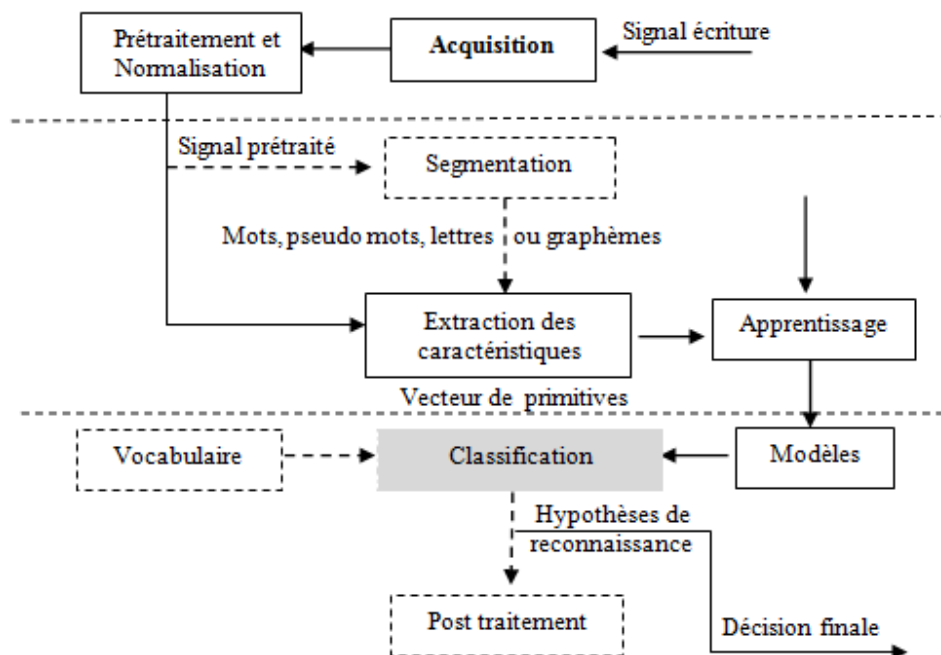


Figure 1-6 : Diagramme de référence d'un système de reconnaissance

La phase de l'acquisition a pour but la numérisation de l'entrée par un scanner.

La phase des prétraitements et normalisation a pour but la réduction de bruit et les variabilités de l'écriture pour n'en garder que les informations utiles.

La phase d'extraction des primitives permet de faire à partir de la représentation de l'image une description synthétique de la forme à reconnaître dans un espace à plusieurs dimensions.

L'étape de segmentation elle permet de décomposer l'image d'un texte en entités (mots, caractères ou graphèmes) afin de réduire la complexité des modules de traitements ultérieures.

La phase apprentissage consiste à trouver les modèles les plus adaptées aux entrées du problème. Le résultat est une base de données d'apprentissage qui constitue la base de référence de système.

Quant à l'étape de la classification du caractère ou du mot, la description du caractère à reconnaître issue de la base de test est comparée aux descriptions des caractères de la base de référence.

En plus de ces phases, on pourra trouver une phase de post-traitement qu'est une opération facultative et dont le rôle consiste à vérifier et éventuellement à corriger l'hypothèse de reconnaissance générée par le module de classification en utilisant des informations contextuelles telles que : lexicales, syntaxiques, sémantiques, pragmatiques, etc.

Par la suite, nous passerons en revue des techniques de mise en œuvre de chaque phase du système.

### **3.2. Acquisition**

Dans les systèmes hors lignes, l'acquisition est une phase où l'écriture est digitalisée à l'aide d'un scanner. Le résultat de cette phase représente l'entrée de système. Il dépend de deux paramètres importants *Résolution* et *Niveau d'éclairage*. L'acquisition est assez simple mais très importante car elle influence sérieusement les étapes suivantes.

### **3.3. Prétraitements et normalisation**

Dans cette section, on s'intéresse aux méthodes de prétraitements les plus couramment utilisées dans les systèmes de reconnaissance d'écriture. Ces applications admettent comme l'entrée des images digitalisées et acquises par des outils spécialisés. De telles images peuvent être en blanc et noire ou en niveaux de gris, qui peuvent subir des éventuels perturbations qui sont globalement liées au contexte de l'acquisition (éclairage ou bougé), ou au bruit lié au capteur (mauvaise utilisation, mauvaise qualité), ou aux parasites liés à la nature de la scène (nuage, poussière) et finalement aux effets de quantification et d'échantillonnage.

Le but est de pallier, ou d'atténuer sinon de supprimer les empâtements (informations indésirables ou des structures inutiles du point de vue de la question traitée) qui sont liés aux différentes sources de bruit citées préalablement, et cela pour préparer l'image aux étapes ultérieures. Des processus de bas-niveaux sont relatés, qui nécessitent très peu d'informations sur le contenu des images. Il s'agit ici des processus de filtrage, seuillage et squelettisation. Ainsi dans le but de réduire la variabilité et la diversité des styles des scripteurs, nous présenterons quelques techniques de normalisation, comme la correction de l'inclinaison des lignes (correction du skew), la correction de l'inclinaison des lettres au sein d'un mot (correction du slant) et la détection de la bande de base qui permet d'effectuer une normalisation en hauteur de l'écriture ([Bela92], [Meng00], [Lamo96], [Lins94]).

### 3.3.1. Filtrage

Le problème de l'amélioration est quant à lui lié à la perception humaine. C'est donc un problème subjectif. Le système visuel humain étant particulièrement sensible aux forts contrastes. Les techniques d'amélioration tentent d'augmenter ceux-ci pour accroître la séparabilité des régions composant une scène. Il existe deux grandes familles de méthodes [Korn98]: les méthodes globales et les méthodes locales. Les méthodes globales permettent de modifier chaque point de l'image en fonction d'une information globale sur les niveaux de gris de l'image. Parmi ces méthodes on trouve celles basées sur la modification de l'échelle des niveaux (contrastage, négatif, extraction de bits, troncature, seuillage) et d'autres basées sur la modification de l'histogramme. Par exemple, la Figure 1-7 ci-dessous illustre le principe de la méthode égalisation d'histogramme.

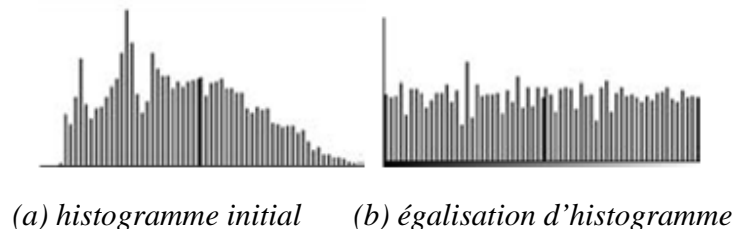


Figure 1-7 : Exemple d'égalisation d'histogramme

Les méthodes locales ou aussi dites de voisinage quant à elles, plutôt que de travailler sur des informations globales, ces méthodes travaillent localement sur des pixels ou sur un voisinage de pixels (cf. l'équation (1.1) ci-dessous).

Les techniques employées sont de type filtrage avec des approches de type fréquentielles et son corollaire dans l'espace pixel : le filtrage linéaire (filtre moyennneur). Le traitement d'images a également développé d'autres méthodes spécifiques de filtrage non linéaire dont le filtre médian et *Min/Max* sont des cas particuliers.

$$I'(i, j) = \sum_{m, n \in V} h(m, n) I(i-m, j-n), \quad (1.1)$$

$$\sum_{m, n \in V} h(m, n) = 1,$$

Où  $I$  est l'intensité de l'image d'origine,  $I'$  est l'intensité de l'image filtrée,  $V$  est le voisinage utilisé et  $h$  est le masque de convolution.

Dans ce cas, l'opérateur  $h$  pourra être un filtre moyenneur simple sur un voisinage  $3 \times 3$  (cf. Tableau 1-1 ci-dessous). Ainsi, chaque pixel est remplacé par la moyenne pondérée de ses voisins.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Tableau 1-1 : Masque filtre moyenneur  $3 \times 3$

Le filtrage non linéaire *Min/Max* consiste à remplacer la valeur du pixel central  $p$  par le *Min* ou le *Max* sur la fenêtre selon la proximité de ces deux valeurs à la valeur du pixel central (cf. l'équation (1.2)).

$$f_w(x, y) = \min (\text{resp. max}) \text{ si } f(x, y) < \frac{\max - \min}{2} \text{ (resp. } f(x, y) > \frac{\max - \min}{2}) \quad (1.2)$$

### 3.3.2. Seuillage

Les systèmes de reconnaissance nécessitent une étape de binarisation qui vise à séparer les pixels de texte des pixels de l'arrière-plan de l'image traitée. En fait, la plupart des systèmes ne fonctionnent que sur des images binaires. La plus simple façon pour obtenir une image binaire est de choisir une valeur seuil, puis de classer tous les pixels dont les valeurs sont au-dessus de ce seuil comme étant des pixels d'arrière plan, et tous les autres pixels comme étant des pixels de texte.

Soit l'image  $I(M \times N)$ , supposons que  $f(x, y)$  représente le niveau de gris du pixel aux coordonnées  $(x, y)$ ,  $0 \leq x \leq M$ ,  $0 \leq y \leq N$  et  $s$  est le seuil choisi, les pixels de l'objet sont ceux ayant le niveau de gris inférieur à  $s$  et les autres ayant le niveau de gris supérieur à  $s$  sont des pixels du fond. Alors, l'image binarisée  $g$  est déterminée par les pixels  $(x, y)$  dont la valeur est donnée par l'équation (1.3) :

$$g(x, y) = \begin{cases} 1 & \text{si } f(x, y) > s \\ 0 & \text{si } f(x, y) \leq s \end{cases} \quad (1.3)$$

Le problème alors est de savoir comment sélectionner le bon seuil. La plupart des méthodes de binarisation des images de texte se basent sur des méthodes de seuillages globales ou locales qui seront discutées dans la section suivante. Ces seuils sont déterminés en fonction



de quelques statistiques concernant les distributions de la luminance ou de la chrominance généralement basées sur des histogrammes, sans tenir compte de la forme des images.

### 3.3.2.1. Seuillage global

La méthode de seuillage globale consiste à calculer un seuil unique à partir d'une mesure globale sur toute l'image. Il nous permet de décider de l'appartenance d'un pixel à l'objet ou au fond sur toute l'image. Cette méthode convient pour les documents simples et de bonne qualité. Néanmoins, elle n'est plus applicable lorsque la qualité d'impression du texte n'est pas constante dans toute la page et également si le fond est bruité ou non homogène, dans ce cas des taches parasites peuvent apparaître. Les méthodes d'Otsu [Otsu78], de Kapur [Kapu85], de Pun [Pun80], ou de Cheng et Chen [Chen98] peuvent être tenues comme des représentants de cette approche. Chacun a de différentes stratégies pour atteindre leur but. Par exemple, la méthode décrite dans [Otsu78] calcul un seuil optimal qui sépare deux classes (fond et arrière-plan) afin que leur variance intra classe soit minimale, tandis que d'autres méthodes dans ([Kapu85], [Pun80], [Chen98]) se basent sur la théorie de maximum d'entropie ou d'entropie floue. Cependant, la solution fournie par ces techniques n'est pas toujours efficace. La Figure 1-8 ci-dessous montre le résultat insuffisant retourné par un seuil global, dans le cas d'une mauvaise illumination du document, ou dans le cas où le texte passerait de noir sur fond blanc à blanc sur fond noir. Pour pallier à ces problèmes, il fallut trouver des techniques permettant d'adapter localement le niveau du seuil.

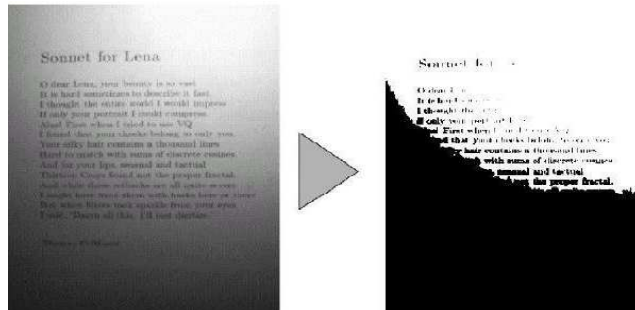


Figure 1-8 : Problème de seuillage global

### 3.3.2.2. Seuillage local

Le principe du seuillage local est d'adopter une étude localisée autour du pixel pour déterminer quel seuil utiliser. Pour réaliser cette étude locale, les techniques utilisent une fenêtre centrée sur le pixel à étudier (voisinage de pixel). Cette fenêtre peut avoir différentes tailles, souvent en fonction de la taille moyenne du texte dans le document. En 1986, Bernsen a proposé pour la première fois une technique [Bern86] donnant de bons résultats. Il calcul le seuil par la formule (1.4) suivante :

$$S(i, j) = \frac{(Min(i, j) + Max(i, j))}{2} \quad (1.4)$$

Avec :

$S(i, j)$  : seuil à appliquer pour le point  $i, j$ ;

$Max(i, j)$ ,  $Min(i, j)$  : valeur du niveau de gris maximal (resp. minimal) dans une fenêtre centrée en  $(i, j)$  de taille  $N \times M$ .

Cependant, ce filtre est très sensible au bruit du fond. À cause de la prise en compte du maximum et du minimum uniquement, dans le cas où la fenêtre est uniquement sur le fond, le bruit sera interprété comme objet, car le seuil sera bas. La même année, Niblack proposa une méthode similaire sur le principe, mais prenant en compte d'autres paramètres [Nibl00]. Dans sa méthode, le seuil est calculé ainsi par l'équation (1.5) ci-dessous :

$$S(i, j) = \mu(i, j) + k \times \sigma(i, j) \quad (1.5)$$

$S(i, j)$  : seuil à appliquer pour le point  $i, j$ ;

$\sigma(i, j)$  : valeur de l'écart type dans une fenêtre centrée en  $i, j$  de taille  $N \times M$ ;

$\mu(i, j)$  : valeur moyenne des niveaux de gris dans la même fenêtre ;

$k$  : constante fixée le plus généralement à 0,2;

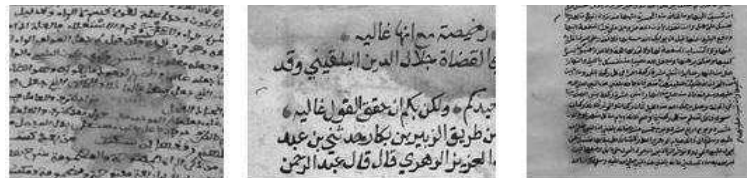
Sauvola, en 2000, proposa une amélioration de la méthode de Niblack afin de réduire sa sensibilité au bruit sur le fond [Sauv00]. Le seuil est alors calculé ainsi par la formule (1.6):

$$S(i, j) = \mu(i, j) + k \times \left( \frac{\sigma(i, j)}{R} - 1 \right) \quad (1.6)$$

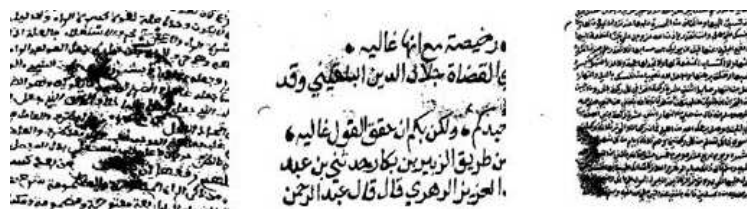
avec les mêmes notations que pour la méthode de Niblack et  $R$  étant une constante permettant d'ajuster la dynamique de l'écart type (fixé en général à 128). Le gain apporté par cette pondération permet à la méthode d'être très satisfaisante dans le cas de documents mal éclairés et fait passer de 89% à 95% le taux moyen de reconnaissance par un logiciel de reconnaissance de caractères (sur une base de 15 documents éclairés non uniformément).

Les Figure 1-9 et Figure 1-10 ci-dessous présentent les résultats obtenus sur des images de manuscrits arabes par quelques méthodes de seuillage :

Originales images



Absolute S=127



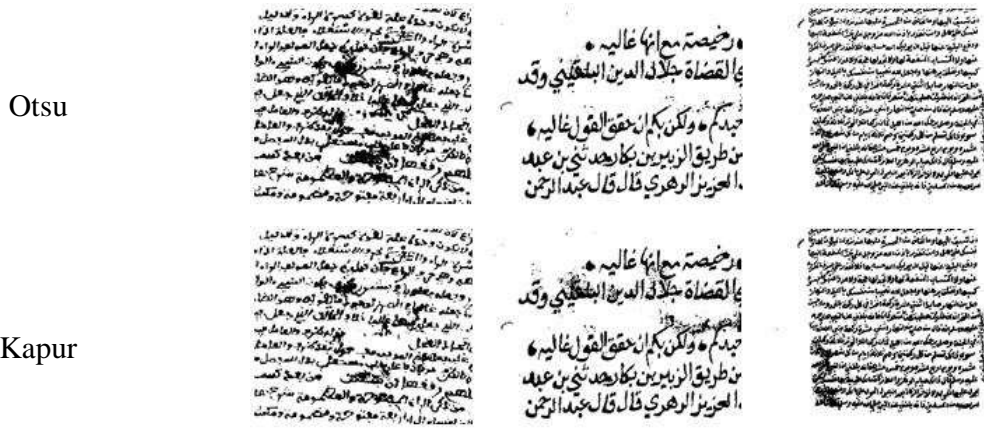


Figure 1-9 : résultats de seuillage globales sur les manuscrits arabes [Kefa10]



Figure 1-10 : résultats de seuillage locales sur les manuscrits arabes [Kefa10].

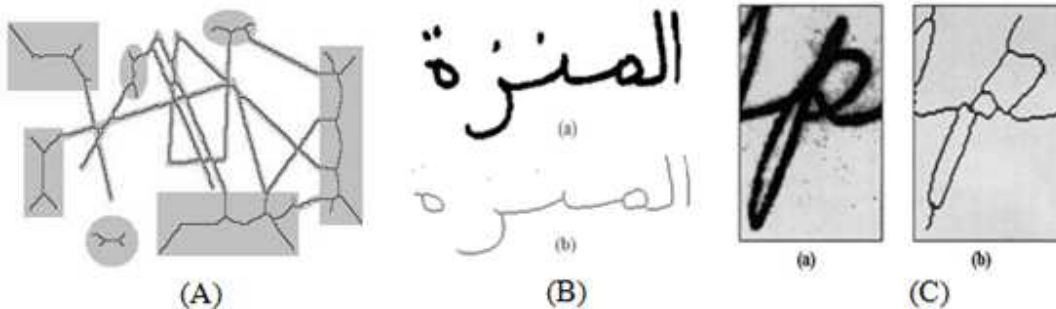
### 3.3.3. Squelettisation

Le squelette est une représentation filiforme de l'écriture, d'épaisseur un pixel. C'est une manière de représenter l'information indépendamment de l'épaisseur initiale de l'écriture. Il permet d'extraire des caractéristiques importantes, comme les intersections, le nombre de tracés et leurs positions relatives. La représentation en squelette doit conserver les propriétés topologiques du tracé d'écriture (nombre de parties, nombre de trous, connexité) et ses propriétés métriques (longueur totale, distance entre parties de la forme), faciliter

l'expression des relations structurelles dans la forme et réduire le temps de traitement en simplifiant la structure de données.

La squelettisation d'écriture peut être utilisée à des fins de segmentation ([Khor00], [Khor03], [Srih06a]), normalisation ([Pech03], [Pech06], [Märg06]); et d'extraction de primitives ([Al-Ma02a], [Al-Ma02b], [Al-Oh02], [Al-Ra06]). Généralement, Il existe de nombreux travaux de recherche sur la squelettisation qui sont regroupés en deux types d'algorithmes [Zhon99]: les algorithmes séquentiels dont les pixels sont parcourus les uns à la suite des autres pour effectuer la squelettisation (eg. Méthode par érosion); et les algorithmes parallèles [Zhan84], où le calcul effectué en un pixel est indépendant de celui effectué sur les autres pixels, ce qui permet d'obtenir le squelette (eg. Le calcul de l'axe médian).

La Figure 1-11 ci-dessous illustre des exemples de squelettisation. Après le calcul de la squelettisation, des distorsions rendent difficile le choix de la trajectoire au niveau des intersections, et dans le but d'améliorer le squelette, D.X. Zhong et al proposent une méthode dans [Zhon99] qui corrige les intersections entre deux segments de droites qui se retrouvent parfois scindés, comme le montre la Figure 1-11-C ci-dessous.



(A) : Squelettisation d'une forme quelconque [Dupr03].

(B) : (a) image binaire du mot  
(b) squelette du (a) [Beno07].

(C) : distorsions provoqués par squelettisation (a) image brute (b) squelette déduit [Doer95].

Figure 1-11 : Exemples de squelettisation

Ainsi un grand nombre de petits segments du squelette peuvent s'avérer non pertinents, comme le montre la Figure 1-11-A ci-dessus. Ces petits traits, appelés barbules, peuvent être nettoyés pour rendre le squelette plus lisible. Plusieurs critères ont été proposés pour supprimer ces traits superflus. Par exemple, on peut utiliser un simple critère de taille : tous les arcs de squelette dont la longueur est inférieure à un seuil sont considérés comme du bruit et sont supprimés [Jang92].

Certains algorithmes de squelettisation sont sensibles au bruit et aux variations intrinsèques de certains types d'écriture latine, chinoise, arabe et tiffinaghe. De ce fait, l'application directe sans adaptation d'un algorithme conçue pour telle ou telle graphie sur une autre écriture ne permet pas d'obtenir un squelette adapté. Par conséquent, elle génère des squelettes qui sont contaminés par des barbules superflues (i.e. faux traits) qui peuvent

faiblir le taux de reconnaissance. Dans la plus part des cas, il est difficile de mesurer la performance des algorithmes de squelettisation en tant que tel. Des mauvaises estimations de résultat détériorent les performances du moteur de reconnaissance. Par conséquent, il faudrait tester l'application complète pour estimer quel est l'algorithme le plus adapté.

### 3.3.4. Normalisation

La normalisation tend à réduire ou d'éliminer autant que possible les variabilités liées aux styles, tailles et orientations d'écriture pour rendre celle-ci la plus indépendante possible du scripteur. L'importance de cette phase est de faire en sorte que l'étape d'extraction de primitives soit la moins influencée par la variation des styles d'écriture.

Les traitements de normalisation présentés ici sont :

- Normalisation de la taille ;
- Correction de l'inclinaison des lignes : (Skew correction) ;
- Correction de l'inclinaison des caractères :( Slant correction) ;
- Estimation de la ligne de base.

Une normalisation de différentes variations rencontrées avec différentes techniques est présentée dans ([Saon97] [Fuji07]).

#### 3.3.4.1. Normalisation de la taille

La taille des caractères peut varier d'un scripteur à l'autre et même au sein de l'écriture d'un seul scripteur ou après agrandissement ou réduction. La Figure 1-12 ci-dessous présente une image de mot avec différentes tailles. Ce qui peut causer une instabilité des paramètres lors de la phase de l'extraction de primitives. De nombreux travaux ont été proposés dans la littérature pour la normalisation de la taille de l'écriture latine et arabe, Cependant aucuns d'entre eux n'est pas acceptés ou considérés comme une méthode standard. La robustesse de chaque technique dépend de type de graphie. Une méthode plus ou moins robuste pour l'écriture latine est présentée dans [Madh99]. Cette technique est basée sur l'estimation de la moyenne des largeurs des caractères. Par la suite, nous présentons le principe de l'algorithme de Srihari [Srih97] qui consiste à ramener les caractères à la même taille.

Cet algorithme opère en deux étapes. La première normalise le caractère en hauteur et la seconde en largeur. L'ordre de normalisation ainsi choisi évite que les caractères fins ne se déforment par rapport à des caractères épais.

Il s'agit de transformer l'image du caractère de dimension  $h_i, l_i$  en une image de dimension  $h, l$ . Une étape intermédiaire consiste à produire une image de dimension  $h', l'$  avec la formule (1.7):

$$P = h/h_i \text{ et } l' = P * l_i \quad (1.7)$$

La normalisation en hauteur est exécutée en transformant chaque pixel  $(x, y)$  noir de l'image du caractère en  $(p*x, p*y)$ . La normalisation en largeur de l'image ainsi obtenue se fait par l'examen de deux cas. Si  $l' < l$ , alors l'image normalisée en hauteur est centrée dans

une surface de dimension  $h, l$ . Si, par contre  $l' > l$ , alors l'algorithme balaye cette image et assigne à noir tout pixel de coordonnées  $((x/l') * l, y)$  si le pixel  $(x, y)$  l'est aussi.



Figure 1-12 : Exemples de mots manuscrits avec des tailles différentes

### 3.3.4.2. Correction de l'inclinaison des lignes (Skew correction)

L'inclinaison des lignes de textes est une distorsion qui est souvent présentée pendant la phase de la numérisation du support, ou des défauts d'orientation causés par les habitudes des scribes. L'angle d'inclinaison est l'angle que forment les lignes de texte avec l'axe horizontal. Par conséquent, la correction Skew consiste à mettre horizontalement chaque ligne de l'écriture inclinée. Généralement, il ya deux types d'inclinaisons : l'inclinaison globale, quand toutes les lignes de textes ont la même orientation ; et l'inclinaison locale, lorsque certaines lignes de textes ont une orientation différente que les autres. Plusieurs méthodes sont disponibles pour l'inclinaison globale qui sont applicables même dans le cas des inclinaisons multiples. Ces algorithmes reposent sur le principe suivant : détection de l'angle de l'inclinaison et correction à l'aide de la rotation par cet angle. Les plus utilisés sont les histogrammes de projection horizontale [Vinc01], le gradient ([Ding00], [Bert07]), l'interpolation des minimums locaux de contour [Bozi89] et la transformée de Hough [Jian97]. Certaines de ces méthodes dépendent de script étudié, par exemple la méthode de [Bozi89] s'adapte bien à l'écriture latine mais pas le cas pour l'écriture arabe.

Dans les techniques basées sur les histogrammes de projection horizontale, le nombre des pixels noirs sont calculés selon des orientations qui sont proches de l'horizontale pour chaque ligne ou à partir d'autre caractéristiques qui sont liées aux composantes connexes. La Figure 1-13 ci-dessous illustre un exemple d'histogrammes de projection.

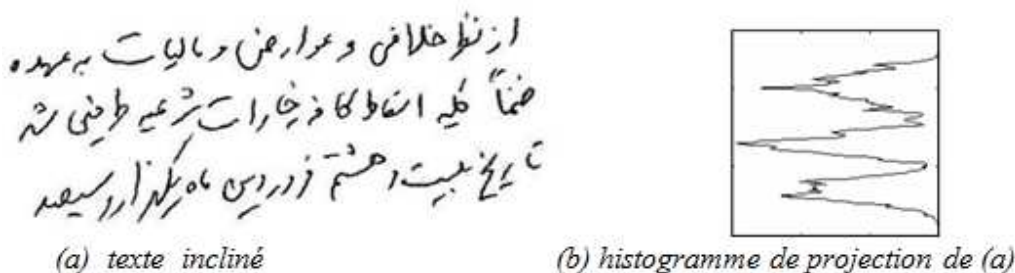


Figure 1-13 : la technique des histogrammes de projection pour la détection d'inclinaison [Haji05]

La direction la plus probable est celle qui maximise l'entropie. L'histogramme d'entropie maximale est celui dont les extremas sont les plus marqués. L'angle de la ligne  $\theta$  est celui

qui correspond à l'histogramme d'entropie maximale. Pour corriger cette inclinaison, il suffit d'appliquer une rotation de l'image d'angle  $\theta$  par la formule (1.8)

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \quad (1.8)$$

La Figure 1-14 ci-dessous montre l'angle  $\theta_s$  d'inclinaison d'un texte et le résultat de la correction par la méthode d'histogramme de projection.

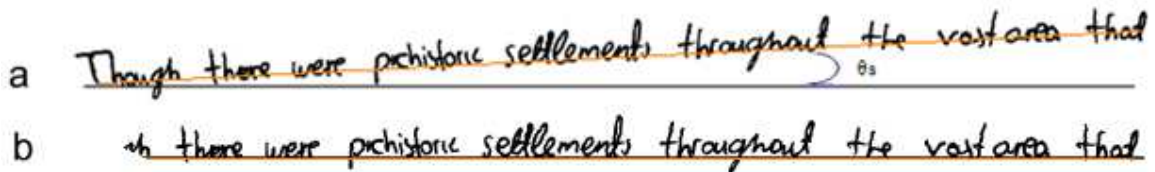


Figure 1-14 : (a) image originale avec l'angle d'inclinaison  $\theta_s$  (b) la correction d'inclinaison par les histogrammes projection [Loul09].

### 3.3.4.3. Correction de l'inclinaison des caractères (Slant correction)

L'inclinaison des caractères est définie comme étant l'angle entre l'axe correspondant à la direction moyenne des caractères et l'axe vertical (cf. Figure 1-16-a ci-dessous). Cette inclinaison de l'écriture est également appelée "Slant". Les lettres peuvent être inclinées vers la droite ou vers la gauche. Pour la même raison que dans la partie précédente, il convient de transformer le mot de façon à ce que cet axe de direction principale devient vertical (cf. Figure 1-16-b ci-dessous).

Les systèmes de reconnaissance corrigent toujours cette inclinaison pour diminuer la variabilité de l'écriture et pour rendre la segmentation en caractères plus facile, notamment dans le cas de l'utilisation des chaînes de Markov qui supposent que l'on peut segmenter en caractères par des lignes verticales (notion de fenêtre glissantes qui se déplace de gauche à droite dans le temps) (cf. section 3.4)

Généralement, il ya 2 types d'inclinaisons : l'inclinaison uniforme, quand toutes les lignes de textes ont la même orientation (inclinées vers la droite ou vers la gauche ou verticale); et l'inclinaison non uniforme (inclinées vers la droite et vers la gauche et verticale), lorsque certaines lignes de textes ont une orientation différente que les autres. Plusieurs techniques sont disponibles pour l'inclinaison uniforme ([Shri95], [Sun97], [Kava00]), qui sont applicables même dans le cas des inclinaisons non uniformes. Ces méthodes sont en général fondées sur la détection et la correction des traits quasi-verticaux. Les plus efficaces sont les histogrammes de projection verticale ([Shri95], [Kava00]), d'autres qui se basent sur le gradient ([Bert07], [Ding00]) et finalement celles qui utilisent des contours [Shri95].

Les histogrammes de projections verticales s'utilisent dans de nombreux travaux ([Vinc01], [Kava01]) de la même façon que pour la correction du Skew du paragraphe précédent, mais

cette fois dans des directions proches de la verticale. La Figure 1-15 ci-dessous montre les histogrammes des projections verticales selon les différentes inclinaisons.

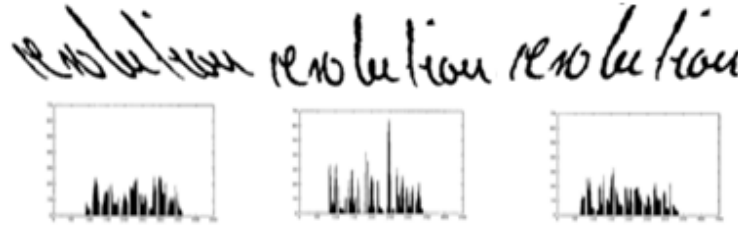


Figure 1-15 : Inclinaison de l'écriture et histogrammes de projection verticale correspondants [Mena08]

La Figure 1-16 ci-dessous présente un texte avec des caractères inclinés et la correction de cette inclinaison par la technique de l'histogramme de projection verticale.

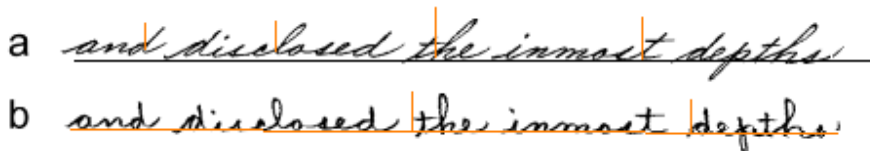


Figure 1-16 : (a) texte incliné avec une angle d'inclinaison par rapport à l'axe vertical; (b) texte corrigé [Loul09]

Dans [Uchi01], S. Uchida et al. proposent une bonne étude sur la correction d'inclinaison non uniforme et une méthode basée sur une optimisation locale permettant de corriger une inclinaison variable des lettres de toutes les positions horizontales dans les mots.

#### 3.3.4.4. Estimation des lignes de base

La ligne de base est définie comme la ligne sur laquelle repose les lettres ne possédant pas de dépassement bas, qui encadre, en général la bande des minuscules. Essentiellement, nous distinguons pour une telle ligne de texte, quatre lignes de références : la ligne haute, la ligne de base supérieure, la ligne de base, la ligne de base inférieure. Ces lignes sont illustrées par la Figure 1-17 ci-dessous. Ces lignes définissent les limites de trois zones significatives d'une ligne de texte : la zone ascendante, la zone médiane et la zone descendante. Le but des algorithmes de cette partie est de détecter ces droites horizontales.

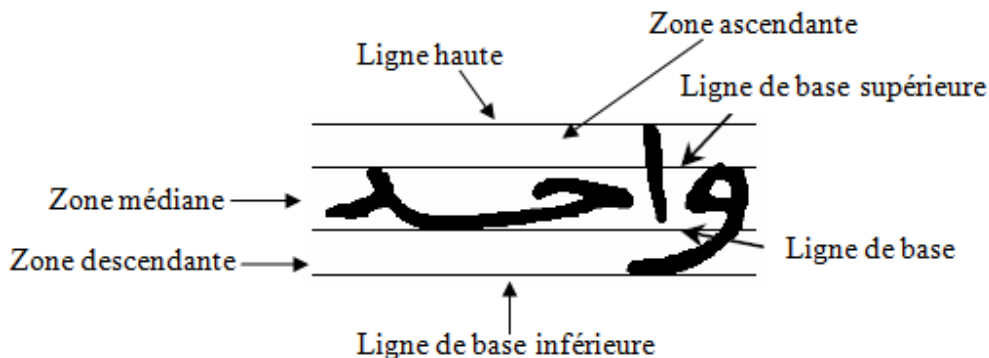


Figure 1-17 : Topologie d'un mot [Beno07]



Il ya plusieurs méthodes qui permettent de déterminer ces lignes de base dont chacune dépende de type d'écriture, par exemple la méthode qui estime la ligne de base en interpolant les minima locaux des contours [Bozi89]. Elle fonctionne bien pour les caractères latins, mais susceptible de ne pas fonctionner pour les lettres arabes. La plus largement applicable est la méthode qui est basée sur la maximisation d'entropie des histogrammes de projection ([Vinc01], [Naga07]) présentée dans la Figure 1-18 ci-dessous. En effet, le maximum se situe dans la bande de base, un seuil haut et un seuil bas permettent de déterminer une approximation de la bande de base. D'autres utilisent les extremas locaux des contours extérieurs de l'image de l'objet et la transformée de Hough [Wang97]. Finalement l'approche basée sur les splines ([Rein71], [Henn01], [Henn02]) dont les zones (ascendantes, médianes et descendantes) peuvent être évaluées de façon plus fine sur une ligne complète. Selon Sherkat et al [Sher05] cette méthode améliore le taux de reconnaissance.

L'utilisation large de ces lignes de base dans l'extraction et la normalisation des primitives, les rendent ainsi moins dépendantes de la hauteur de l'écriture. Par conséquent ils sont importants en reconnaissance.



Figure 1-18 : Histogramme de projection horizontale et bande de base [Adam06]

### 3.4. Segmentation

La segmentation est le processus consistant à décomposer l'image d'un texte en entités (mots, caractères ou graphèmes) qui font partie d'un alphabet prédéfini selon le but visé. Par ailleurs, elle permet de réduire la complexité des modules de traitements utilisés par la suite ([Case96], [Lu96]).

Dans les systèmes de reconnaissance la segmentation est une opération très critique [Hach04]. En effet, la séparation des lignes, des mots, des pseudo-mots, des caractères et des graphèmes constituent des opérations difficiles et coûteuses, tant les écritures sont variées, les lignes sont parfois enchevêtrées et les caractères généralement liés (cas de l'arabe : l'écriture est semi-cursive) les unes aux autres.

De ce fait, au cours de la mise au point de la segmentation, les chercheurs souvent ont recouru à deux phénomènes qui sont naturels, la sur-segmentation ou fausse détection (un caractère est découpé en plusieurs entités qui peuvent être problématiques ou non, selon l'utilisation qui en sera faite. Par exemple, la segmentation en petits segments) et la sous-segmentation ou non détection (quelques objets de segmentation sont à cheval sur deux caractères consécutifs. Ce cas pose un problème, car on ne peut pas reconnaître correctement certains caractères qui composent le mot).

Selon la littérature, le problème le plus ardu c'est le cas de la segmentation de l'écriture cursive, où la communauté de la reconnaissance de manuscrits admis le Paradoxe de Sayre suivant [Sayr73] « *une lettre ne peut être segmentée avant d'avoir été reconnue et ne peut être reconnue avant d'avoir été segmentée.* ». Dans le but de résoudre cette problématique plusieurs algorithmes de segmentation existent mais posent souvent une polémique autour de choix de l'un par rapport à l'autre, de ce fait, l'utilisation d'un algorithme est conditionnée par son efficacité et le type de la graphie étudiée. Cette efficacité ne peut pas être déterminée que relativement au traitement qui sera fait en aval de cette étape. Les solutions proposées se basent sur deux stratégies de segmentation différentes ci-dessous :

#### 3.4.1. Segmentation explicite

La segmentation explicite, s'appuie sur un découpage à priori de l'image en sous-unités qui peuvent être des lettres ou des graphèmes (cf. Figure 1-19 ci-dessous). Cette décomposition se base directement sur une analyse morphologique du texte ou de mot, ou sur la détection des points caractéristiques tels que les points d'intersection, les points d'inflexion, les boucles à l'intérieur du texte ou de mot pour localiser les points de segmentation potentiels [Blum02]. Dans le cas de segmentation en graphèmes, les textes ou les mots sont alors reconnus non comme une suite de lettres reconnues indépendamment, mais comme une suite de graphèmes globalement comparés à l'entrée.

L'avantage de cette segmentation c'est que l'information est localisée explicitement et ca va dans le sens de delimite de Sayre cité précédemment, puisque on sépare les lettres non pas d'après leur reconnaissance, mais d'après des critères topologiques ou morphologiques. Le défaut majeur de cette segmentation vient en premier lieu de choix des limites indépendant des critères des modèles : les limites sous-optimales pour les modèles et modélisation sous-optimale. En deuxième lieu, il n'existe pas de méthode de segmentation fiable à 100%, toute erreur de segmentation pénalise les performances de système.

Plusieurs approches proposent la segmentation directe d'un texte ou mot en graphèmes primitifs, suivi par une étape de combinaison de ces graphèmes en caractères ([Leco90], [Plam00], [Tay01]). Il existe quatre approches pour la mise en oeuvre d'une segmentation explicite : les approches basées sur des analyses par morphologie mathématiques [Mota97], permettent la sélection des points de segmentation en utilisant le principe de régularité et singularité ; les approches basées sur l'analyse des contours ([Srig01], [Dupr03], [Ball06]), déterminent les candidats de coupure en s'appuyant sur les extremas locaux du contour ; les approches basées sur l'analyse du profil d'histogramme de projection verticale [Yani07] et les approches basées sur l'analyse du squelette ([Dupr03], [Zerm07]), consistent à repérer les points de coupure sur le squelette on se basant sur des seuils ajustés.

#### 3.4.2. Segmentation implicite

La segmentation est dite implicite lorsque celle-ci est basée sur un moteur de reconnaissance pour valider et classer les hypothèses de segmentation (recherche de chemin des points de segmentation possibles) (cf. Figure 1-19 ci-dessous). Dans ce cas, la segmentation et la reconnaissance sont réalisées conjointement, d'où le nom parfois employé de "segmentation-reconnaissance intégrée".

Contrairement à la segmentation explicite, on ne procède pas à une segmentation à priori en entrée, mais à une segmentation aveugle du mot dans le sens où elle ne dépend en aucun

cas d'une analyse de l'image à segmenter et qui dépend d'une compétition des classes des lettres ou graphèmes en sortie du classifieur. Cette dernière recherche dans la séquence des segments, des composantes ou des regroupements de graphèmes qui correspondent à ces classes de lettres. Cette recherche peut se faire au moyen de la segmentation basée sur les fenêtres glissantes ([Dupr03], [Pech03], [Pech06], [Märg06], [Biad06]). L'avantage de cette segmentation c'est que l'information est localisée par les modèles des lettres et la validation se fait par ses modèles. Il n'y aura pas d'erreur de segmentation et enfin on contourne le dilemme de Sayre car en connaissant les lettres, on n'engendre pas d'erreur de segmentation.

Dans les approches à segmentation implicite, la tâche de segmentation est accomplie par le système. Elle est soutenue simultanément par un processus de reconnaissance, en évitant la pré-segmentation d'un mot en lettres ou entités plus fines ([Mile97], [Pech03]).

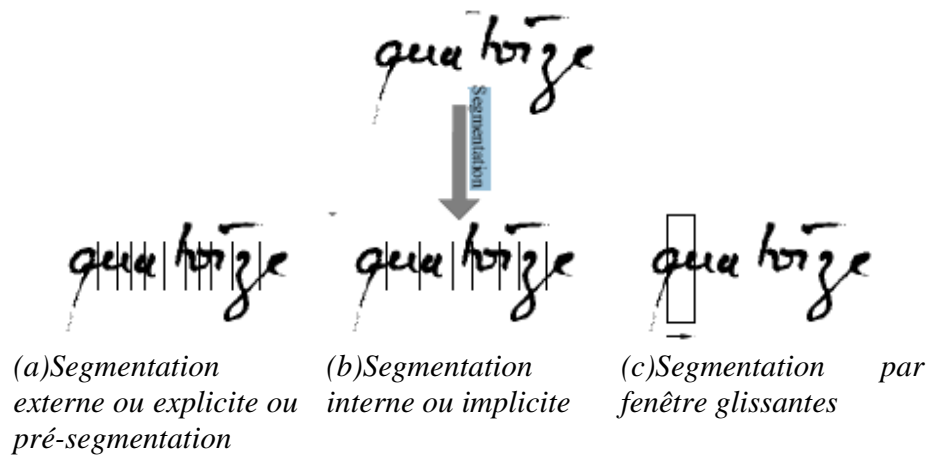


Figure 1-19 : Différentes segmentation pour le mot quatorze [Augu01].

Dans la suite, nous citons quelques exemples de segmentation en fonction de la taille de l'unité de base considérée et de la nature de problème posé :

**Segmentation de documents** : regroupe les méthodes ascendantes et descendantes, elle consiste à déterminer les zones d'informations dans chaque page. Cette segmentation permet d'étiqueter la nature du média représenté dans chaque zone (texte, graphique, photographie,...etc.). De ce fait, ces approches permettent la séparation et l'extraction des zones multi structurelles dans les documents ([Lebo00], [Hadj01], [Hadj03], [Anta03]).

**Segmentation d'un bloc de texte en lignes** : le but de cette étape est la séparation des différentes lignes du texte afin de faciliter les autres niveaux de segmentation. En s'appuyant souvent sur les histogrammes de projection horizontale. Dans ([Shi04], [Zaho07], [Yin08]) la projection horizontale est appliquée au document entier ou à une bande (horizontale ou verticale) de celui-ci. En suite, les maximas et minimas sont déterminés et les composantes connexes entre deux minimas consécutifs forment généralement les lignes en question. Cependant, dans certains cas, le chevauchement et l'interférence des ascendants et des descendants des lignes voisines rendent l'opération de segmentation difficile ([Likf07], [Ariv07]).

**Segmentation des textes ou des mots en caractères ou graphèmes :** Cette opération permet de découper l'image d'un texte, mot ou d'un pseudo-mot selon les lettres ou les graphèmes qui le composent. C'est une étape primordiale, critique et décisive dans quasiment tous les systèmes de reconnaissance analytique [Sell05]. Sa difficulté issue du fait que les frontières d'un caractère sont parfois absentes ou floues et ne peut se faire sans avoir identifié le caractère (Sayre).

### 3.5. Extraction de primitives

L'extraction des primitives souvent appelées caractéristiques, consiste à représenter les données d'entrée (mots, caractères, graphèmes) en un vecteur de primitives de dimension fixe. C'est une étape cruciale et critique dans les systèmes de reconnaissance. Le but de cette phase est la sélection de l'information pertinente, discriminante et de dimension limitée pour l'étape de la classification, toute en évitant le risque de perte des informations importantes et significatives. En effet, un mauvais choix des primitives influence négativement et nettement les résultats même si on utilise un classifieur très performant. Selon [Beno07] l'extraction de primitives est un « **problème d'extraction à partir de l'image, de l'information la plus pertinente, pour un problème de classification donné, c'est-à-dire celle qui minimise la variabilité intra-classe et qui maximise la variabilité interclasse** ».

La communauté de la reconnaissance considère cette étape critique pour les raisons :

- Mauvaise représentation des données par des primitives (manque de propriétés) conduit à des insatisfaisants résultats dans la reconnaissance.
- Beaucoup de redondance de la représentation conduit à des confusions (l'ensemble des échantillons disponibles pour l'apprentissage et classification est limité).

De ce fait, de très nombreux travaux récréments ([Gran00], [Oliv01], [Brit04]), sont consacrés à la recherche sur des méthodes d'extraction de primitives pertinentes, vue l'importance de sélectionner un ensemble de primitives discriminantes pour la conception des systèmes de reconnaissance puissants et performants (possédant des taux de reconnaissance élevés).

Dans la littérature, les primitives sont généralement classées en trois catégories principales : les caractéristiques statistiques, les primitives structurelles et topologiques et les transformations globales.

#### 3.5.1. Caractéristiques statistiques

Les primitives statistiques ([Dehg01a], [El-Ha05]) tentent de représenter l'information distribuée sur toute la totalité de l'image. Elles sont générées généralement par l'estimation de la densité de pixels appartenant à l'image entière (caractères ou mots) ou dans certaines parties uniquement, en utilisant des mesures statistiques (entropie, moyenne, variance, etc.) associées à des caractéristiques locales ([Al-Ma02a], [Al-Ma02b]) qui peuvent être par exemple:

- L'histogramme directionnel qui représente le nombre de pixels sur une ligne dans une direction quelconque de l'image.
- Les moyennes locales de pixels d'une fenêtre dans l'image (principe du "zoning").

- Les projections verticales et horizontales des densités de pixels noir/blanc qui représentent le nombre de pixels sur chaque ligne ou chaque colonne de l'image.
- Les directions des contours dans une fenêtre locale.
- L'histogramme des transitions noir/blanc et vice versa qui, comme son nom l'indique, permet de retenir le nombre des transitions 0-1 et 1-0 entre pixels.

La description des primitives statistiques se base sur les densités de pixels, de fait ces caractéristiques se considèrent moins discriminantes par rapport aux autres types car elles portent une information faible sur le signal d'écriture. Dernièrement, des tendances en littérature vers des primitives hybrides qui proposent la combinaison de différents types (statistiques, géométriques, structurelles...) afin de compenser les faiblesses de l'un par l'autre.

### **3.5.2. Caractéristiques structurelles**

Les primitives structurelles représentent les propriétés topologiques et géométriques de la forme. Ces caractéristiques sont extraites à partir de la représentation de la forme par le squelette ou le contour et non de l'image brute de l'écriture, il s'agit essentiellement de : concavités, convexités, occlusions, ascendants, descendants, composantes connexes, segments de droites et leurs attributs (position, orientation, ...), mesures de pentes, arcs, boucles, croisements, jonctions des traits, paramètres de courbures, angularités, points extrêmes et points terminaux, longueur et épaisseur des traits, surfaces et les périmètres ([Pal01], [Kapo03], [El-Ha05], [El-Ha06], [El-Ha07]). Par contre au type précédent, les primitives structurelles ont une grande capacité discriminative très forte, cette prétention influe sur la rapidité de la décision de l'appartenance d'une classe lors de la classification. De ce fait, la mauvaise gestion de ces primitives (mauvaise détection de leur présence) génère automatiquement des résultats insatisfaisants pendant le processus de reconnaissance. Ce type de primitives sera la base des séquences d'observations traitées par notre système Markovien proposé dans le chapitre 5.

### **3.5.3. Transformations globales**

Les transformations globales sont généralement largement utilisées auparavant dans le domaine de traitement de signal. Leurs objectifs c'est le changement d'espace de représentation de l'image (caractères ou mots) pour faciliter l'extraction de caractéristiques pertinentes. Les techniques les plus utilisées dans la reconnaissance de l'écriture sont :

La transformation de Hough a été définie à l'origine par Paul Hough en 1962 [Houg62] pour détecter la présence d'éventuelles formes génériques (droites, cercles, ellipses...) [Duda72]. Récemment cette méthode est largement explorée dans le domaine de la reconnaissance d'écriture vue sa robustesse et son efficacité dans l'extraction des primitives de haut niveau ([Maitre85], [Gink04]). Elle sera utilisée pour extraire le vecteur de primitives dans notre première contribution présentée plus tard dans le chapitre 4.

La deuxième est la transformée de Fourier est certainement une des plus utilisée en reconnaissance de l'écriture ([Khor00], [Mezg05a], [Mezg05b], [Zerm07]). Elle est invariante aux rotations et aux changements d'échelle. Dans cette transformation les descripteurs de Fourier, qui sont basés sur les coefficients complexes des séries de Fourier, représentent généralement les caractéristiques recherchées.

La troisième méthode est la transformée en ondelettes qui est très semblable à celle de Fourier [Shen99], mais n'est pas invariante à la translation. La raison pour laquelle est moins utilisée en reconnaissance, un simple décalage changera totalement les coefficients d'ondelettes. Ces dernières permettent d'obtenir une information fréquentielle localisée pour un signal en entrée (image de caractères ou de mots). Les primitives extraites sont particulièrement intéressantes dans la décision.

Une autre transformation globale est la transformation de Karhunen-Loève (KL), qu'est largement utilisée récemment pour la reconnaissance de l'écriture ([Laak98], [Pech03], [Pech06], [Märg06]). Le défaut majeur de cette transformation est qu'elle est coûteuse en terme de calcul, par rapport à d'autres, comme les transformations Fourier, Walsh et Haar.

Dans cette catégorie figure aussi la grande famille des moments invariants. Ces outils sont des caractéristiques intéressantes car ils sont invariants à la rotation, à la translation et au changement d'échelle [Wood98]. Les primitives issues de cette famille sont un résultat d'une transformation exécutée globalement au niveau des pixels de l'image. Elles se calculent principalement par des mesures statistiques de la distribution des pixels autour du centre de gravité du caractère. La grande utilisation des moments s'explique par leur nature, sont très peu sensibles aux variations locales de la forme, et ainsi que leur invariance aux opérations précitées. Les moments invariants existent sous divers formes, les plus populaires par exemple, les moments de Hu ([Aziz02a], [Aziz02b], [Klas02]) et celle de Li [Li92]. Le premier et le deuxième moment de Hu seront exploités dans la génération de la séquence d'observations utilisée pour entraîner les modèles de Markov cachés dans notre système présenté plus tard dans le chapitre 5.

Cependant les plus utilisés actuellement sont ceux dérivés des polynômes complexes de Zernike ([Mezg02], [Mezg03], [Pete04], [Mezg05a], [Mezg05b]). La raison est que ces dernières ont des performances supérieures en termes d'invariance.

En fin, certains auteurs utilisent les propriétés des filtres de Gabor afin d'extraire des caractéristiques [Shah06]. Le Tableau 1-2 ci-dessous illustre un exemple récapitulatif de quelques caractéristiques utilisées dans certains travaux proposés dans la littérature.

<b>Proposition</b>	<b>Primitives</b>
Menasri 2009 [Mena08]	Descripteurs de Fourier, quelques propriétés basées sur le squelette (boucles, points de branchements...).
Benouareth 2006 [Beno07]	Statistiques : intensité et distribution de pixels d'écriture structurelles : concavités et directions des traits locaux, points de branchement, points d'inflexion et boucles.
El-hajji 2005 [El-Ha05]	Les densités de pixels, les transitions de densité, concavité et centre de gravité.
Lorigo 2005 [Lori06]	Structurelles : les minimums de contour, les boucles, les traits en haut (upstrokes), points de branchements et points terminaux.
Safabakhsh 2005 [Safa05]	Les descripteurs de Fourier, nombre de boucles, le rapport hauteur/largeur, les densités de pixels et les positions de connexions gauche et droite.

Mozaffari 2005 [Moza05]	Moyen et variance de X et Y qui varient sur les portions de squelette.
Alma'adeed 2004 [Al-Ma04]	Ascendants, descendants, les primitives structurelles et caractéristiques basées sur les frames.
Khorsheed 2003 [Khor03]	Structurelles : extraites sur le squelette du mot comme les points diacritiques, les points de croisement, les liens et les boucles.
Pechwitz 2003 [Pech03]	Statistiques: intensité de pixels dans des fenêtres verticales.

Tableau 1-2 : Exemples de certaines caractéristiques utilisées en reconnaissance de l'écriture

### 3.6. Apprentissage

La faculté d'apprendre est essentielle à l'être humain pour reconnaître une voix, une personne, un objet, symbole, caractère... L'apprentissage par généralisation où l'on apprend à partir d'exemples un modèle qui nous permettra de reconnaître de nouveaux exemples.

Dans les systèmes de reconnaissance d'écriture, il est difficile d'apprendre par généralisation de la même manière que l'homme. Par exemple, il leur est difficile de construire un bon modèle d'un mot ou d'un caractère et d'être ensuite capable de le reconnaître efficacement dans de nouvelles images.

Dans la littérature, plusieurs recherches sont penchées sur la question de l'apprentissage automatique ([Assr07], [Augu00], [Dupr03], [Beno07]). D'après ces travaux l'apprentissage peut se décliner en différents types principaux suivants :

**Apprentissage supervisé** : qui suppose que chaque donnée d'apprentissage soit étiquetée, par exemple lorsqu'on dispose d'un ensemble de données dont on connaît la classe d'appartenance a priori. Il s'agit alors de chercher les surfaces de décision séparant au mieux les classes. Ainsi c'est l'ensemble des techniques qui visent à deviner l'appartenance d'un individu à une classe.

**Apprentissage non-supervisé** : Contrairement à l'apprentissage supervisé, dans l'apprentissage non-supervisé il n'y a pas des données étiquetées. Autrement lorsqu'on dispose d'une base de données sans connaître leur classe d'appartenance. Il s'agit de découvrir la structure sous-jacente aux données sans imposer aucun modèle.

**Apprentissage semi-supervisé** : Cet Apprentissage est un bon compromis entre apprentissage supervisé et non-supervisé. Car, il suppose qu'on dispose de peu de données étiquetées et d'un grand nombre de données non étiquetées. L'apprentissage s'effectue alors à partir des deux sources de données.

**Apprentissage par renforcement** : désigne toute méthode adaptative permettant de résoudre un problème de décision séquentielle. Ce type d'apprentissage est utile dans le cas d'apprentissage interactif. Pendant cet apprentissage, le système adaptatif agit en interaction avec son environnement, et en retour reçoit des signaux de renforcement.

Récemment, des méthodes d'*apprentissage croisé* à base des HMMs sont proposées pour résoudre les problèmes de ré-estimation des modèles de lettres dans les approches analytiques. Ces méthodes consistent à entraîner les modèles des lettres globalement, au

travers des mots ([Vinc00], [Choi03], [El-Ha06]). Cela se traduira par la maximisation de la vraisemblance des mots au travers des modèles de lettres (cf. section 3.3.3).

### 3.7. Classification

Dans le processus complet d'un système de reconnaissance de formes, la classification joue un rôle important en se prononçant sur l'appartenance d'une forme à une classe. L'idée principale de la classification est d'attribuer un exemple (une forme) non connu à une classe prédéfinie à partir de la description en paramètres de la forme. Nous présenterons par la suite, une panoplie de classifieurs couramment utilisés en reconnaissance de l'écriture, et en montrant leurs forces et de leurs faiblesses. Nous passerons en revue, les méthodes connexionnistes, méthodes structurelles et syntaxiques, les machines à vecteurs supports (SVM) et les Modèles de Markov cachés.

#### 3.7.1. Méthodes connexionnistes

Les réseaux de neurones artificiels (RNA), un exemple est donné dans la Figure 1-20 ci-dessous, ces moyens sont appelés des réseaux connexionnistes. Ils sont des moyens fortement connectés de processeurs élémentaires distribués fonctionnant en parallèle et qui servent à l'apprentissage et la classification.

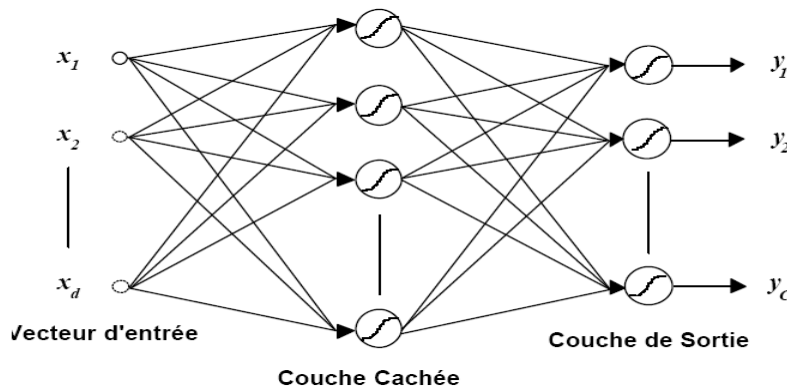


Figure 1-20 : Exemple d'RN : perceptron Multicouche à une couche caché

L'idée principale est que chaque processeur calcule une sortie à partir des informations qu'il reçoit en entrées. Ce processeur est représenté physiquement par un neurone formel, modèle mathématique inspiré du modèle biologique (cf. Figure 1-21 ci-dessous).



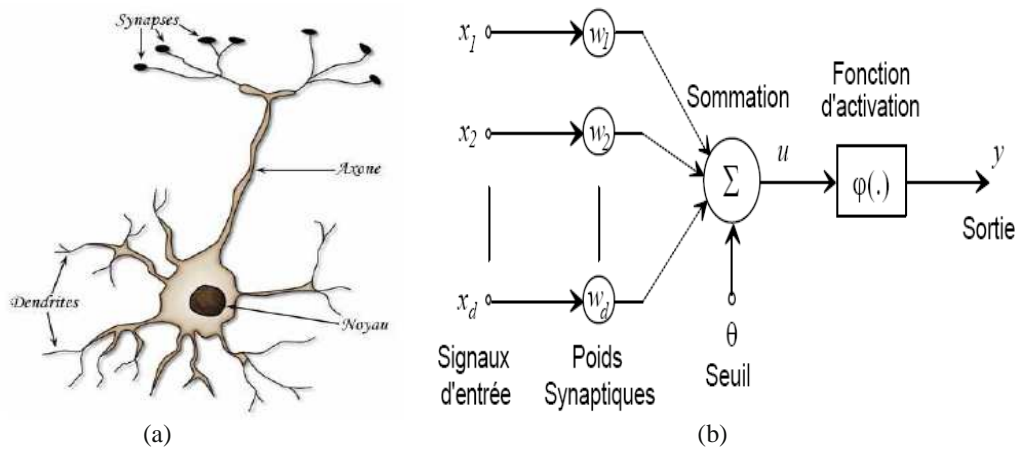


Figure 1-21 : (a) : Modèle biologique, (b) : modèle mathématique.

Le neurone formel est un automate possédant  $n$  entrées réelles. Son traitement consiste à effectuer à sa sortie le résultat d'une fonction de seuillage de la somme pondérée de ses entrées. Récemment, les réseaux de neurones connus un grand succès dans plusieurs domaines, notamment le domaine de reconnaissance grâce à la simplicité et l'efficacité de leurs algorithmes d'apprentissage [Bish06]. Plusieurs approches d'entraînement sont utilisées pour apprendre les RNA, tel que l'apprentissage par correction d'erreurs, l'apprentissage de Hebb, et l'apprentissage compétitif. L'objectif de l'apprentissage d'un réseau de neurones est de lui faire apprendre à fournir les sorties voulues pour un ensemble de valeurs d'entrée. Pour cela, on se base sur un très grand nombre d'exemples qui permettent d'ajuster les paramètres - les poids des connexions - de manière à obtenir les sorties désirées en fonction des entrées.

Les RNA peuvent être classifiés selon l'acheminement de l'information et leurs topologies, en deux grandes classes principales : les réseaux de neurones à propagation directe dont le sens des données est direct sans retour en arrière, et les réseaux de neurones récurrents.

Actuellement, les types les plus utilisés dans les systèmes de reconnaissance de l'écriture sont les perceptrons multicouches (MLP) ([Beng94], [Plam00], [Blum02], [Amin03], [Aziz02a], [Klas02], [Sell05], [Fara06], [Al-Ma06], [Soui06], [Zerm07]), à propagation directe, les mémoires associatives ou les cartes de Kohonen (SOM) "Self Organizing Map" ([Mezg02], [Mezg03], [Mezg05a], [Mezg05b]) qui sont des RNA récurrents. Elles permettent de détecter automatiquement les prototypes de caractères dans un ensemble d'exemples d'apprentissage. On trouve aussi les réseaux à fonctions de base radiales [Alim97], les réseaux à délai [Alle03], les réseaux à convolutions [Lecu98], les réseaux profonds (DBN) [Lecu07] et les réseaux diabolos [Werf99].

Bien que les réseaux de neurones sont des classificateurs qui possèdent un grand pouvoir discriminant, 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); la détermination du paramètre d'ajustement des poids de connexions est très délicate; le risque de sur apprentissage et le problème de convergence vers des minima locaux.

### 3.7.2. Méthodes structurelles et syntaxiques

Les méthodes structurelles de reconnaissance se basent directement sur la structure physique des mots ou des caractères. Elles consistent à décrire ces formes complexes par des primitives plus simples. Dans le cas de reconnaissance de l'écriture, les primitives sont de genre topologique et géométrique (traits, boucles, points,...). Dans cette approche on distingue entre les techniques de classification structurelle suivantes :

#### - *Les méthodes syntaxiques*

Ces méthodes se basent directement sur la théorie des langages formels, chaque forme est représentée par une phrase dans un langage où le vocabulaire est constitué de primitives. Définir une classe de formes revient alors à établir des règles syntaxiques (la grammaire) caractérisant les phrases acceptables. La reconnaissance syntaxique consiste alors à déterminer si la phrase de description de la forme peut être générée par la grammaire. Cette approche a été utilisée en reconnaissance de l'écriture [Bagh05]. La faiblesse retenue pour ces méthodes est l'absence des algorithmes efficaces concernant l'inférence grammaticale directe, c.-à-d. comment construire une grammaire à partir d'un ensemble fini de phrases convenablement choisies (on parle alors l'apprentissage syntaxique).

#### - *Les méthodes structurelles*

Ces méthodes se déclinent selon la structure utilisée :

**À base d'arbres et de graphes** : Avec les méthodes graphiques [Roch95], les unités d'écriture (graphèmes, caractères, et mots) sont représentées par des arbres et des graphes dans la phase d'entraînement. Ces graphes permettent de décrire les caractéristiques primitives de ces unités (traits, segments, points d'inflexion, points de branchement...). Dans la phase de la décision, un graphe inconnu est assigné à une des classes des unités d'écriture, en utilisant des mesures de similitude d'arbres ou de graphes.

**À base de chaînes** : Dans ce cas, les unités d'écriture (graphèmes, caractères et mots) sont représentées par des chaînes de primitives. La méthode consiste à mesurer la similitude entre les chaînes des entités à reconnaître et un modèle de référence par un calcul de distance (distance d'édition) [Alsa06].

### 3.7.3. Machines à vecteurs supports : Support Vector Machine (SVM )

Ces méthodes sont des classifieurs à deux classes introduites par le mathématicien Vladimir Vapnik ([Vapn95], [Burg98]), qui possèdent une grande capacité de généralisation. Les SVMs sont un ensemble d'algorithmes d'apprentissage qui permettent de discriminer les formes. L'idée principale est que deux classes peuvent être linéairement séparées dans un espace de grande dimension comme le montre la Figure 1-22 ci-dessous. Dans le cas où les points sont séparables, on trouve souvent une infinité d'hyperplans séparateurs.

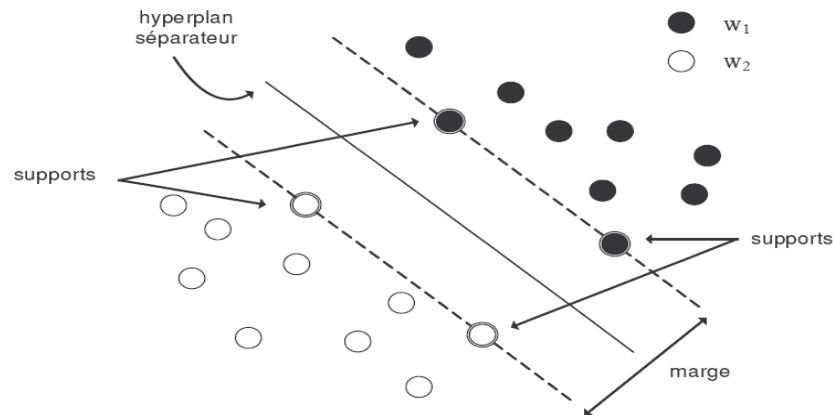


Figure 1-22 : SVM: Hyperplan séparateur de deux classes  $w_1$  et  $w_2$ .

Les SVMs sont des modèles discriminants qui tentent de minimiser les erreurs d'apprentissage tout en maximisant la marge entre classes, c'est-à-dire l'espace sans exemple autour de la frontière de décision. Pour cela l'algorithme d'apprentissage sélectionne judicieusement un certain nombre de "vecteurs support" parmi les exemples de la base d'apprentissage, qui définissent la frontière de décision optimale.

### 3.7.4. Modèles de Markov cachés

Les HMMs ou modèles de Markov cachés ([Rabi89], [Rabi93]) ont été introduits dans les années 70. Ils sont des outils statistiques puissants qui permettent de calculer la probabilité d'appartenance d'une forme à une classe. Contrairement aux approches précédemment décrites, les approches Markoviennes sont bien adaptées à la modélisation des données séquentielles. En conséquence, elles sont largement en reconnaissance automatique de l'écriture ([Dupr03], [Mena08]).

Les HMMs possèdent notamment plusieurs avantages en reconnaissance de l'écriture. En effet, ils prennent en compte la variabilité des formes et du bruit qui perturbent l'écriture notamment dans le manuscrit. Aussi, ils permettent de prendre en compte des séquences de longueurs variables. Cette qualité est particulièrement primordiale en reconnaissance de l'écriture manuscrite, où la longueur des lettres, des mots peuvent varier considérablement en fonction des styles d'écriture et des habitudes des scribes. Deux séquences de la même classe ne sont pas forcément de longueurs identiques. Finalement ils s'appuient sur un ensemble d'algorithmes standards et éprouvés. L'implémentation de ces algorithmes s'appuie sur des techniques bien maîtrisées de la programmation dynamique. En plus, un certain nombre de bibliothèques HTK, GHMM et ESMERALDA et de modules sont également publiquement accessibles pour l'apprentissage et le décodage de Modèles de Markov Cachés [Youn93].

En reconnaissance de l'écriture, les graphèmes, les caractères ou les mots sont souvent modélisés par des états cachés dont leur succession est donnée par une chaîne de Markov pour laquelle quelques exemples sont donnés dans la Figure 1-23 ci-dessous.

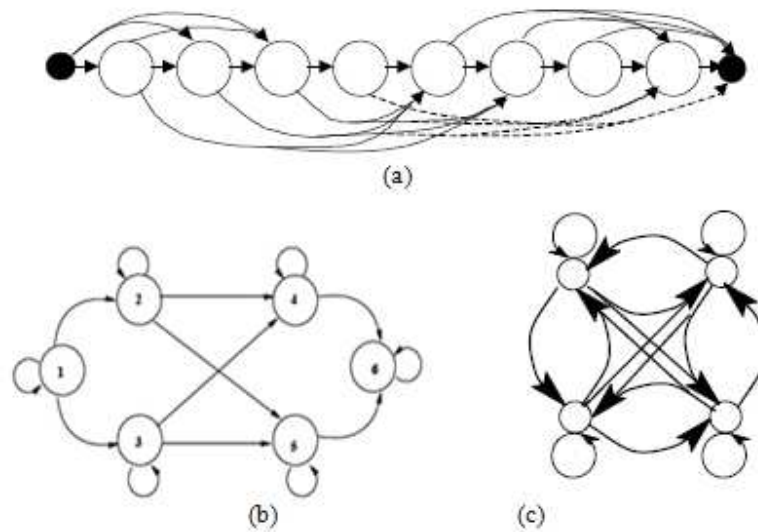


Figure 1-23 : Quelques exemples d'architecture de HMM (a) : linéaire, (b) : parallèle, (c) ergodique

Nous détaillerons succinctement ces modèles dans le chapitre 2, puisque c'est les classifieurs qu'on développés et utilisés dans nos contributions.

### 3.7.5. Combinaison de classifieurs

Dans le but d'améliorer les performances des systèmes de reconnaissance et pour pallier certaines limitations, telles que les résultats moins satisfaisants et instabilités ou complexité trop importante, de nombreux travaux actuels visent à combiner plusieurs classifieurs. En vue, d'exploiter au maximum la complémentarité des modélisations pour obtenir des décisions plus robustes et compenser les faiblesses de chaque classifieurs [Zoua05] Les types de combinaison sont regroupés en trois schémas différents :

- **La combinaison séquentielle** : consiste à appliquer une méthode de classification sur les résultats obtenus par une méthode précédente. Elle est donc organisée en niveau successifs de décision permettant le filtrage progressif des décisions et réduction de l'ambiguïté (cf. Figure 1-24 ci-dessous).

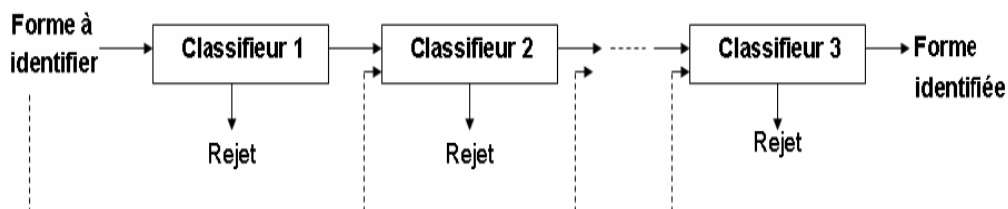


Figure 1-24 : schéma de combinaison séquentielle

- **La combinaison parallèle** : dans laquelle la forme à reconnaître est présentée simultanément à plusieurs classifieurs (cf. Figure 1-25 ci-dessous). Les réponses sont "concaténées" pour former le vecteur d'entrée d'un module de combinaison qui prend la

décision finale en fusionnant les informations fournies par chacun des classifieurs [Fara06]. Cette technique est caractérisée par sa facilité de mise en oeuvre.

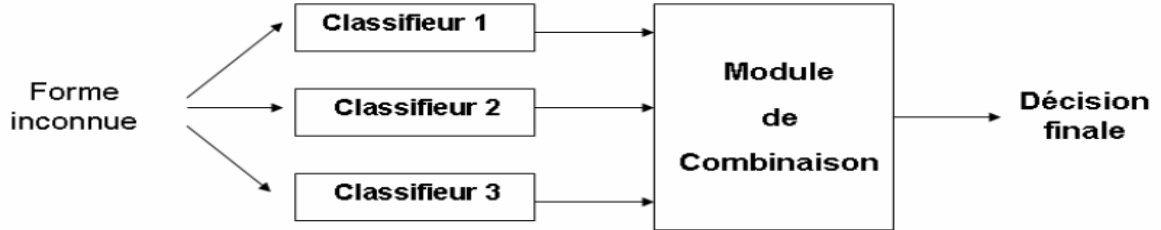


Figure 1-25 : Schéma de combinaison parallèle

- **La combinaison hybride** : consiste à combiner des architectures séquentielles et parallèles. L'idée principale, est de combiner les avantages de ces deux architectures et d'empêcher les inconvénients. Les méthodes de combinaison hybride sont généralement difficiles à optimiser, et la performance de ces méthodes dépende des données à exploiter [Zoua05].

### 3.8. Post-traitements

Lorsque la reconnaissance aboutit à la génération d'une suite de mots possibles éventuellement classés par ordre de vraisemblance, les post-traitements permettent d'améliorer les taux de reconnaissance par la prise en compte de connaissances pragmatiques et de connaissances linguistiques.

**Connaissances pragmatiques** : Dans certaines applications comme celles des systèmes de lecture d'adresses, le nombre de chiffres du code postal est connu. D'autre part, la confrontation entre la reconnaissance de ce code postal et du bureau distributeur permet de lever certaines ambiguïtés ([El-Ya99], [El-Ya02]).

**Connaissances linguistiques** : L'intégration, à un système de lecture, de connaissances linguistiques de différents niveaux (lexical, syntaxique ou sémantique) permet au système d'effectuer des vérifications et des corrections. L'utilisation d'un lexique ou d'un dictionnaire permet de valider a posteriori la reconnaissance effectuée. Mais pour éviter un temps de calcul prohibitif, diverses techniques de pré-organisation et d'interrogation du lexique peuvent être utilisées [Gill96]. L'utilisation d'un modèle du langage permet de moduler le taux de confiance des hypothèses de mots reconnus [O'boy94]. Les statistiques de type n-grammes et les modèles de Markov, notamment, permettent d'introduire cette connaissance dans le processus de décision finale ([O'boy94], [Kuhn94]).

La syntaxe confirme ou non, suivant les règles grammaticales prédéfinies, la séquence de mots proposés. Celle-ci est largement utilisée dans le cadre de la lecture des montants littéraux et numériques des chèques [Saon97]. Le contrôle sémantique est lié à l'aspect polysémique d'un mot ou d'une phrase. Il permet de réduire la liste des mots candidats, mais sa généralisation à de grands vocabulaires pose des difficultés.

## 4. Mesure des performances

Afin de juger le degré d'efficacité de ces systèmes de reconnaissance de l'écriture et de mesurer leurs performances. Plusieurs critères ont été proposés dans la littérature [Mena08] Les évaluations sont généralement caractérisées, par les différents taux ci dessous :

- *le taux de reconnaissance* : pourcentage de formes bien reconnues;
- *le taux de confusion* : pourcentage de formes pour lesquels le système fait une erreur ;
- *le taux de rejet* : pourcentage de formes pour lesquels le système refuse de se prononcer ;
- *le taux de confiance* : pourcentage de formes bien reconnues par rapport à la somme des formes bien reconnues et des formes mal reconnues.

Néanmoins, il ne faut pas considérer ces évaluations de manière absolue mais les replacer dans le contexte de l'application. Pour un type d'application donné, la comparaison des performances des différents systèmes de reconnaissance ne prend une signification que si ces systèmes sont testés sur des bases de données communes. Certains systèmes peuvent présenter des bonnes performances pour un type d'application et être très décevants pour un autre type. Tout dépend de la difficulté de l'application. Cette difficulté est déterminée par deux facteurs : l'ambiguïté des formes à reconnaître qui est difficilement mesurable, la dimension de l'espace des solutions.

## 5. Domaines d'applications de la reconnaissance de l'écriture

La reconnaissance automatique hors ligne de l'écriture présente un intérêt indéniable dans la réalisation des nombreuses et grandes applications comme celles que l'on rencontre dans certains secteurs d'activités parmi lesquelles on peut citer :

- l'authentification et la lecture de chèques bancaires par la reconnaissance des montants littéraux et des montants numériques manuscrits [Augu00];
- la lecture automatique de formulaires et de documents administratifs [Héro98];
- la lecture des adresses postales et le tri automatique du courrier ([El-Ya02], [Gran03], [Srih97]) ;
- l'échange de fichiers informatisés à distance dans le domaine de télécommunications ;
- l'indexation et l'archivage automatique de documents [Lebo07] ;
- la reconnaissance de documents techniques (schémas électronique, dessins techniques, plans architecturaux, plans cartographiques, etc.) ;
- la transcription assistée par ordinateur appliqué à la reconnaissance du manuscrit mono-scripteur qui permettra à la rétro-conversion de manuscrits anciens;
- la reconnaissance de numéros minéralogiques pour le contrôle routier, l'authentification et l'identification de manuscrits et l'identification du scripteur ;
- la recherche d'information dans une base de documents manuscrits telle que l'identification du scripteur [Bens04] ;

- les applications de reconnaissance de l'écriture en ligne [Liwi05] à travers les PDA, Tablet- PC, Ordinateurs sans clavier, ou stylo caméra.

## **6. Conclusion**

Ce chapitre a permis de présenter les définitions et les notions liés à la chaîne de reconnaissance de l'écriture. Nous avons en premier lieu exposé globalement les principaux aspects liés à cette tâche, en se basant sur un diagramme fonctionnel de référence comportant plusieurs étapes. Nous avons abordé chaque étape du schéma (acquisition, prétraitement, normalisation, etc.) avec analyse et spécification des différentes méthodes et techniques qui y sont développées et utilisées.

Dans le chapitre suivant nous allons introduire la théorie des modèles de Markov cachés, et nous mettrons l'accent sur les différents algorithmes d'apprentissage et de la reconnaissance de ces outils en soulignant leurs particularités. En effet les systèmes de reconnaissances développés et utilisés au cours de ce travail sont basés sur une modélisation markovienne des caractères et des graphèmes.

# Chapitre 2 : Les Modèles de Markov Cachés

---

## 1. Introduction

Les modèles de Markov cachés (HMMs : Hidden Markov Models) constituent des outils statistiques qui fondent sur une théorie solide. Ils se caractérisent par une grande efficacité et souplesse dans les algorithmes qu'ils offrent pour la modélisation des problèmes traités. Ces modèles sont récemment largement utilisés en domaine de reconnaissance de l'écriture profitant de leurs grand succès dans la reconnaissance de la parole où les HMMs sont initialement et fréquemment employés.

Grâce à leur relative simplicité, robustesse et élasticité, les HMMs ont été utilisés dans plusieurs domaines. Parmi ces domaines : la modélisation des langages, la modélisation de la météorologie ou de climats, la cardiologie, l'évaluation des scénarios lors de la simulation du déplacement d'un robot, la théorie de codage, la modélisation des finances, le contrôle biologique, la bioinformatique, etc.

Le but de ce chapitre est de présenter les concepts théoriques et les outils requis pour la compréhension et l'utilisation des HMMs. Il est organisé comme suit :

La première partie rappelle les différentes définitions nécessaires pour la compréhension des concepts liés aux HMMs. La deuxième partie présente en détails les HMMs classiques appelés HMMs unidimensionnels (1D) qui considèrent la forme comme un signal unidimensionnel. La deuxième partie étend partiellement la notion des HMM 1D au domaine planaire : PHMM (Planar-HMM) ou HMM pseudo-2D. La troisième partie introduit les champs aléatoires de Markov qui constituent de véritables HMMs 2D. Nous terminerons ce chapitre par les applications possibles de ces outils.

## 2. Définitions préliminaires

Les définitions données ci-après sont adoptées de ([Bela97],[Saon97]).

### 2.1. Variable aléatoire

Une variable aléatoire est une fonction mesurable  $X : \Omega \rightarrow S$ , définie sur un espace de probabilités  $\Omega$  avec des valeurs dans un ensemble de réalisations (appelé parfois ensemble d'états).



## 2.2. Processus stochastique

Un processus stochastique  $\Omega\{X_t, t \in T\}$  est un ensemble de variables aléatoires définies sur un espace probabilisé  $\Omega$ , souvent noté par :  $X_t, t \in \{t, \dots, T\}$  avec  $t$  représente le temps.

## 2.3. Chaîne de Markov

Une chaîne de Markov discrète d'ordre  $n$  est un processus stochastique discret  $X = \{X_t | t = 1, \dots, T\}$  avec  $X_t$  des variables aléatoires discrètes, vérifiant la propriété de Markov :

$$p(X_t = s_t | X_{t-1} = s_{t-1}, \dots, X_{t-n} = s_{t-n}, X_{t-n-1} = s_{t-n-1}, \dots, X_1 = s_1) = p(X_t = s_t | X_{t-1} = s_{t-1}, \dots, X_{t-n} = s_{t-n}) \quad (2.1)$$

$\forall t \in [1, T]$  et  $s_1, \dots, s_2 \in S$

$S = \{s_1, \dots, s_T\}$  représente l'ensemble d'états. C'est à dire, l'état courant dépend seulement des  $n$  états précédents.

Pour une chaîne de *Markov discrète d'ordre 1*, seul l'état actuel et son prédécesseur sont considérés,

$$p(X_t = s_t | X_{t-1} = s_{t-1}, \dots, X_1 = s_1) = p(X_t = s_t | X_{t-1} = s_{t-1}) \quad (2.2)$$

Une chaîne de Markov d'ordre 1 est dite *stationnaire*. C'est à dire l'état courant dépend seulement sur un état précédent et il ne dépend pas du temps, formellement :

$$\forall t, k \ p(X_t = s_t | X_{t-1} = s_j) = p(X_{t+k} = s_i | X_{t+k-1} = s_j) \quad (2.3)$$

Dans ce qui suit, nous nous limitons à la présentation uniquement des HMMs discrets et continus d'ordre 1. Nous renvoyons aux références ([Rabi89], [Juan91]) et aux ouvrages ([Huan90], [Rabi93], [Bish06]), pour une synthèse plus détaillée des HMMs.

## 3. Modèles de Markov cachés unidimensionnels ( HMM 1D)

Un modèle de Markov caché unidimensionnel (MMC), en anglais Hidden Markov Model (HMM) ([Rabi89], [Rabi93]), est une chaîne de Markov stationnaire où l'observation est une fonction probabiliste de l'état, qui est caractérisée par un système à état doublement stochastique constituant deux processus. Le premier est un processus de changement d'état appelé processus caché ou interne et qui n'est pas observable. Cependant, il peut être observable par l'intermédiaire d'un second processus d'émission appelé processus externe (cf. Figure 2-1 ci-dessous).

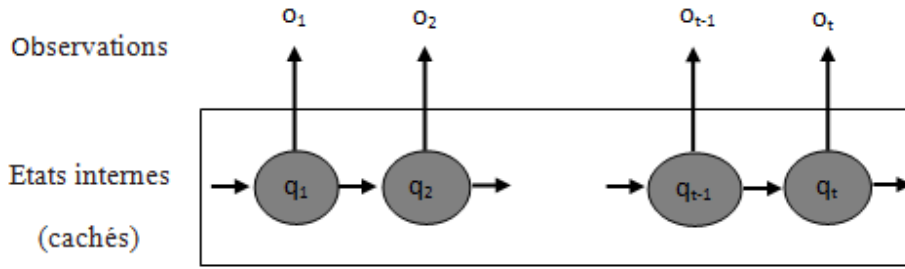


Figure 2-1 : les composantes cachées et observables de modèle

Le comportement de modèle de Markov caché HMM est lié donc aux deux suites de variables aléatoires qui sont associées respectivement à la composante observable et cachée du modèle stochastique.

La suite cachée correspond à la suite des états  $q_1, q_2, \dots, q_T$ , représentée par  $Q(1:T)$  avec les  $q_i \in \{s_1, \dots, s_N\}$  l'ensemble de  $n$  états de modèles.

La suite observable correspondant à la séquence d'observations  $o_1, o_2, \dots, o_T$ , notée par  $O(1:T)$  qui peut être continue ou discrète selon l'application donnée.

Un modèle de Markov caché d'ordre 1 est noté par l'ensemble des paramètres  $\lambda = (N, M, A, B, \Pi)$  avec :

$N$  : représente le nombre d'états dans le modèle HMM, l'ensemble des états  $\in \{s_1, \dots, s_N\}$ .

$M$  : représente le nombre de l'ensemble des symboles distincts observables dans chaque état. Ces symboles appartiennent à l'alphabet  $\in V = \{v_1, \dots, v_M\}$ .

$$A = \{a_{i,j}\}_{1 \leq i,j \leq N} \quad \text{avec}$$

$$a_{i,j} = P(q_t = s_j \mid q_{t-1} = s_i) \quad (2.4)$$

$$\sum_{j=1}^N a_{i,j} = 1$$

Avec  $q_t$  désigne l'état à l'instant  $t$  et  $A$  est la matrice des probabilités de transition entre états.

$$B = \{b_j(k)\}_{1 \leq j \leq N; 1 \leq k \leq M} \quad \text{avec}$$

$$b_j(k) = P(v_k = o_t \mid q_t = s_j) \quad (2.5)$$

$$\sum_{k=1}^M b_j(k) = 1$$

on désigne par  $o_t$  le symbole observé à l'instant  $t$  et  $B$  est la matrice des probabilités d'émission dans les états.

$$\begin{aligned} \Pi &= \{\pi_i\}_{1 \leq i \leq N} && \text{avec} \\ \pi_i &= P(q_1 = s_i) && (2.6) \\ \sum_{i=1}^N \pi_i &= 1. \end{aligned}$$

$\Pi$  est le vecteur des probabilités initiales des états.

La Figure 2-2 ci-dessous présente le graphe d'un HMM à 3 états :

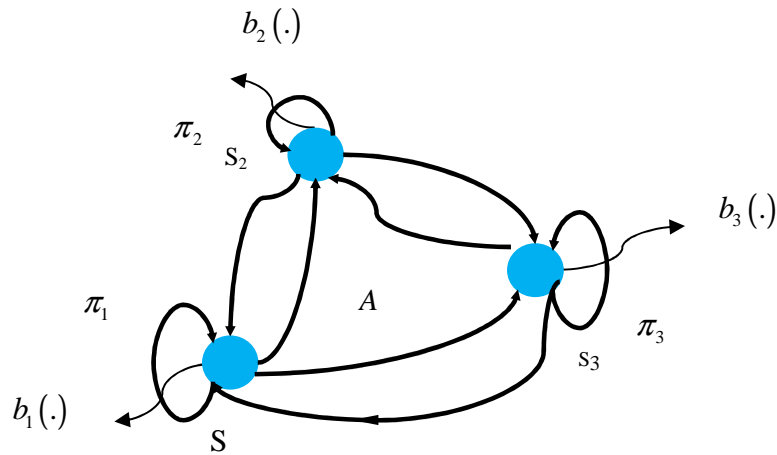


Figure 2-2 : Représentation graphique d'un modèle HMM à 3 états.

### 3.1. Topologie des modèles de Markov cachés

Selon la topologie de réseau des états, nous distinguons entre deux types principaux des HMMs utilisés en reconnaissance de l'écriture. Ce sont le modèle *ergodique* et le modèle *gauche-droite*.

*Modèle ergodique* : c'est un modèle sans contraintes où toutes les transitions d'un état vers les autres sont possibles (cf. Figure 2-3 ci-dessous), c'est à dire :

$$a_{i,j} > 0 \quad \forall (i, j) \in [1, N]$$

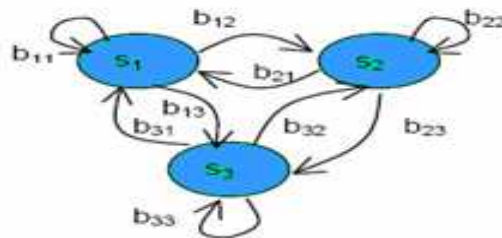


Figure 2-3 : HMM ergodique à 3 états.

*Modèle gauche droite* : C'est un modèle où il y a des contraintes sur des transitions : Seulement la transition d'un état ayant un indice bas vers un état ayant un indice haut est

acceptée (cf. Figure ci-dessous), c'est-à-dire :

$$a_{i,j} = 0 \text{ si } i > j \quad \forall (i, j) \in [1, N]$$

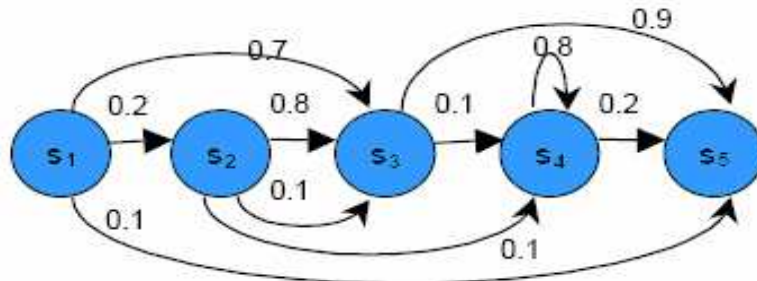


Figure 2-4 : HMM gauche droite à 4 états.

### 3.2. Types de densité d'observations

Selon la nature de la densité d'observation, nous distinguons entre trois types des HMMs. Dans un premier temps, si la distribution de probabilité d'observations est discrète ou obtenue par quantification vectorielle, les HMMs sont de type discret. Dans ce cas, cette probabilité dans chaque état de modèle sera calculée sur un ensemble discret de symboles possibles. De ce fait, les séquences d'observations générées par ce type de modèle appartiennent à un espace discret. Cependant, dans la majorité des applications, les observations sont continues d'où la nécessité d'utilisation de la quantification vectorielle pour discrétiser le signal continu.

La quantification vectorielle permet de construire un livre de code ("codebook") formé à partir de toutes les classes, chaque classe est représentée par son centre appelé vecteur de code ("codevector"). Lors de la quantification, un vecteur d'observation appartenant à l'espace continu est remplacé par le vecteur de code le plus proche au sens d'une distance donnée. Ceci permet de discrétiser l'espace d'observation puisqu'on aboutit à un espace discret formé des vecteurs de code. Cette transformation peut induire une perte d'information qui influencera les étapes ultérieures.

En second lieu, l'utilisation d'une distribution continue généralement approximée par un mélange de gaussiennes ([Huan89], [Huan90]), conduit à des HMMs continus. Ce type évite les approximations provenant du processus de quantification. Les émissions états sont modélisées par des lois paramétriques qui représentent les distributions continues de leurs probabilités. La modélisation continue nécessite une augmentation importante du nombre de paramètres à estimer.

Pratiquement  $b_j(k)$  est modélisée par un mélange de  $M$  lois gaussiennes multivariées :

$$b_j(k) = \sum_{m=1}^M C_{jm} \mathbb{N}(k, \mu_{jm}, \sigma_{jm}) \quad 1 \leq j \leq M \quad (2.7)$$

avec  $\sum_{m=1}^M C_{jm} = 1$

- $\mathbb{N}(k, \mu_{jm}, \sigma_{jm})$  : dénote une distribution gaussienne normale.
- $C_{jm}$  : le coefficient de pondération de la  $m^{\text{ème}}$  composante de mélange.
- $\mu_{jm}$  : vecteur moyen associé.
- $\sigma_{jm}$  : matrice de covariance associée.

Le dernier cas constitue un compromis entre ces deux familles, appelé HMMs semi-continus (SCHMMs : pour Semi-continuous HMMs) ([Huan90], [Rabi93]). Ce type est une alternative consistant à réduire ou à éviter la perte d'information due à la quantification des signaux continus dans le cas des HMMs discrets. Ils permettent aussi de diminuer la quantité des données ainsi que le coût calculatoire requis pour entraîner des HMMs continus [Huan93]. Dans les HMMs semi-continus, la loi de distribution continue est calculée à partir d'un mélange de densités gaussiennes (comme dans le cas continu). Mais l'ensemble des gaussiennes est partagé entre les états, ce qui permet de réduire l'ensemble des paramètres des lois d'observation. De ce fait, ce type optimise le nombre global de paramètres du modèle.

### 3.3. Problèmes fondamentaux associés aux HMMs

L'utilisation des HMMs se résume à résoudre trois problèmes de base qui sont posés par Rabiner dans [Rabi93].

- L'évaluation ou le calcul de la probabilité d'une séquence d'observations  $O(o_1, o_2, \dots, o_T)$  suivant un modèle de Markov caché  $\lambda = (A, B, \Pi)$  donné, cette probabilité est notée par  $p(O \setminus \lambda)$ .
- La recherche et le décodage de la séquence d'états de  $\lambda = (A, B, \Pi)$  la plus probable compte tenu d'une séquence d'observations  $O(o_1, o_2, \dots, o_T)$  donnée. C'est-à-dire, la séquence d'états qui a la probabilité maximale d'avoir générée  $O$ .
- Ajustement et calcul des paramètres du modèle  $\lambda = (A, B, \Pi)$  pour maximiser la vraisemblance  $p(O \setminus \lambda)$  en se basant sur un ensemble d'exemples de séquence d'observations  $O(o_1, o_2, \dots, o_T)$ .

La section suivante, présentera les solutions proposées dans la littérature pour ces trois questions.

### 3.3.1. Évaluation de la probabilité d'une séquence d'observations

Il existe plusieurs techniques d'évaluer la vraisemblance d'une séquence d'observation, parmi lesquelles on peut citer l'évaluation directe et l'évaluation par les fonctions Forward-backward [Augu01].

La question posée est : Étant donné une séquence d'observations  $O = \{o_1, o_2, \dots, o_n\}$  et un modèle  $\lambda = \{\Pi, A, B\}$ , comment peut on calculer efficacement la probabilité de la séquence  $O$  par rapport au modèle  $\lambda$ , c'est-à-dire  $p(O/\lambda)$ .

**Évaluation directe :** L'évaluation directe consiste à déterminer  $p(O/\lambda)$ , en sommant sur tous les chemins d'états possibles  $Q = \{q_1, q_2, \dots, q_N\}$  des probabilités conjointes de  $O$  et de  $Q$ , soit :

$$p(O/\lambda) = \sum_Q p(O, Q/\lambda) = \sum_Q p(O/Q, \lambda)p(Q/\lambda) \quad (2.8)$$

La probabilité d'emprunter le chemin  $Q$  et en utilisant les hypothèses Markoviennes est :

$$p(Q/\lambda) = p(s_1, s_2, \dots, s_n | \lambda) = \pi_1 \times a_{12} \times a_{23} \times \dots \times a_{n-1n} \quad (2.9)$$

Et la probabilité pour que cette séquence d'états  $Q$  émette les observations de la séquence  $O$  est :

$$p(O/Q, \lambda) = p(O/s_1, s_2, \dots, s_n, \lambda) = b_1(o_1) \times b_2(o_2) \times \dots \times b_n(o_n) \quad (2.10)$$

(Hypothèse d'indépendance des observations entre elles conditionnellement à la séquence d'états cachés  $Q$ ):

On obtient finalement :

$$p(O/\lambda) = \sum_Q \pi_1 \times a_{12} \times b_1(o_1) \times a_{23} \times b_2(o_2) \times \dots \times a_{n-1n} \times b_n(o_n) \quad (2.11)$$

Pour un modèle à  $N$  états et pour  $T$  observations, cet algorithme nécessite  $(2T-1) \cdot N^T$  multiplications et  $N^T - 1$  additions. Soit de  $2TN^T$  opérations, c'est-à-dire une complexité en  $O(N^T)$ . Ce qui est pratiquement incalculable pour des valeurs de  $T$  raisonnables. Par exemple, pour  $N=5$  états et  $T=100$  observations, il faut, approximativement,  $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$  opérations.

#### **Procédure Forward-Backward :**

Constitue une autre manière d'évaluation de la probabilité  $p(O/\lambda)$ , le principe considère que l'observation peut se faire en deux étapes suivantes :

**La variable Forward :** représente l'émission de la séquence d'observations  $O = \{o_1, o_2, \dots, o_t\}$  et la réalisation de l'état  $q_t$  au temps  $t$ , soit :

$$\alpha_t(i) = p(O, q_t = s_i / \lambda) \quad (2.11)$$

qui constitue la probabilité de générer  $O = \{o_1, o_2, \dots, o_t\}$  et de se trouver dans l'état  $q_t$  à l'instant  $t$  (cf. Figure 2-5 ci-dessous).

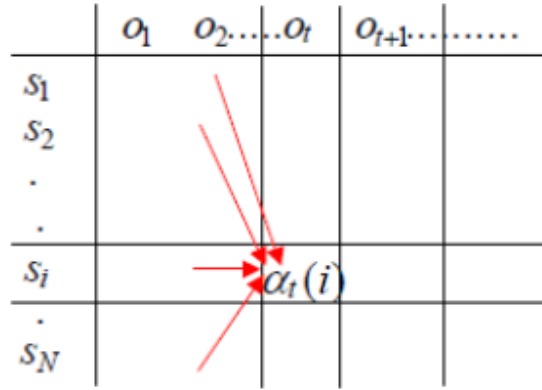


Figure 2-5 : Calcul de la variable Forward.

$$\alpha_t(i) \xrightarrow{\text{Forward}} \alpha_{t+1}(j)$$

La probabilité  $\alpha_t(i)$  est calculée de manière récursive comme suit :

**Algorithme Forward :**

Initialisation :  $\alpha_1(i) = \pi_i b_i(o_1) \quad 1 \leq i \leq N;$

Récurrence :  $\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] \times b_j(o_{t+1}) \quad t \in [1, T-1], 1 \leq j \leq N;$

Terminaison :  $p(O / \lambda) = \sum_{i=1}^N \alpha_T(i)$

La variable *Backward* : représente l'émission de la séquence d'observations

$O = \{o_{t+1}, o_{t+2}, \dots, o_T\}$  en partant de l'état  $q_t$  au temps  $t$ , soit :

$$\beta_t(i) = p(O / q_t = s_i, \lambda) \quad (2.12)$$

qui constitue la probabilité de générer  $O = \{o_{t+1}, o_{t+2}, \dots, o_T\}$  en sachant que l'on est dans l'état  $q_t$  à l'instant  $t$  (cf. Figure ci-dessous).

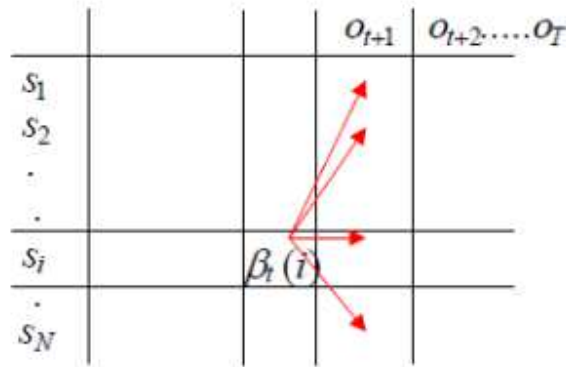


Figure 2-6 : Calcul de la variable Backward

$$\beta_t(i) \xleftarrow{\text{Backward}} \beta_{t+1}(j)$$

On déduit  $\beta_t$  de  $\beta_{t+1}$  par l'agorithme suivant :

**Algorithme Backward :**

Initialisation :  $\beta_T(i) = 1 \quad 1 \leq i \leq N$ ;

Récurrence :  $\beta_t(i) = \left[ \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \right] \quad T-1 \geq t \geq 1, 1 \leq i \leq N$ ;

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 par l'émission au début de la séquence  $O(1:t)$  et d'aboutir à  $q_t$  à l'instant  $t$ , puis, émission de la fin de l'observation  $O=(t+1:T)$  en partant de l'état  $q_t$  au temps  $t$ , connaissant  $\lambda$ . Le calcul de  $\alpha_t(i)$  se fait avec  $t$  croissant tandis que celui de  $\beta_t(i)$  se fait avec  $t$  décroissant, d'où l'expression *Forward-Backward*.

$p(O/\lambda)$  peut être défini à chaque instant  $t \in [1, T]$  par :

$$p(O/\lambda) = \sum_{i=1}^N \alpha_t(i) \times \beta_t(i) \quad (2.13)$$

Dan le cas où  $(t=0)(T=0)$  on obtient :

$$p(O/\lambda) = \sum_{i=1}^N \alpha_T(i) = \sum_{i=1}^N \pi_i \beta_0(i) \quad (2.14)$$

En totalité, le nombre d'opérations dans ces deux algorithmes, 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 de l'évaluation direct précédent ( $N=5, T=100$ ), nous



avons environ 3000 opérations à exécuter, contre  $10^{72}$  opérations par la méthode directe, ce qui permet donc un gain très important en complexité de calcul.

### 3.3.2. Analyse et décodage

Le décodage d'un modèle consiste à déterminer le chemin correspond à l'observation  $O = \{o_1, o_2, \dots, o_t\}$ , c'est-à-dire de chercher et trouver la meilleure suite d'états  $Q = \{q_1, q_2, \dots, q_t\}$  qui maximise la vraisemblance de la séquence d'observations  $O = \{o_1, o_2, \dots, o_t\}$  (cf. Figure 2-7 ci-dessous)

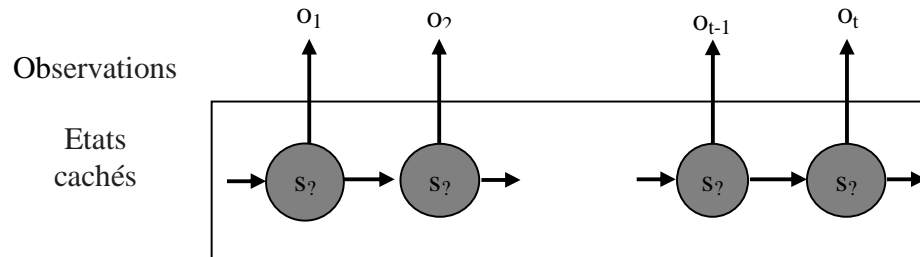


Figure 2-7 : Représentation de décodage.

Pour résoudre ce problème on a recourt à l'algorithme de Viterbi [Forn73]. Cette solution est la plus utilisée. Elle est basée sur les techniques de programmation dynamique. Viterbi est un algorithme récursif qui permet de trouver à partir d'une suite d'observations, une solution optimale au problème d'estimation de la suite d'états.

- Probabilité du meilleur début de chemin de taille  $t$  finissant par  $s_i$  est donnée par :

$$\delta_t(i) = \max_j P(o_1, o_2, \dots, o_t, s_t = s_i | \lambda) \quad (2.15)$$

- Règle d'induction

$$\delta_1(i) = \pi_i * b_i(o_1) \quad (2.16)$$

$$\delta_{t+1}(i) = \max_{j=1..n} (\delta_t(j) * a_{j,i}) * b_i(o_{t+1}) \quad (2.17)$$

On sauve aussi, à chaque  $t$ , l'état optimal qui mène à l'état  $i$  au temps  $t+1$

$$\Psi_{t+1}(i) = \arg \max_{j=1..n} (\delta_t(j) * a_{j,i}) \quad (2.18)$$

On stocke ainsi les  $n-1$  meilleurs états successifs intermédiaires

Résultat final : Prendre le chemin qui maximise  $\delta_n(i)$ .

**Algorithme de Viterbi :**Initialisation :  $t=1, 1 \leq i \leq N, \delta_1(i) = \pi_i \times b_i(o_1)$ Récurrence :  $t \in [2, N], 1 \leq i \leq N$ 

$$\delta_{t+1}(i) = \max_{j=1..n} (\delta_t(j) * a_{j,i}) * b_i(o_{t+1})$$

$$\Psi_t(i) = \arg \max_j (\delta_{t-1}(j) * a_{j,i})$$

Terminaison :  $s(N) = \arg \max_i \delta_T(i)$ Retour en arrière :  $N-1 \geq t \geq 1 \quad s(t) = \Psi_{t+1}(s(t+1))$ 

Cet algorithme est basé sur la programmation dynamique qui 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. La Figure 2-8 ci-dessous présente un schéma illustratif de cet algorithme. 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  $LN^2$ .

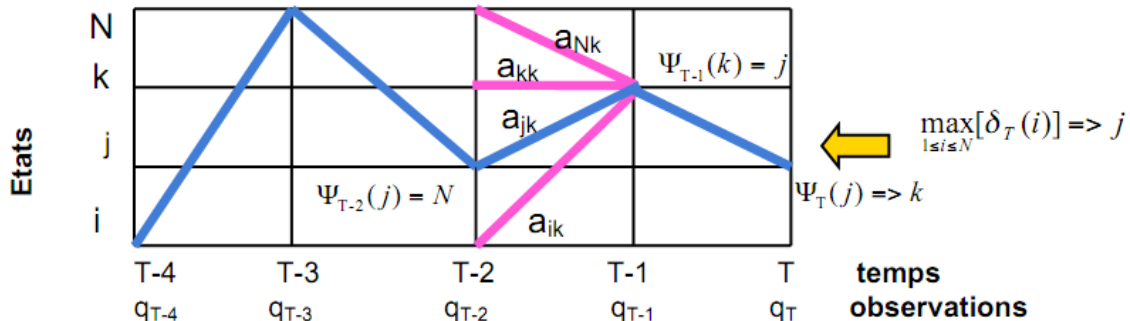


Figure 2-8 : Le retour en arrière.

**3.3.3. Apprentissage et calcul des paramètres du modèle**

L'idée de l'apprentissage est d'ajuster les paramètres du modèle  $\lambda = \{\Pi, A, B\}$  pour maximiser la vraisemblance  $p(O/\lambda)$ . Ceci se traduit par la recherche des paramètres optimaux pour le modèle  $\lambda$ , par l'un des méthodes d'entraînement des HMMs [Rabi89] qui se basent sur des critères bien définis. Un des critères d'entraînement les plus connus est celui basé sur la maximisation de vraisemblance (ML : "Maximum Likelihood")  $p(O/\lambda)$ . Cette problématique est résolue souvent par des algorithmes itératifs avec de bonnes propriétés de convergence qui sont de type Estimation- Maximisation (EM).

Ces algorithmes se basent en général sur le principe suivant :

Étant donnée une base d'apprentissage composée de séquences d'observation, les paramètres d'un HMM sont premièrement initialisés puis, re-estimés itérativement en augmentant la vraisemblance du modèle vis à vis des séquences d'observations d'entraînement. Le processus d'entraînement d'un modèle s'arrête quand la vraisemblance du modèle atteint une valeur maximale. Ceci affine le modèle, petit à petit, les étapes ci-dessous expliquent formellement et brièvement ce principe :

→ Choisir un ensemble initial de paramètres  $\lambda_0$ .

→ Calculer  $\lambda_1$  à partir de  $\lambda_0$ .

→ Répéter ce processus jusqu'à un critère de fin (stabilité).

Partant de  $\lambda_n$ ,  $\lambda_{n+1}$  doivent vérifier :

$$\prod_r p(O^r / \lambda_{n+1}) \geq \prod_r p(O^r / \lambda_n) \quad (2.19)$$

La procédure de Baum - Welch (Forward - Backward) constitue une version accélérée de l'algorithme EM pour l'apprentissage des HMMs. Elle se base aussi sur le critère de la maximisation de vraisemblance ML. Dont le détail est présenté dans ce qui suit.

On définit pour cela les fonctions  $\xi_t$  et  $\gamma_t$ .

$\xi_t(i, j)$  Probabilité que l'on passe par  $s_i$  en  $t$  et  $s_j$  en  $t+1$  en générant  $O$  avec  $\lambda$

$$\xi_t(i, j) = p(q_t = s_i, q_{t+1} = s_j | O, \lambda) \quad (2.20)$$

Par définition des fonctions *Forward* (2.11) et *Backward* (2.12), on peut écrire  $\xi_t(i, j)$  sous la forme :

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \times \beta_{t+1}(j)}{p(O / \lambda)} = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \times \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{i=1}^N \alpha_t(i) a_{ij} b_j(o_{t+1}) \times \beta_{t+1}(j)} \quad (2.21)$$

$\gamma_t(i)$  Représente la probabilité d'être dans l'état  $s_i$  à l'instant  $t$ , sachant  $O$  et  $\lambda$

Par définition de  $\xi_t(i, j)$ , on peut donc écrire :

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j) \quad (2.22)$$

Le modèle  $\lambda_1$  est calculé par les formules de ré-estimation :

$$\bar{\pi}_i = \gamma_1(i) \quad 1 \leq i \leq N \quad (2.23)$$

(2.23) : Nombre de fois que HMM s'est trouvé dans  $s_i$  à l'instant 1

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad 1 \leq i, j \leq N \quad (2.24)$$

(2.24) : représente  $\frac{\text{Nombre de fois ou la transition de } s_i \text{ à } s_j \text{ a été utilisée}}{\text{Nombre de transitions effectuées à partir de } s_i}$

$$\bar{b}_i(k) = \frac{\sum_{t=1 \cap o_t = v_k}^{T-1} \xi_t(i, j)}{\sum_{t=1}^T \gamma_t(i)} \quad 1 \leq i \leq N \quad (2.25)$$

(2.25) : représente  $\frac{\text{Nombre de fois ou le HMM s'est trouvé dans l'état } s_i \text{ en observant } o_k}{\text{Nombre de fois ou le HMM s'est trouvé dans l'état } s_i}$

#### Algorithme de Baum-Welch

Fixer les valeurs initiales :

$$a_{ij}^0, b_j^0(k), \pi_i^0 \quad 1 \leq i, j \leq N, 1 \leq k \leq N$$

Calculer à l'aide des fonctions de *Forward-Backward* :

$$\xi_t(i, j) \quad \gamma_t(i) \quad 1 \leq i, j \leq N, 1 \leq t \leq T-1$$

Et  $\bar{\lambda}$  en utilisant les formules de ré estimation.

Recommencer en 2 jusqu'à ce qu'un certain critère de convergence soit rempli.

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  $\mu_{jm}$ , matrice de covariance  $\sigma_{jm}$  et le coefficient de pondération de la m<sup>ème</sup> composante de mélange  $C_{jm}$  composante de la mixture, ou  $j$  désigne un état. Ces paramètres sont calculés par les formules :

$$(2.26)$$

$$(2.27)$$

$$\begin{aligned}
C_{jk} &= \frac{\sum_{t=1}^T \gamma_t(j,k)}{\sum_{t=1}^T \sum_{k=1}^M \gamma_t(j,k)} & \mu_{jk} &= \frac{\sum_{t=1}^T \gamma_t(j,k) O_t}{\sum_{t=1}^T \gamma_t(j,k)} \\
\sigma_{jk} &= \frac{\sum_{t=1}^T \gamma_t(j,k) (O_t - m\mu_{jm})(O_t - m\mu_{jm})'}{\sum_{t=1}^T \gamma_t(j,k)} & & (2.28)
\end{aligned}$$

**Remarques :**

- Le choix du modèle initial influe sur les résultats. Par exemple, si certaines valeurs de  $A$  et  $B$  sont égales à 0 au départ, elles le resteront jusqu'à la fin d'apprentissage. Ceci permet en particulier de garder la topologie des modèles notamment pour la structure gauche-droite qui est très utilisée en reconnaissance de l'écriture.
- L'algorithme converge vers des valeurs de paramètres qui assurent un maximum local de  $p(O/\lambda)$ . Il est donc important, si l'on veut être aussi près que possible du maximum global, de bien choisir la structure et les valeurs initiales des paramètres.
- Le nombre d'itérations est fixé empiriquement. L'expérience prouve que, si le point précédent a été correctement traité, la stabilisation des paramètres ne correspond pas à un sur-apprentissage : il n'y a donc en général pas besoin de contrôler la convergence par un ensemble de validation.
- Il existe d'autres méthodes d'apprentissage basé sur le principe «EM», parmi ces méthodes on cite l'apprentissage Viterbi : qui représente une variante modifiée de la procédure de Baum-Welsh dans laquelle on 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.

### 3.4. Modèle discriminant et chemin discriminant

L'application des modèles de Markov cachés dans le domaine de la reconnaissance de l'écriture, se base souvent sur deux stratégies principales pour la construction de HMMs: soit par l'approche de recherche du *modèle discriminant (DM-HMM)* dans le cas d'un ou plusieurs modèles par classe, soit par la technique recherche du *chemin discriminant optimal (DP-HMM)* dans le cas d'un seul modèle pour toutes les classes.

Dans la modélisation à base *DM-HMM*, la reconnaissance est généralement effectuée en estimant les probabilités d'émission de la suite d'observations  $O$  de la forme à reconnaître par les différents modèles préalablement construits. La forme à reconnaître est affectée à la classe dont le modèle qui maximise la probabilité

$$\lambda^* = \underset{\lambda \in \Lambda}{\text{Max}} P(O/\lambda) \quad (2.29)$$

où  $\Lambda$  désigne l'ensemble des modèles.

Dans le deuxième cas, la reconnaissance se réalise en décodant le meilleur chemin correspondant à l'observation, c'est-à-dire à trouver dans le modèle unique, la meilleure suite d'états, appelée *suite d'états de Viterbi*, qui maximise la quantité  $P(O/\lambda)$ . Ceci revient à trouver le meilleur chemin dans le treillis. L'algorithme de Viterbi est particulièrement bien adapté à ce type de problèmes.

La modélisation *DM-HMM* est pratiquement utilisée dans le cas où le nombre de classes à reconnaître est relativement limité (application à vocabulaire limité). Toutefois elle devient coûteuse en temps de calcul et place mémoire quand ce nombre dépasse le millier puisque chaque classe possède au moins un modèle qui lui est propre. Cependant la modélisation *DP-HMM* présente l'avantage d'être "indépendante" du nombre de classes de reconnaissance (en termes de mémoire et de vitesse) puisque le modèle est construit pour toutes les classes. L'apprentissage d'une nouvelle forme dans ce cas, nécessite seulement de recalculer les probabilités de transition du modèle, dans le cas *DM-HMM* un nouveau modèle est construit à partir d'un nombre élevé d'échantillons de la forme à apprendre, ce qui est difficile à collecter pour certains types d'applications [Chen94].

Nos contributions sont basées essentiellement sur les modèles de Markov cachés unidimensionnels (HMM 1D). En effet, nous présenterons plus tard dans les chapitres 4 et 5 deux systèmes (DM-HMM 1D et DP-HMM 1D). En effet, Tous les aspects relatifs à l'implémentation des HMMs 1D comme le choix des paramètres initiaux, le choix du type de modèle, la sélection de la topologie, le problème d'insuffisance de données pour l'apprentissage, etc. ne sont pas abordés dans ce chapitre, et nous renvoyons le lecteur aux références ([Rabi89], [Rabi93]), pour une discussion plus détaillée de ces aspects. Toutefois, certains de ces aspects sont discutés dans les chapitres consacrés à notre contribution.

L'utilisation des HMMs 1D en reconnaissance automatique de l'écriture a permis d'obtenir des résultats intéressants pour certains types d'applications. Nous renvoyons le lecteur à la section 4.1, pour un état de l'art détaillé. Cependant, ces modèles, de par leur nature unidimensionnelle, considèrent l'image d'écriture comme un signal unidimensionnel, occultant ainsi ses propriétés planaires. Dans ce qui suit, nous faisons une brève présentation des HMMs planaires.

#### 4. Modèles pseudo bidimensionnels : Planar HMM(PHMM )

L'aspect 2D de l'écriture a poussé les chercheurs à penser à des autres variantes des modèles de Markov qui peuvent être apportées des améliorations à la version unidimensionnelle. C-à-dire étendre ces modèles afin qu'ils soient applicable aux images. Les PHMMs ou Planar HMMs ont été proposés initialement par Levin [Levi92] comme

solution intermédiaire entre les approches HMM 2D conduisant à une complexité excessive (section suivante) et les approches HMM 1D occultant les propriétés planaires de l'écriture. En pratique, un HMM planaire est une extension des HMM 1D dont les probabilités d'émission du HMM principal sont remplacées par la vraisemblance d'un deuxième HMM. En général, le HMM principal décrit l'image verticalement zone par zone, tandis que les HMM secondaires décrivent une zone site par site. Ces sites sont un voisinage de pixels qui doivent être à la fois suffisamment grands pour permettre des mesures, et suffisamment petits pour s'attacher à extraire une caractéristique locale. Les états du modèle principal sont appelés super-états (cf. Figure ci-dessous).

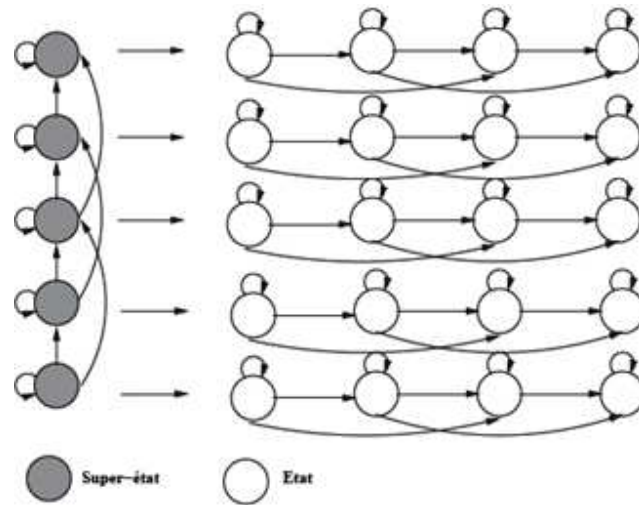


Figure 2-9 : Exemple d'architecture d'un PHMM, d'après [Bela97].

Un PHMM donné par  $\eta = (A, \Pi, \Lambda)$  est défini de la manière suivante :

$S = \{s_1, \dots, s_N\}$  l'ensemble de  $N$  super états de modèle. On désigne un super-état pour la ligne  $y$  par  $q_y \in S$ .

$V = \{v_1, \dots, v_M\}$  l'ensemble discret des  $M$  symboles. On désigne un symbole à la position  $(x, y)$  par  $O_{xy} \in V$ .

$$A = \{a_{i,j}\}_{1 \leq i,j \leq N} \quad \text{ou} \\ a_{i,j} = P(q_y = S_j | q_{y-1} = S_i) \quad (2.30)$$

la matrice des transitions entre super états.

$$\Pi = \{\pi_i\}_{1 \leq i \leq N} \quad \text{ou} \\ \pi_i = P(q_1 = s_i) \quad (2.31)$$

$\Pi$  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.

$\lambda^k = \{\Pi^k, A^k, B^k\}$ , avec les paramètres :

$S^k = \{s_1^k, \dots, s_{N^k}^k\}$  l'ensemble des  $N^k$  états du modèle. l'état localisé en  $(x, y)$  noté par

$q_{xy} \in S^k$ .

$A^k = \{a_{i,j}^k\}_{1 \leq i, j \leq N^k}$  ou  $a_{i,j}^k = P(q_{x+1y} = s_j^k | q_{xy} = s_i^k)$  Représente la matrice de transitions entre états.

$B^k = \{b_j^k(l)\}_{1 \leq j \leq N^k; 1 \leq l \leq M}$ , ou  $b_j^k(l) = P(o_{xy} = o_l | q_{xy} = s_j^k)$ .

$\Pi^k = \{\pi_i^k\}_{1 \leq i \leq N^k}$ , ou  $\pi_i^k = P(q_{1y} = s_i^k)$ .

Dans les systèmes de reconnaissance de l'écriture à base PHMM, le modèle secondaire est souvent associé aux bandes horizontales (blocs de lignes) ou 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 spécifiées de la graphie traité particulièrement de la morphologie de la forme et des principales zones horizontales d'observation que l'on veut mettre en évidence.

#### 4.1. Procédure d'apprentissage des PHMMs

L'apprentissage des PHMMs se ressemble beaucoup l'apprentissage des HMM 1D, en réalité les PHMMs étant à la base constitués de HMMs, donc l'estimation des paramètres des modèles secondaires est effectuée, le plus souvent par l'algorithme de Baum-Welch ou la variante de Viterbi. Enfin les paramètres du modèle principal sont généralement estimés par comptage.

#### 4.2. Procédure de reconnaissance des PHMMs

La reconnaissance au niveau d'un PHMM  $\lambda$  et une observation  $O$ , s'effectue par la recherche des super-états maximisant  $P(O|\lambda)$ . Ceci est mis en œuvre par un algorithme de Viterbi doublement intégré [Kuo94]. La première phase de cet Algorithme sert à déterminer le meilleur chemin sur la bande horizontale (une ou plusieurs ligne). La deuxième permet de calculer la probabilité d'associer une bande (une ou plusieurs lignes) à un super-état.

L'application du formalisme PHMMs à la reconnaissance de l'écriture sera donnée dans la section 4.2.

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.

## 5. Modèles de Markov bidimensionnels : Champs de Markov (HMM2D)

Les champs de Markov ont été introduits dans le domaine de l'analyse d'images en 1984 avec les travaux de Geman [Gema84]. Contrairement aux PHMM, les champs de Markov présentent une vraie structure qui modélise l'aspect 2D des images. Du fait de la dépendance bidimensionnelle locale des sites au sein d'un voisinage. Ainsi, la probabilité de la réalisation d'une variable aléatoire connaissant toutes les réalisations des autres variables aléatoires ne dépend que du voisinage fixé.

Leurs domaines d'application est très vaste, ils sont utilisés avec succès par exemple, dans la restauration et la segmentation des images et des documents multi structurelles, détection des objets et des contours, la reconstruction de surfaces, l'analyse de textures, la fusion de données, etc. En revanche ils sont moins utilisés en reconnaissance de l'écriture manuscrite, même s'ils sont de plus en plus étudiés.

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  $\Omega = L^{|S|}$  ou  $|S|$  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 :

$$\forall x \in q^{|S|} = P(X = x) \geq 0 \quad (2.33)$$

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

– propriété de Markov :

$$\forall s \in S \text{ et } \forall x \in q^{|S|} \\ P(X_s = x_s | X_r = x_r, r \in S - s) = P(X_s = x_s | X_r = x_r, r \in V_s) \quad (2.34)$$

Cette propriété implique que l'état du champ en tout site ne dépend que des états du champ sur les sites voisins. La Figure 2-10 ci-dessous illustre quelques systèmes de voisinage.

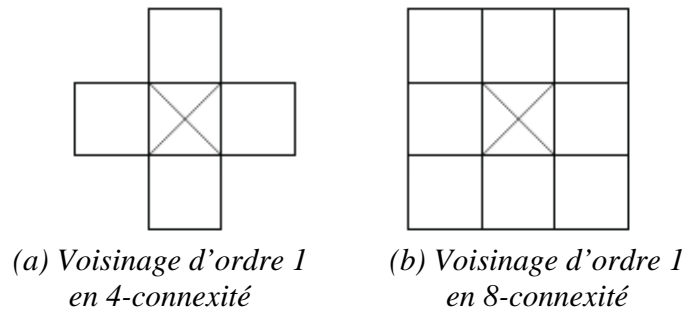


Figure 2-10 : Exemple de voisinage

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) [Park96] et les champs de Markov unilatéraux (Non-Symmetric Half-Plane Markov Chains ou NSHP) [Saon97].

Après cette introduction rapide des champs aléatoires, nous présentons leur application à la reconnaissance de l'écriture dans la section 4.3.

## 6. Domaines d'application des Mdèles de Markov cachés

Les Modèles de Markov cachés comme nous l'avons présentées ci-dessus sont des outils statistiques très puissants. Ils sont de ce fait très utilisés dans de nombreuses applications, à savoir :

- la reconnaissance automatique de l'écriture ([Dupr03] [Beno07] [Mena08] [El-Ha09]) ;
- la reconnaissance de la parole [Juan91];

- la biologiques et la biomédicale (décryptage des séquences d'ADN) [Bald95];
- le traitement automatique du langage naturel [Guyo95];
- le traitement et l'analyse d'image [Gema84] ;
- la reconnaissance de forme;
- le traitement du signal ;
- l'économie et modélisation des finances ;
- la robotique ;
- le diagnostic et vérification de signature ;

## 7. Conclusion

Nous avons dressé tout au long de ce chapitre un panorama sur les différentes spécificités des modèles de Markov cachés. 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) ou le décodage (viterbi). Ces algorithmes relativement faciles à mettre en œuvre nous serviront de référence pour notre travail.

Nous avons mis l'accent sur les modèles Markoviens PHMM et HMM 2D qui sont une tentative de modéliser la nature 2D de l'information véhiculée par les images de l'écriture sans avoir besoin de la transformation classique en données séquentielles analysées par les HMM 1D. Enfin leurs champs d'application.

Nous présentons dans le chapitre suivant un état l'art détaillé de l'application des modèles de Markov cachés en reconnaissance automatique de l'écriture hors-ligne, nous mettrons l'accent sur les différentes approches de reconnaissance markoviennes les plus répandues dans ce domaine avec leurs particularités, limitations et avantages. Cette description se poursuit par la réalisation d'une étude comparative de différents systèmes proposés dans la littérature sur un ensemble de base de données standards. Nous dressons l'évaluation et l'analyse des résultats obtenus par ces différents systèmes au cours des différentes expériences réalisées sous forme des tableaux récapitulatifs, afin de mettre en évidence les points forts et les faiblisse de chaque système.

# Chapitre 3 : La Reconnaissance d'écriture par les Modèles de Markov cachés

---

## 1. Introduction

Depuis deux décennies, les modèles de Markov sont devenus des outils incontournables et incontestables dans la reconnaissance de l'écriture. Ceci est due à quatre raisons fondamentales : (1) leur capacité d'intégration du contexte, (2) leur disposition d'absorption du bruit qui entache souvent le signal d'écriture, (3) leur grande utilisation massive dans la reconnaissance automatique de la parole (RAP) et (4) offrent des algorithmes puissants au niveau de l'apprentissage et de la reconnaissance. En plus, un certain nombre de bibliothèques HTK, GHMM et ESMERALDA et de modules sont également publiquement accessibles pour l'apprentissage et le décodage de Modèles de Markov Cachés [Youn93].

Ce qui nous intéresse plus particulièrement dans ce chapitre ce sont les applications liées à la reconnaissance de l'écriture. Dans ce sens, nous présenterons plusieurs systèmes et plusieurs approches basées sur les HMMs, afin d'étudier leur apport dans ce domaine.

Le plan de ce chapitre sera composé en parties principales ci-dessous.

Nous introduisons, analysons et discutons en premier lieu, les différentes étapes principales pilotant un système markovien. Mais, avant d'aller plus loin dans la description de différents systèmes existants et des approches utilisées, il nous paraît important de souligner un passage obligé à travers des bases de données de références dans le domaine. Ces différentes bases de données sont utilisées pour évaluer et valider les performances des systèmes, souvent appelées Benchmark [Beno07]. Dans ce contexte, nous évoquons quelques bases de données internationales pour le script (Arabe, Latin et Amazigh). En suite, nous présentons d'une manière détaillée les approches et les systèmes à base des HMMs qui ont été proposés dans la littérature sur le problème de la reconnaissance de l'écriture. Ces techniques seront catégorisées selon la technique de modélisation markovien unidimensionnel (MMC 1D), planaire (PHMMs) et bidimensionnel (MMC 2D), et aussi selon la stratégie de reconnaissance analytique et globale.

## 2. Architecture de référence des systèmes à base HMM

Généralement, les architectures des systèmes de reconnaissance de l'écriture possèdent une structure commune comme nous l'avons déjà présentée et détaillée dans le premier chapitre. Bien que cette architecture soit commune aux différents systèmes, nous noterons

qu'il existe des variations spécifiques à chaque approche. Dans cette section, nous décrirons les étapes principales de la structure spécifique aux systèmes Markoviens et les différentes techniques largement utilisées dans chaque phase.

Les systèmes de reconnaissance de l'écriture Markoviens nécessitent tous une première phase de la numérisation de document à reconnaître par un scanner ou une caméra numérique (cf. section 3.2). De ce fait, souvent l'image capturée est formée par des zones hétérogènes (textes, logos, tableaux, figures), dans le cas des documents multi structuraux. Le traitement de ce type passe par l'extraction et la localisation des zones de textes (paragraphe) à l'aide d'un ensemble techniques de séparation texte/graphiques (ascendantes et descendantes) proposées dans la littérature. Dès que les blocs de textes ont été localisés et extraits à partir de l'image. Ceux-ci peuvent encore être segmentés en lignes de texte ou en mots. Par la suite, on suppose qu'on dispose des images de mots ou de caractères. Ces dernières sont ensuite soumis à un certain nombre d'opérations pendant l'étape de prétraitement et de normalisation qui visent, d'une part, à réduire les variabilités de l'écriture [Fuji07], et d'autre part, à supprimer ou au moins à réduire le bruit dans l'image ( la binarisation, le lissage...)[Trie95]. Cette phase est décrite succinctement dans la section 3.3.

De ce fait, l'architecture d'un système markovien de reconnaissance de l'écriture (caractères ou de mots isolés) est constituée essentiellement de trois grandes phases (cf. Figure 3-1 ci-dessous) : la segmentation, l'extraction des primitives et la modélisation markovienne.



Figure 3-1 : Synopsis de système de reconnaissance à base HMM de l'écriture

## 2.1. Segmentation

La segmentation directe de l'image mots en petites unités de base (lettres ou graphèmes) est communément admise comme une tâche extrêmement difficile à mettre en œuvre le dilemme de Sayre (cf. section 3.4). Elle influence fortement les résultats de la classification [Mori02]. De ce fait, il semble intéressant de réussir cette tâche. Pour se faire, plusieurs techniques basées sur Les Modèles de Markov Cachés ont été proposées. Les plus performantes sont celles à base de la programmation dynamique, qui utilisent l'algorithme de *Viterbi* ([Ball06], [Lori06]). Par conséquent, elles cherchent la meilleure segmentation (chemin optimal) qui correspond généralement au mot qu'a été reconnu, ou bien dans d'autres cas un choix est fait entre un ensemble de solutions proposées. La meilleure solution est retenue et validée sur le modèle globale de reconnaissance ([Brit01], [Gran03], [Kund07]). Bien que la combinaison de ce type de segmentation et les HMMs ont donné de bons résultats dans la tâche de reconnaissance de manuscrits. Malgré les différentes stratégies qu'ont été développées pour réduire la dépendance de reconnaissance par rapport la segmentation. Cette stratégie aborde directement un problème qui semble impossible à

résoudre sans l'exploitation de l'un des avantages majeurs des HMMs. Ce dernier permet d'intégrer la classification et la segmentation dans un seul module ([Pech03], [El-Ha05]). Ce type est connu sous la segmentation implicite.

La segmentation implicite est effectuée par un modèle globale HMM formé par l'ensemble des modèles HMMs de caractères comportant des états représentant des entités de base (cf. Figure ci-dessous). Le principe consiste à chercher dans l'espace globale des états le chemin d'états le plus probable qui correspond à la segmentation désirée et qui sera déterminée implicitement pendant la phase de classification par l'algorithme de Viterbi.

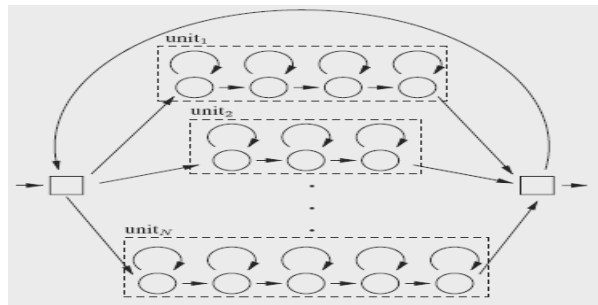


Figure 3-2 : Modèle globale comportant les modèles de caractères

D'après la littérature, la technique standard pour les systèmes à base les HMMs et la segmentation implicite est celle qui se base sur les fenêtres glissantes (cf. section 3.4.2). Le vecteur de primitives engendré par cette technique sera analysé parallèlement par la classification et la segmentation ([El-Ha09], [Brak04], [Vinc04]). La segmentation implicite à l'aide des HMMs à beaucoup d'avantages. Par exemple, récemment dans [Su07], Su et al ont signalé l'importance de celle-ci pour la reconnaissance de l'écriture. En plus, la préparation d'une base de lettres segmentées et étiquetées à partir des documents, des textes ou des mots par la segmentation directe nécessite toujours une intervention humaine. Une telle solution manuelle s'avère coûteuse et pas toujours fiable. Dans le cas des méthodes à base de MMCs, le processus de segmentation implicite permet d'éviter l'étiquetage manuel.

## 2.2. Extraction de primitives

Les données analysées par les systèmes à base des HMMs doivent être séquentielles ou temporellement ordonnés. D'où la nécessité de convertir l'aspect bidimensionnel 2D de l'image de mots ou de caractères en représentation séquentielles. De nombreuses techniques pour franchir ce passage (2D->1D) existent dans la littérature. Cependant, la méthode la plus largement utilisée pour les approches markoviennes est celle basée sur les fenêtres glissantes ([El-Ha06], [Plöt08], [Vinc04], [Mart01], [Scha05], [Beno06a], [Nata08]), (cf. Figure 3-3-a ci-dessous), appelée aussi bandes. Elle est utilisée pour la première fois par les chercheurs de Centre de Daimler-Benz pour la tâche de reconnaissance de l'écriture manuscrite [Kalt93].

Le principe de cette méthode consiste à déplacer une fenêtre latéralement dans le sens de l'écriture sur l'image de mot ou de caractère en extrayant des primitives qui peuvent être

statistiques ou structurelles. Les premières sont calculées directement à partir des intensités des pixels ou à partir de leur densité de distributions, par exemple la moyenne et la médian qui sont largement utilisées ([Mart01], [Vinc04], [El-Ab07]) ou après l'utilisation de certaines transformations globales (DCT, FFT, Radon, Ondelettes) ([Coet04], [Boug09]). Concernant les primitives structurelles, elles représentent généralement une description des propriétés géométriques et topologiques de la forme, comme les boucles, les ascendants et les descendants [Beno06a], les informations directionnelles [Touj07] et quelques primitives de concavités [El-Ha05].

Récemment les travaux ([El-Ha05], [El-Ha06], [Beno06a], [Beno06b], [Wien05]), s'orientent vers des approches hybrides qui combinent les primitives statistiques et structurelles, qui visent à exploiter les avantages de deux types.

Ces primitives sont organisées sous forme de sous séquences, qui seront combinées dans un vecteur de caractéristiques transformé par la suite en séquence d'observations au sens Markovien. Les fenêtres glissantes sont considérées communément comme une méthode robuste et quasi standard pour les systèmes markoviens, qui existent en d'autres alternatives selon les dimensions de la fenêtre (uniforme ou on-uniforme Figure 3-3-b ci-dessous), leurs chevauchements (0% 50%) et dans certains cas l'angle d'inclinaison [El-Ha06].

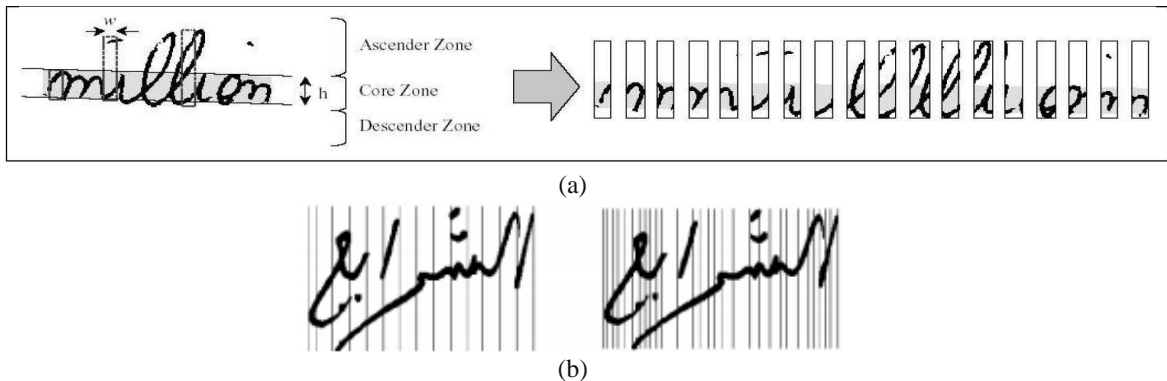


Figure 3-3: (a) :Techniques de fenêtres glissantes [Tay01], (b) : fenêtres uniformes et non uniformes [Beno07]

### 2.3. Modélisation Markovienne

La modélisation de l'écriture par les HMMs se ramène généralement à la transformation de la forme étudiée en une séquence de primitives convenablement choisies, traduisant la forme initiale en un signal unidimensionnel composé d'une suite d'observations destinée à être traitée par un modèle de Markov. En conséquence, le problème d'analyse d'images d'écriture se ramène à un problème d'analyse d'une chaîne d'observations. Les observations utilisées dans le contexte de la RAE sont en général indentifiables aux lettres

ou à des portions de celles-ci, mais elles peuvent aussi correspondre aux colonnes de pixels de l'image.

De ce fait, dans le cas des mots, la démarche la plus évidente, repose sur l'utilisation des modèles des caractères qui les composent. Par conséquent, le modèle de mot sera représenté par la suite par la concaténation des modèles de leurs lettres. Par exemple la reconnaissance des mots arabes nécessite un grand nombre de modèles de lettres. L'alphabet arabe comporte 28 lettres, la forme des lettres dépend de leur position (début/milieu et fin/isolé) dans le mot. Certaines lettres prennent jusqu'à 4 formes différentes [Lori06]. Cependant la reconnaissance de mots latins nécessite en totalité ( $26 \times 2$ ) modèles pour les majuscules et les minuscules ([Wien05], [Nata08]).

La modélisation de l'écriture uniquement aux niveaux des caractères, peut faiblir ou limiter les performances des systèmes. Par conséquent, la recherche d'autres manières de modélisation est une question de premier ordre. Certaines approches utilisant aussi des sous parties des lettres, appelées encore graphèmes comme des entités de base de la modélisation ([Touj07], [Gran03], [Xue02]). Ces graphèmes sont combinés à l'aide de l'alignement dynamique (chemin discriminant dans le modèle globale) pour former les modèles de lettres [Scha05] en se basant sur l'algorithme de *Viterbi*.

L'avantage majeur de cette modélisation c'est qu'elle augmente l'indépendance à la grande variabilité de l'écriture, donc une amélioration de la robustesse des systèmes.

L'aspect séquentiel de l'écriture conduit évidemment à l'utilisation des topologies linéaires pour les modèles HMM dans la majorité des systèmes.

La topologie linéaire (cf. Figure 3-4-a ci-dessous) se caractérise par des états qui sont reliés à eux et à leurs successeurs droits pour le latin ou gauches dans le cas de la graphie arabe. En outre, pour permettre plus de variabilité dans les longueurs de segments de certains modèles de base d'autres topologies ont été inventées, comme la topologie de Bakis [Mart01], qui est linéaire avec un saut immédiat de son successeur (cf. Figure 3-4-b ci-dessous). Quelques systèmes à base des deux topologies ont été présentés dans ([El-Ha05], [El-Ha06], [Beno06a], [Beno06b], [Wien05], [Nata08], [Plöt08], [Mart01], [Scha03]).



Figure 3-4 : Topologies (a) linéaire, (b) Bakis

Pratiquement le nombre d'états par modèle de caractère se détermine on se basant sur des calculs heuristiques ou parfois empiriquement. Cette notion est souvent associée à des unités de base (lettres, graphèmes, bandes verticales) selon l'application donnée. Certains travaux sont dédiés à la recherche et l'optimisation de la topologie pour la modélisation de l'écriture comme ([Augu01], [Zimm02], [Günt03]) qui permettent de booster les taux de reconnaissance. D'autres cherchent à modéliser la durée que le modèle peut passer sur un état comme la contribution proposée par Benouareth et al ([Beno06a], [Beno06b]). Il à



montré que la modélisation de cette durée par la distribution Gama donne de bons résultats.

Le problème fondamental de la modélisation markovienne réside dans la façon de gérer les données d'entrée, c.-à-d. il s'agit de comment les faire correspondre aux sorties générées par les HMMs. De nombreux travaux utilisent différentes techniques pour la modélisation des distributions d'émissions, nous citons ci-dessous les plus utilisées :

- Combinaison des HMMs discrets et la quantification vectorielle ([Feng02] [Beno06a], [Kund07]). Dans ce cas, un codage de données en un ensemble fini de symboles discrets est effectué, qui est simple mais elle est très coûteuse en terme de mémoire et de temps;
- HMMs semi continu (continu et discrètes) [Xue06] ;
- HMMs discrète et réseaux de neurones qui sont utilisés directement pour calculer les probabilités d'émissions [Mena08];
- HMMs continus qui décrivent les probabilités d'émission par des mélanges de Gaussiennes ([Mart01], [Vinc04], [El-Ha05], [Su07]).

Le grand succès des approches markoviennes est justifié par l'existence des algorithmes efficaces pour l'estimation et le décodage des paramètres modèles [Rabi89]. De ce fait, l'apprentissage des modèles par exemple s'effectue souvent par la variante *Baum-Welsh* de la classe des procédures Expectation maximisation « EM ». La procédure réestime les paramètres de modèle initiale de telle sorte à avoir un deuxième modèle qui améliore la vraisemblance :

$$P(O|\lambda_1) \geq P(O|\lambda_0)$$

$\lambda_1$  désigne le modèle adapté obtenu à partir de  $\lambda_0$  en appliquant les formules de ré-estimation.

Le modèle d'apprentissage est itéré jusqu'à certaine stabilité ( $P(O|\lambda_n) - P(O|\lambda_{n-1}) \leq \epsilon$ ).

Quant au décodage où la classification est communément réalisée par la procédure de *Viterbi* qui consiste à déterminer la séquence  $s^*$  des états qui maximise  $P(O|\lambda)$ .

### 3. Bases de données de référence (Benchmark)

L'évaluation des systèmes de reconnaissance de l'écriture est généralement réalisée à l'aide d'expériences pratiques sur des corpus de données. De ce fait, l'estimation des paramètres de modèles statistiques choisis (HMM1D, PHMM ou HMM 2D) est effectuée sur les données d'apprentissage ou d'entraînement (annotées) et les performances de reconnaissance sont mesurées sur l'ensemble de données de tests (base de tests). Ci-dessous une description brève de quelques bases de données les plus fréquemment utilisées.

### 3.1. IFN/ENIT

Cette base de données est développée par l'institut des technologies de communication (IfN) en coopération avec l'école nationale d'ingénieurs de Tunis (ENIT) en 2002 [Pech03]. Il s'agit d'une base de données d'images de noms de villes tunisiennes. 411 scripteurs de différents âges et professions, ont été priés de remplir des formulaires avec les noms de villes/villages tunisiennes et les codes postaux correspondants, pour collecter environ 26400 noms de villes (dans un lexique de 937 villes) et plus de 210000 caractères.

Dans cette base, l'annotation des données est effectuée automatiquement et la vérification est faite manuellement. Les fichiers d'annotations contiennent des informations sur l'image, le nom de ville/village en code ASCII, la séquence détaillée des caractères, la position de la ligne de base, le nombre de mots, caractères et pseudo-mots, et enfin des informations sur la qualité de la base de ligne et de l'écriture (un exemple de cet annotation est présente dans le Tableau 3-1 ci-dessous).


Image	
Annotation :	
code postal	3070
nom en ASCII	قرقنة
séquence des caractères	اEتMانBاقاEراBق
ligne de base y1, y2	77,83
qualité de la ligne de base	B1
nombre de mots	1
nombre de pseudo-mots	2
nombre de caractères	5
qualité de l'écriture	W1

Tableau 3-1 : Un exemple d'image de la base de données IfN/ENIT. Les symboles M, B, A, E présentent la position des caractères dans les mots (au milieu, au début, seul, ou à la fin d'un mot).

Grâce à sa taille et à sa disponibilité gratuite, la base IFN/ENIT s'est imposée comme la base de données de référence pour la comparaison des performances des systèmes de reconnaissance de l'écriture arabe manuscrite. Elle est également un support à l'organisation de la compétition internationale : ICDAR Arabic Handwritten Competition, 2005, 2007 ([Marg05] [Marg07]).

### 3.2. AMHCD

C'est une base de données de caractères Amazighes qui comprend 25740 caractères amazighes manuscrit, isolés et étiquetés produite par 60 scripteurs. Cette base de données a été développée par Es saady au Laboratoire IRF-SIC de l'Université Ibn Zohr d'Agadir. Elle est conçue pour l'apprentissage et le test des systèmes de reconnaissance de caractères Amazighes manuscrits. Cette base de données est disponible pour les recherches et les

usages académiques [Essa11]. Cette base sera utilisée pour tester et évaluer les performances de notre première approche dans le chapitre 4.

### **3.3. Base des patterns de la graphie amazighe**

Est une base des patterns de différentes fontes amazighes et de tailles variées [Aito09]. Elle contient au total 12 polices de caractères et les tailles du 10 points au 28 points pour chaque modèle. Les patterns sont fournis sous forme d'images bitonales de tailles variables. La taille maximale est de 102×129 pixels, tandis que la taille minimale est de 19×2 pixels. Une telle disparité s'explique par le fait que le caractère ya (a) est un petit cercle, et est donc beaucoup plus petit que les autres caractères. Outre le cas particulier du caractère ya (a), la base est constituée des patterns de différentes fontes amazighes et de tailles variées, qui ne sont pas normalisées. Dans le chapitre 5, nous présenterons un système qui sera testé et validé sur les données de cette base.

### **3.4. MNIST**

C'est une base gratuite proposée par [Lecu98], elle est obtenue à partir de la base NIST (National Institute of Standards and Technology) « *NIST composée d'images extraites à partir des formulaires de recensement. Elle contient à la fois des caractères isolés et des textes écrits par 3.600 scripteurs pour un total de 80.000 d'images* ». Cependant, MNIST se constitue uniquement de chiffres manuscrites codés sur 8 bits et prétraités (taille fixe, caractères normalisés, caractères centrés). Elle comporte en totalité 60 000 images pour l'apprentissage, et 10 000 images pour la phase de test.

### **3.5. IAM-DB**

Cette base est composée d'un ensemble de phrases manuscrites en anglais [Mart02]. Elle est basée sur le corpus de données LOB qui regroupe environ une collection de texte formée d'un million de mots. Dans sa version 3.0, la base de données comprend des images de 1539 formulaires qui ont été produits par 657 scripteurs, dont les résultats dans un total de plus de 11500 images de mots. Cette base existe sous la forme en ligne qui n'est pas accessible au public, il s'agit de la base IAM-OnDBCam ([Liw05], [Liw05a]), elle se compose d'images de textes en couleur de tableau blanc qui ont été prises avec un appareil photo numérique avec une résolution de 3264 × 2448 pixels. La base de données contient 491 documents écrits par 62 scripteurs, sans contraintes (le style d'écriture). Le texte écrit sur le tableau blanc est basé aussi sur le corpus LOB. En total, la base de données comprend un dictionnaire de 11059 mots.

### **3.6. Cambridge**

Encore appelée base Senior & Robinson, elle contient des documents manuscrits de Cambridge, qui sont regroupés en 353 lignes de texte manuscrit qui sont divisées en données

d'apprentissage (153 lignes), données de validation (83 lignes), et les données de test (117 lignes) [Seni98]. Après la décomposition en mots, la base se constitue globalement 2360 d'images de mots d'apprentissage, 675 d'images de mots de validation, et 1016 mots de test. En totalité, il se compose de 1334 mots.

La majorité de ces base de données ont été considérées et utilisées comme des supports de référence pour évaluer les performances des systèmes présentés dans ce qui suit.

#### **4. Systèmes et Approches antérieurement proposées**

Dans ce qui suit nous allons présenter quelques travaux en RAE utilisant des HMMs 1D, PHMMs et HMMs 2D. Malgré que la majorité des travaux adaptent la modélisation markovienne qui s'adapte bien à la nature d'un signal d'écriture, l'approche peut être différente d'une contribution à une autre en fonction de plusieurs aspects suivants :

La stratégie de reconnaissance (modèle discriminant ou chemin discriminant); la méthode de segmentation (analytique ou globale); le sens donné aux états de modèle; la nature de l'entité modélisée (graphème, lettre, mot); la topologie du modèle de l'entité de base (gauche, droite ou ergodique) et le type de la densité d'observation (discrète, continue, semi continue);

Nous présentons dans cette partie les différents systèmes et approches à base des HMM 1D, HMMs planaires et des HMMs2D antérieurement proposées en les classifiant par le mode de reconnaissance globale et analytique.

##### **4.1. Les modèles de Markov cachés unidimensionnel (HMM 1D)**

Aujourd'hui, les HMMs 1D sont devenus les outils prédominants en reconnaissance automatique de l'écriture (RAE). Dans la littérature de nombreux travaux basés sur les HMM 1D sont utilisés avec succès dans ce domaine ([Chen94], [Gill96], [El-Ya99], [Masm00], [Augu01], [Pech02], [Gran03], [Dupr03], [Fara06], [Pech06], [Beno07], [El-Ab07], [Naga07], [Mena08], [El-Ha09]). Ceci grâce à leur capacité d'intégration du contexte et d'absorption du bruit qui entache souvent le signal d'écrit. Nous présenterons alors dans cette partie, des approches markoviennes 1D globales et locales.

###### **4.1.1. Approches globales**

Dans [Boug09], Bougamouza et al proposent une approche basée sur les HMMs 1D pour la reconnaissance globale des noms de villes tunisiennes, pour cela ils ont procédé à effectuer les démarches décrites ci-dessous :

- Prétraitements de l'image
  - Suppression des allongements horizontaux, des espaces verticaux et des espaces horizontaux.
  - Transformation de l'image en signal 1D.
- Extraction de primitives
  - Primitives basées sur les coefficients spectraux (MFCC) de la transformation de Fourier.
  - Chaque vecteur de caractéristiques contient 42 coefficients.

- La modélisation
  - Un modèle de Markov caché (HMM1D) gauche droite est utilisé pour modéliser chaque mot.
  - Le nombre d'états par HMM1D est égale à 17 inclus les deux états non émetteurs de début et de fin (fixé empiriquement).
  - L'apprentissage et la reconnaissance sont effectués par viterbi.
- Les résultats
  - Les tests ont été effectués sur la base IFN/ENIT : Apprentissage (6823 images), tests (2650 images), dictionnaire (59 noms villes).
  - Scores : 70.78% sans utilisation des prétraitements.
    - : 85.60% avec utilisation des prétraitements.
    - : 96.05% en Top 5 (les 5 meilleurs candidats incluent le mot approprié).

Dans [Al-Ma02b], Al-Ma'adeed et al ont proposé une méthode basée sur HMM1D à densité d'observations discrets de type modèle discriminant pour la reconnaissance des montants littéraux manuscrits et des chèques postaux. Elle comporte les principales étapes suivantes :

- Prétraitements et normalisation
  - Correction de l'inclinaison locale et globale de mots.
  - Normalisation d'épaisseur et de la hauteur des lettres.
- Extraction de primitives
  - Utilisation de technique de fenêtrage.
  - Chaque fenêtre est transformée en un vecteur de 29 caractéristiques.
  - La quantification vectorielle est utilisée pour construire un dictionnaire (codebook) de taille qu'a été empiriquement fixée à 70.
- Modélisation
  - Le mot est modélisé par un HMM1D dont le nombre d'états correspond aux caractères qui le composent.
  - Le nombre total des lettres (ou portions de lettres) est 55.
  - La classification est effectuée par une version modifiée de l'algorithme de Viterbi.
  - L'apprentissage est réalisé par l'algorithme de Baum-Welch.
- Les résultats
  - Les expériences ont été effectuées sur la base de données AHDB créée par les mêmes auteurs (contenant plus de 15 000 échantillons produits par 100 scripteurs).
  - 2/3 de l'ensemble de données c.-à-d. 10 000 images pour l'apprentissage, et le reste pour les tests.
  - Le taux de reconnaissance obtenu en top 1 environ de 45%.

Dans le but d'étendre l'approche précitée, les auteurs présentent aussi dans [Al-Ma04], une amélioration au niveau de la classification. En effet, cette extension comporte les étapes ci dessous :

- Modélisation
  - Combinaison séquentielle d'un classifieur fondé sur des règles heuristiques avec plusieurs classifieurs HMM1D fonctionnant en parallèle.
  - Chaque classifieur HMM1D consiste à discriminer une partie de mots du lexique (8 groupes de classes de mots).
  - Le classifieur heuristique effectue une discrimination globale du mot en se basant sur des primitives visuelles (nombre de points hauts, nombre de points bas et nombre de segments).
  - La reconnaissance est effectuée en deux phases, en premier lieu, le classifieur détermine le groupe auquel le mot appartient, puis le classifieur HMM1D associé à ce groupe est activé pour identifier le mot.
- Les résultats
  - Le corpus de données d'apprentissage 4200 images de mots extraites de la base AHDB, le reste représente les données de test.
  - un score de reconnaissance de 65% en top 1.

Dans [Dehg01a], Dehghan et al mettent au point un système basé sur une approche globale pour la reconnaissance des manuscrits Arabes/Farsi, le principe de cette approche est le suivant :

- Extraction de primitives
  - Primitives calculées à partir du contour de l'image par le principe de fenêtres glissantes verticales.
  - Les fenêtres ont toutes une largeur fixe avec la possibilité de chevauchement de 50% entre 2 bandes consécutives.
  - Les fenêtres se déplacent dans le sens de l'écriture arabe (de droite à gauche) (cf. Figure 3-5 ci-dessous).
  - Chaque bande est découpée horizontalement en 5 zones de même hauteur.
  - Chaque zone horizontale est transformée en un vecteur de primitives de taille 20.
  - Utilisation de la quantification vectorielle basée sur les cartes auto-organisatrices de Kohonen pour réduire la taille de la séquence d'observations obtenue à partir de vecteur de primitives.
- Modélisation
  - Le mot est considéré comme l'entité de base, puis modélisé par un HMM 1D discret.
  - Apprentissage est effectué par la procédure itérative de Baum-Welsh.
- Les résultats
  - Base de données est de taille supérieure à 17,000 exemples.
  - Base d'apprentissage est de 60% de l'ensemble de données, le reste pour la base de test.
  - Les taux obtenus sur un vocabulaire de 198 villes Iraniennes, sont :
  - 65% en premier choix et de 76.1 % en second choix.

- Interprétation et analyse des résultats
  - Simplicité de l'approche utilisée notamment au niveau de la phase d'extraction de primitives.
  - Nature cursive de script Farsi/arabe.
  - Similarité entre certains mots de lexique.

Pour améliorer cette approche, les auteurs proposent une autre version dans [Dehg01b], qui se résume en phases ci-dessous :

- Utilisation de la quantification vectorielle floue (algorithme C-moyen floue).
- Modélisation
  - Chaque mot est modélisé par un modèle FVQ/HMM de topologie droite-gauche.
  - Utilisation de la procédure Baum-Welsh modifiée pour l'apprentissage.
- Les résultats
  - Les scores de reconnaissance obtenus sont 67 % en top 1 et 96,5 en top 20 sur la même base de données que dans [Dehg01a].

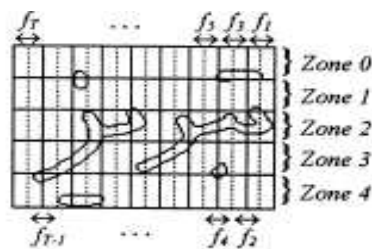


Figure 3-5 : Extraction de primitives par le principe de fenêtrage par [Dehg01a].

Dans [Masm00], Masmoudi et al présentent une méthode globale pour la reconnaissance des mots arabes manuscrits dans un vocabulaire de taille très limitée (10 classes). Pour cela, ils ont suivi la démarche ci-dessous :

- Extraction de primitives
  - Description globale de mots par des primitives visuelles (type de points diacritiques, boucles, occlusions, ascendants, descendants, traits saillants).
- Modélisation
  - Chaque mot est modélisé par un HMM1D de topologie linéaire droite-gauche à 6 états déterminés de manière empirique.
- Les résultats
  - La base de données utilisée est constituée de 1100 images de mots écrits par plusieurs scripteurs.
  - 72% de l'ensemble de données représente la base d'apprentissage, le reste est utilisé pour les tests.
  - Le taux de reconnaissance obtenu est de 96%.

Dans [Khor00], Khorsheed propose une approche de reconnaissance globale basée sur les HMMs 1D de type modèle discriminant pour les mots arabes manuscrits :

- Prétraitement et normalisation
  - Utilisation de la transformation polaire normalisée de l'image de mot.
- Extraction de primitives
  - Primitives extraites par les coefficients spectraux résultant de la transformation de Fourier.
  - Chaque mot est représenté par une séquence de primitives de taille 6.
- Modélisation
  - Le mot est représenté par un modèle droite gauche à 6 états.
  - Chaque état est associé à un secteur angulaire dans le spectre de Fourier.
  - Apprentissage par Baum-Welsh sur un vocabulaire de tailles 128, puis 256 entrées.
- Les résultats
  - Les performances en reconnaissance s'échelonnent entre 68.4% et 85.44%.

Ce type d'approches montre leur efficacité sur des vocabulaires limités. Cependant, elle devient problématique lorsque la taille du vocabulaire augmente (cf. section 2.6.1).

#### 4.1.2. Approches Analytiques

Dans [El-Ha05], R. Al-Hajj et al proposent un système de reconnaissance à base de HMM 1D à fenêtres glissantes pour la reconnaissance de mots manuscrits arabes, dont les principaux axes sont :

- Extraction de primitives
  - Primitives basées sur les densités de pixels (16) et les concavités locales (12) par les fenêtres verticales intercalées sur l'image binaire. Au total, chaque fenêtre engendre 28 caractéristiques.
  - Hauteur de la fenêtre correspond à la hauteur de mot.
  - Estimation de deux lignes de bases : ligne haute et ligne basse.
- Modélisation
  - Le modèle de mot est la concaténation des modèles de caractères.
  - Le caractère est modélisé par un HMM1D de gauche-droite à 4 états.
  - Chaque état est modélisé par 3 gaussiennes.
  - 167 modèles de caractères arabes (forme de milieu, fin, début,...).
  - La reconnaissance est effectuée par l'algorithme de Viterbi.
- Les résultats :
  - Utilisation de la base de données IFN/ENIT.
  - Un taux de reconnaissance en top 1 de 84.03% et un taux d'erreur de 15.97%.
  - Ce système a remporté la compétition ICDAR 2005 [Marg05], sous l'appellation TPar/UOB.

Dans [El-Ha06], Les auteurs ont présenté une version améliorée de l'approche précédente. Elle comporte brièvement les démarches suivantes :



- Extraction de primitives
  - Utilisation de 3 types de fenêtres : une fenêtre verticale, fenêtre inclinée d'un angle  $+\alpha$  et une fenêtre inclinée d'un angle  $-\alpha$  (cf. Figure 3-6 ci-dessous).
  - Chaque fenêtre engendre un vecteur de primitives.

Ces fenêtres inclinées consistent à remédier aux problèmes de points diacritiques et de l'inclinaison de textes.

- Modélisation
  - Association de 3 classifieurs HMMs 1D (1 référence, 2 classifieurs) aux 3 fenêtres précédentes.
  - La mise au point d'un algorithme de post traitement qui permet de fusionner la liste des candidats proposés par les 3 classifieurs. Cette fusion se fait en accumulant les logs-vraisemblances afin de calculer la réponse globale de système.
- Les résultats
  - Le taux de reconnaissance en top 1 est de 85.02% et la valeur de taux d'erreur est de 14.98%. en top 2 ce score à remonté jusqu'à 87% dans la compétition ICDAR sur la même base de données.

Cette méthode produite une amélioration notable (1.0 % en valeur absolue, ou une réduction de 6.27 % de taux d'erreur). La combinaison des 3 classifieurs permet ainsi d'obtenir des performances meilleures que chacun des trois pris séparément. Pour réaliser la fusion des résultats, trois approches sont proposées à partir des listes de résultats : une somme, un vote, ou l'utilisation d'un réseau de neurones de type MLP qui renvoie le classifieur à sélectionner à partir des scores des trois listes de reconnaissance.

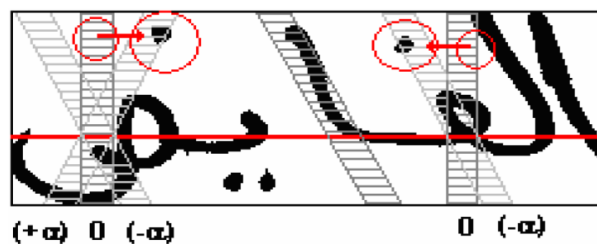


Figure 3-6 : Extraction de vecteurs de primitives avec des fenêtres inclinées [El-Ha06]

Dans ([Beno06a], [Beno06b]), Benouareth et al présentent une technique de type HMMs 1D semi continue avec une durée d'état explicite pour la reconnaissance des mots manuscrits arabes. Elle se résume essentiellement en étapes ci-dessous :

- Les prétraitements
  - Utilisation de la binarisation, lissage, estimation de la ligne de base et l'amincissement avec l'algorithme de Pavlidis.
- Extraction de caractéristiques

- Primitives sont basées sur les densités de pixels noirs, boucles, les points singuliers, les points extrêmes, les points de jonctions, les points d'inflexion, les points de rebroussement et les points diacritiques.
- Partitionnement de l'image en fenêtres verticales (uniforme ou non uniforme). La largeur de la fenêtre est déterminée par les minimas locaux d'histogramme de projection verticale adaptée à la modélisation HMM 1D (la largeur optimale est de 20 colonnes fixées empiriquement).
- Chaque fenêtre engendre un vecteur de 41 primitives.
- Quantification vectorielle :
  - Permet de faire correspondre au vecteur continu (vecteur de primitives) un indice discret d'un dictionnaire de référence (codebook), cette correspondance devienne un calcul de type plus proche voisin (PPV).
- Modélisation
  - La durée d'état à été modélisé par de différentes distributions (gaussienne, poisson et gamma). La durée maximale par état correspond au nombre maximal de trames fixé à 5.
  - Les modèles de mots sont obtenus par concaténation des modèles de lettres.
  - La notion d'état est associée au caractère.
  - Le modèle de caractère est de topologie droite gauche avec un saut inter-état (le type moins gourmant en temps de calcul et en nombre de paramètres).
  - Apprentissage est effectué par Baum-welsh avec les formules Russel et Levinson.
  - La classification est effectuée par Viterbi on se basant sur le critère de maximum de vraisemblance ML.
- Résultats

Les résultats des expériences obtenus sur la base de référence IFN-ENIT (3 parties utilisées en apprentissage et un pour les tests) montrent que les meilleurs taux de reconnaissance sont obtenus avec la distribution gamma pour la durée d'état, ils sont de 87.01% avec des fenêtres uniformes et de 89.79 % avec des fenêtres non uniformes.

Dans [Pech03], Pechwitz et al ont proposé une approche de reconnaissance de mots arabes manuscrits basée sur les HMMs 1D semi-continus consistant en étapes ci-dessous :

- Prétraitements et normalisation
  - Lissage, squelettisation et épaississement de trait par un filtre gaussien.
  - Estimation de la ligne de base et extraction de contours.
  - Normalisation de la taille (la largeur des caractères dans le mot et la hauteur des ascendants et descendants).
- Extraction de primitives
  - Primitives sont basées sur la technique des fenêtres glissantes de largeur 3 pixels (cf. Figure 3-7 ci-dessous).
  - Chaque fenêtre se déplace de droite à gauche en générant un vecteur de caractéristiques.

- La transformation de Karhunen- Loeve est utilisée pour réduire le nombre de primitives obtenues pour chaque fenêtre.
- Modélisation
  - Chaque caractère est modélisé par un HMM 1D de type gauche-droite.
  - Chaque modèle de caractère contient 7 états.
  - Les modèles de mots sont obtenus par concaténation de modèles de leurs lettres.
  - Des modèles additionnels pour modéliser les ligatures possibles entre les caractères et les caractères avec des marques diacritiques additionnelles
  - Le dictionnaire de 946 mots associés à la base IFN/ENIT est modélisé par 160 modèles
  - L'apprentissage et la reconnaissance sont effectués par l'algorithme de Viterbi.
- Les résultats
  - L'approche a été entraînée et testée sur la base IFN/ENIT en utilisant  $\frac{3}{4}$  des données pour l'apprentissage et le reste pour les tests.
  - Les scores de reconnaissance de 89.87 % en top 1 sans estimation de la ligne de base et de 84.94% avec estimation de la ligne de base ont été enregistrés.

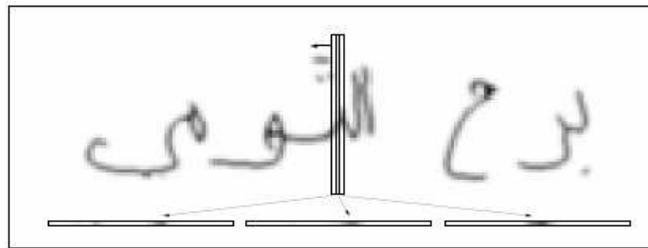


Figure 3-7 : Extraction de primitives par les fenêtres glissantes à trois colonnes de pixels dans [Pech02][Pech03]

En vue de hausser les performances de l'approche décrite précédemment. Les auteurs ont été introduit dans [Pech06], une autre stratégie pour extraire les caractéristiques qui incluse les étapes :

- Extraction de primitives
  - Les primitives sont extraites à partir des fenêtres glissantes. Chaque fenêtre est divisée horizontalement en 5 zones de hauteur fixe (cf. Figure 3-8 ci-dessous).
  - Un chevauchement de 50% possible entre deux fenêtres consécutives.
  - Les primitives correspondes aux longueurs (normalisées par rapport à la hauteur de la zone) des segments de lignes calculées suivant les directions nord, sud, est, ouest.
  - Chaque fenêtre engendre un vecteur de 20 composantes.
- Les résultats
  - Un taux de reconnaissance de 89.1% en top 1 avec estimation de la ligne de base.

Ceci a mis en évidence l'influence des caractéristiques extraites sur la capacité de discrimination des HMMs 1D.

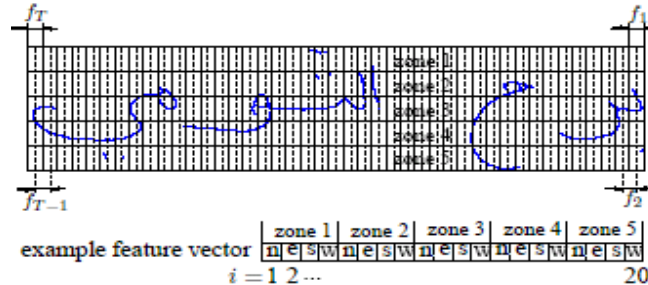


Figure 3-8 : Extraction des primitives par zonage [Pech06].

Dans [Khor03], Khorsheed et al ont utilisé une méthode basée sur un HMM 1D discret pour la reconnaissance analytique des mots arabes issus des manuscrits anciens, cette méthode consiste à entraîner un seul modèle de Markov caché global (PD-HMM : Path Discriminant HMM) en utilisant les étapes suivantes :

- prétraitements
  - La squelettisation, décomposition de graphe de squelette en segments par approximation linéaire.
- Extraction de primitives structurales
  - Les segments sont décrits à travers leurs longueurs et leurs angles d'orientation par rapport à l'axe horizontal (cf. Figure 3-9 ci-dessous).
  - Une quantification vectorielle est appliquée pour transformer la séquence de segments linéaires en une séquence de symboles discrets.
- Modélisation
  - Chaque modèle de caractère est de type gauche-droite.
  - La séquence de symboles est examinée par le modèle HMM 1D global.
  - Utilisation en totalité de 296 états pour tous les 32 caractères.
  - Le HMM 1D produit une liste ordonnée de séquences de lettres (N meilleurs chemins) pour chaque réalisation en entrée.
  - Classification est effectuée par l'algorithme de Viterbi.
- Résultats
  - Le meilleur taux de reconnaissance obtenu est 72% sans post-traitement, et est 87% avec post-traitement de la séquence de lettres générée.



Figure 3-9 : Exemple de squelettisation et approximation linéaire du mot manuscrit [Khor03]

Dans [Xue06], Xue et al ont proposé une approche basée sur HMM 1D groupant entre des symboles discrets avec d'autres continus (cf. Figure 3-10 ci-dessous) pour la reconnaissance des mots manuscrits. Pour cela, les auteurs proposent la démarche suivante :

- Extraction de primitives
  - Primitives visuelles de nature discrètes (les rebroussements courts et longs, les arcs, les ouvertures avec leurs directions « est, oust », les occlusions, les cercles, et les espaces).
  - Un ensemble des attributs de nature continu (position, orientation, angle, largeur) chacun est associé aux caractéristiques discrètes précitées.
  - La probabilité discrète  $P(u)$  est modélisée classiquement par des probabilités discrètes.
  - La probabilité  $P(v|u)$  continue est modélisée par un mélange de distributions gaussiennes multi-variables.
  - La probabilité d'observation mixte ( $u$  discrète,  $v$  continue) est exprimée par:
 
$$P(u, v) = P(u)P(v, u).$$

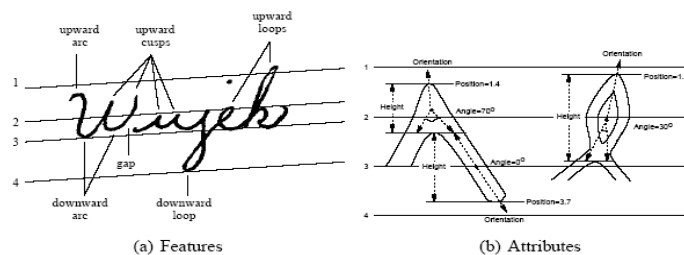


Figure 3-10 : Caractéristiques perceptuelles avec leurs attributs continus [Xue06].

- Modélisation
  - Utilisation de deux variantes des HMMs 1D (modèles d'états et modèles d'arcs).
  - L'apprentissage est réalisé par l'algorithme de Viterbi modifié et réadapté pour analyser les observations mixtes (combinaison discrètes et continues).
  - Les variantes majuscule et minuscule de chaque caractère sont modélisées par un HMM 1D de topologie linéaire et de gauche-droite.
  - Le nombre d'états par HMM 1D est fixé à la longueur moyenne des séquences d'apprentissage.
  - Les modèles de mots sont construits par concaténation des modèles de leurs lettres.
- Les résultats
  - Base d'apprentissage comporte 5596 échantillons manuscrits de noms de villes et de noms d'états, extraits de la base CEDAR.

- Base de test est constituée de 3000 échantillons manuscrits de noms de villes et de noms d'états tirés de la base CEDAR BHA.
- Les meilleurs taux de reconnaissance en top 1 pour un dictionnaire de 20 000 mots sont 51.13% pour les modèles d'états et 58.14% pour les modèles d'arcs.

*Siemens* est un système à base des HMMs 1D semi continus ([Scha05], [Scha08], [Caes93]), qui a été conçu initialement pour la reconnaissance de mots latins et la lecture automatique des adresses postales par le groupe de recherche Daimler Benz. Récemment il a été adapté pour la reconnaissance de mots manuscrits arabes et utilisé et remporté la compétition de script arabe ICDAR'07 [Marg07], dont voici les principales étapes.

- prétraitements et normalisation
  - Squelette, taille, largeur et rotation.
- Extraction de primitives
  - L'image du mot est divisée en plusieurs trames en appliquant une fenêtre glissante dont les paramètres sont la hauteur de texte et la largeur égale à 11 pixels et un chevauchement de 2/3 de la largeur.
  - Chaque fenêtre est divisée en six zones.
  - Les primitives sont basées sur les composantes connexes binaires et les contours noirs et blancs des polygones.
  - Un vecteur de caractéristiques géométriques (hampes, points de croisement...) de dimension 20 est extrait sur chaque trame.
  - Utilisation de la transformation LDA (A Linear Discriminant Analysis) pour la réduction des données.
- Modélisation
  - Le modèle d'écriture est basé sur l'ensemble de graphèmes.
  - Un caractère est modélisé par un HMM 1D avec self-transition composé de plusieurs chemins de type droite-gauche pour les caractères arabes et de gauche-droite pour les latins (cf.Figure 3-11 ci-dessous).
  - Les chemins correspondent aux différents modèles de graphèmes d'un caractère.
  - Les modèles des mots sont obtenus par concaténation des modèles de leurs lettres.
  - 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 [Scha08]. A noter qu'à partir de ce système, un deuxième a été proposé combinant trois classifieurs HMMs 1D avec trois jeux de caractéristiques différentes. C'est ce dernier système qui a gagné la compétition ICDAR'07 [Marg05].
- Les résultats
  - La lecture des adresses postales
    - Un taux entre 88% et 92% selon la taille de lexique utilisé (100 et 1000 noms de villes issus des données privés à la société Siemens).
    - Le taux de reconnaissance est de 88% sur la base de données IFN/ENIT.

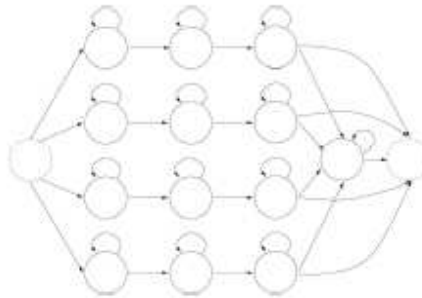


Figure 3-11 : Topologie d'un HMM modélisant un caractère selon [Scha08]

Dans ([Nata01],[Nata08]), Natarajan et al proposent un système **BBN** à base des HMMs 1D pour la reconnaissance des mots manuscrits. Le système a été proposé au départ pour la reconnaissance de la parole, puis converti et adapté pour la reconnaissance de l'écrit multilingues. Le système se résume en phases ci-dessous :

- Prétraitements et normalisations
  - Binarisation et correction de l'inclinaison des lignes.
- Extraction de primitives
  - Utilisation de la technique de fenêtres glissantes.
  - La largeur de chaque fenêtre est égale à 1/15 de la hauteur de texte.
  - La hauteur de chaque fenêtre est la hauteur de texte.
  - Deux bandes successives se chevauchent d'une 2/3 largeur de leurs largeurs.
  - Les primitives basées sur les centiles noires sont de nombre 20 par bande et d'autres sont basées sur l'angle et la corrélation.
  - En totalité, le vecteur de caractéristique est de 81 pour chaque fenêtre.
  - Utilisation de la transformation LDA (A Linear Discriminant Analysis) pour réduire les données de 81 jusqu'à 15 primitives.
- Modélisation
  - Chaque mot est représenté par un Modèle de Bakis 1D de 14 états.
  - Modélisation des émissions par des mélanges des gaussiennes.
  - L'apprentissage est effectué par Baum-welsh.
- Les résultats
  - Les taux de reconnaissance est de 59,9% sur la base de données IAM-DB.
  - Le score de reconnaissance est de 89,4% sur la base de données IFN/ENIT.
  - Les taux est de 83% sur la base ETL9B (200 exemples pour 71 Hiragan et 2965 lettres de Kanji en totalité 3036 caractères).

Le **CENPARMI** (Center for Pattern Recognition and Machine Intelligence) est un Système de reconnaissance à base des HMMs 1D développé par le groupe de recherche de Ching Suen pour être utilisé dans plusieurs applications telles que, les chèques bancaire, la lecture des adresses et la vérification des signatures ([El-Ya99], [Gran03],[Khor02], [Koer02], [Koer06]).

- Prétraitements et normalisations
  - Correction d'inclinaison de lignes (Skew).
  - Correction d'inclinaison de lettres (Slant).
  - Normalisation de la ligne de base supérieure.
  - Le lissage.
- Segmentation explicite
  - Le processus effectue une sur segmentation et il génère tous les points potentiels de segmentation qui décompose le mot en sous partie (graphèmes) et pas forcément en lettres. En effet, plusieurs solutions sont proposées et la meilleure sera retenue lors de la classification.
- Extraction de primitives
  - Utilisation de fenêtres glissantes.
  - Pour chaque point de segmentation (PS), le voisinage est divisé en bande.
  - Le centre de chaque bande est représenté par un point de segmentation (PS).
  - 3 types de primitives sont extraits à partir de cette bande.
  - Primitives globales (boucle, ascendants, descendants).
  - Primitives basées sur l'histogramme des transitions de contours pour chaque segment dans les directions horizontales et verticales (14 symboles).
  - Primitives basées sur la segmentation qui reflètent les relations entre les segments.
  - Utilisation de LDA (A Linear Discriminant Analysis) pour réduire la dimensionnalité des données.
- Modélisation
  - Le modèle de l'écriture est basé sur les modèles des graphèmes.
  - La topologie de modèle est spéciale comporte 8 états pour les segments spécifiques dont les transitions sont compliquées.
  - Un autre modèle est développé pour l'espace séparateur de topologie linéaire sans inter transition.
  - Modélisation des émissions par une distribution discrète.
  - L'apprentissage est réalisé par Baum-Welsh.
- Les résultats
  - La lecture des adresses postales

Pour un lexique des noms des villes françaises de 10 à 1000 entrées. Le score de reconnaissance est 78% sur un ensemble de test qui contient entre 400 et 1100.

- Les chèques bancaires

Pour les mots des mois manuscrits (un lexique de 12 entrées). Le score est de 90% sur un ensemble de test qui comporte 402 d'images.

Le système *IAM* à base des HMMs 1D ([Bert07], [Vinc04], [Mart01]), est proposé au sein de l'Institut de l'Informatique et de Mathématique Appliquée par le groupe Horst Bunke pour la reconnaissance de texte en-ligne et hors ligne, le système hors ligne consiste en :

- Prétraitements et normalisation



- Correction d'inclinaison des lignes et des caractères.
- Normalisation de ligne de texte.
- Extraction de primitives
  - Primitives basées sur la technique de fenêtres glissantes qui se déplacent au long de l'image dans le sens de l'écriture.
  - Pas de chevauchement entre deux bandes consécutives.
  - La hauteur de la bande est identique à la hauteur de texte.
  - Extraction de 9 primitives géométriques (poids de fenêtre, centre de gravité) sur chaque bande et d'autres sont basées sur la position et l'orientation de contours et en plus de certaines primitives statistiques.
- Modélisation
  - Le système est basé sur les modèles HMM 1D de caractères majuscules et minuscules de topologies linéaires.
  - Le nombre d'états par modèle est fixé empiriquement à 14 états avec des émissions continues.
  - Le modèle de mot est obtenu par la concaténation des modèles de lettres qui le composent.
  - Classification et apprentissage par Viterbi et Baum-welsh respectivement.
- Les résultats
  - Les taux de reconnaissance entre 47% et 62,7% sur la base IAM-DB.
  - Le taux de reconnaissance est de 91% sur la base Cambridge.

Le système *TUDO* (TU Dortmund University) à base des HMMs 1D semi continus ([Plöt08], [Wien05]), consiste à reconnaître de l'écriture manuscrite. Il a été adressé dans un premier temps pour la lecture des tableaux blancs, en plus il intègre un module de détection de texte permettant d'extraire les lignes de textes à partir de l'image. Les principales étapes de *TUDO* sont résumées ci-dessous :

- Normalisation
  - Correction des lignes et de texte et normalisation de la taille.
- Extraction de primitives
  - Utilisation de la technique de fenêtres glissantes.
  - Deux bandes successives se chevauchent de 75%.
  - La largeur de la bande est 8 pixels et la hauteur est identique à celle de texte.
  - Extraction de 9 primitives géométriques à partir de chaque bande et d'autres primitives basées sur l'analyse locale au niveau de pixels.
- Modélisation
  - Les modèles de mots sont obtenus par la concaténation des modèles séparés des caractères majuscules et minuscules, les modèles des chiffres et les modèles des ponctuations, en totalité 70 modèles.
  - La topologie utilisée pour les modèles est celle de Bakis.

- Le nombre d'états par modèle dépend de la longueur de segment (en moyenne 30).
- L'apprentissage et le décodage est effectué par Baum-Welsh et Viterbi respectivement.
- Les résultats
  - Le taux de reconnaissance est de 71,1 sur la base IAM-DB.
  - Le taux de reconnaissance est de 60,2 sur des images des tableaux blancs de la base IAM-OnDB.

Le système *TUM* a été proposé par le groupe de recherche de l'institut « Communication homme machine » à l'Université Technique de Munich. Le système à base des HMMs 1D continu est conçu pour la reconnaissance automatique de l'écriture manuscrite non contrainte (lecture des adresses) dans des larges vocabulaires ([Brak01], [Brak04]) dont les principales étapes sont détaillées ci-dessous :

- Les prétraitements et normalisation
  - Amélioration (binarisation et débruitage).
  - Normalisation (correction des lignes et des caractères, taille).
- Localisation et extraction
  - Utilisation des composantes connexes binaires pour extraire les lignes de texte.
  - Détection de la ligne de base supérieure et la ligne de base inférieure qui sont utilisées pour la rotation pendant la normalisation.
  - Détection des ascendants et descendants et la hauteur de texte.
- Extraction de primitives
  - Utilisation de la technique de fenêtres glissantes qui se déplacent dans le sens de l'écriture avec quelques chevauchements.
  - La hauteur de la fenêtre correspond à la hauteur de texte et de largeur inférieure à la largeur de caractère.
  - Chaque fenêtre est divisée horizontalement en 5 zones.
  - Utilisation des primitives géométriques (tirets, points, hampes, courbes, lignes verticales ou horizontales).
  - Application de la transformation LDA (A Linear Discriminant Analysis) pour réduire les données de 60 à 30.
- Modélisation
  - Le système est basé sur les modèles de lettres à 3 états (77 modèles en totale).
  - Chaque modèle est de topologie linéaire.
  - L'apprentissage et le décodage sont effectués par Baum Welsh et Viterbi respectivement.
  - Intégration des techniques spécifiques à l'adaptation aux modèles de mots, en particulier la méthode la plus efficace « scaled likelihood linear regression » SLLR.
  - Les mesures de confiance sont utilisées pour les hypothèses de rejet dans le cas où les taux sont faibles.

- Renforcement également des taux par l'intégration d'une version des approches HMM 1D qui utilise d'autres primitives pour la combinaison des HMMs 1D horizontaux et verticaux.
- Les résultats
  - La base de données utilisée est SEDAL qui contient les images des adresses qui ont été numérisés dans les bureaux de la poste Allemand.
  - Pour un ensemble de test de 2000 mots.
  - Le score obtenu est 86% pour les noms de villes pour un lexique 20K mots.
  - Le taux obtenu est 63,9 pour les noms des rues sur un dictionnaire 20K mots.

Le Tableau 3-2 ci-dessous résume les principales techniques d'extraction de caractéristiques, les performances des approches et les systèmes à base des modèles de Markov cachés unidimensionnel HMM 1D de reconnaissance de l'écriture manuscrite que nous avons détaillé précédemment dans cette partie.

Proposition	Prétraitement Normalisation	Primitives	Topologie des HMMs	reconnai-ssance	Performance
[Boug09]	-Suppression des allongements horizontaux, des espaces verticaux et horizontaux). -Transformation de l'image en signal 1D	-Coefficients MFCC	Linéaire gauche droite, 17 états par HMM	globale	70.78% / IFN/ENIT (sans prétraitements) 85.60% /IFN/ENIT (avec les prétraitements)
[El-Ha05]	-Binarisation -Estimation de la ligne de base	Structurelles et statistiques sur les fenêtres glissantes (28 primitives)	Modèle de caractères linéaire de gauche-droite à 4 états.	Analytique Fenêtres glissantes	en top 1 : 84.03% et le taux d'erreur : 15.97% / IFN/ENIT
[El-Ha06]	-Binarisation -Estimation de la ligne de base	3 types de fenêtres: verticale, inclinée d'un angle (+alpha, -alpha) en entrée de 3 classifieurs	Modèle de caractères linéaire de gauche-droite à 4 états.	Analytique Fenêtres glissantes	en top1 : 85.02% et la valeur de taux d'erreur : 14.98%. en top 2 remonte jusqu'à 87% ICDAR./ IFN/ENIT
[Beno06a] [Beno06b]	-Binarisation, -lissage, -Estimation de la ligne de base - Amincissement Algorithme de Pavlidis	Structurelles et statistiques (les densités de pixels noirs, boucles, les points singuliers, extrêmes et points de jonctions, les points d'inflexion, les points de rebroussement et diacritiques) extraites sur les fenêtres glissantes (uniforme ou non uniforme) → 41 primitives	topologie est droite gauche avec un saut inter-état et avec durée d'état explicite par gaussienne, poisson, et gamma état=caractère (durée maximale = 5 trames par états)	Analytique Fenêtres glissantes	87.01% / IFN/ENIT (avec la distribution gamma et des fenêtres uniformes,  89.79 % /IFN/ENIT (avec des fenêtres non uniformes)

<b>[Pech03]</b>	-Lissage, squelettisation, Épaississement de trait par un filtre gaussien, Estimation de la ligne de base, extraction de Contours. -Normalisation de la taille	technique des fenêtres glissantes	HMM de type gauche-droite à 7 états (caractère)	Analytique Fenêtres glissantes	: 89.87 % en top 1 (sans estimation de la ligne de base). : 84.94% (avec estimation de la ligne de base) /IFN/ENIT
<b>[Pech06]</b>	-Lissage, squelettisation, Épaississement de trait par un filtre gaussien, Estimation de la ligne de base, extraction de contours. -Normalisation de la taille	-Division horizontale des fenêtres en 5 zones. -Les primitives correspondent aux longueurs, des segments de lignes calculées suivant les directions nord, sud, est, ouest. En total : 20 primitives	HMM de type gauche-droite à 7 états (caractère)	Analytique Fenêtres glissantes	: 89.1% en top 1 (avec estimation de la ligne de base)  IFN/ENIT
<b>[Xue06]</b>		-primitives visuelles (les rebroussements courts et longs, les arcs, les ouvertures avec leurs directions (est, oust), les occlusions, les cercles, les espaces) - primitives continues (la position, l'orientation, l'angle, la largeur)	modèle de type gauche-droite nombre d'état est fixé à la longueur moyenne des séquences d'apprentissage.	Analytique	top 1 : 51.13% (pour un modèle d'états) : 58.14% (pour un modèle d'arcs)  / CEDAR et CEDAR BHA
<b>[Khor00]</b>	-Transformation polaire normalisée de l'image de mot.	-les coefficients spectraux résultant de la transformation de Fourier	-un modèle droite gauche à 6 états. - état = un secteur angulaire dans les spectres de Fourier	globale	68.4% 85.44% Mots issus à partir du manuscrit arabe ancien intitulé « gamhart alnasab libni lkalbi » par « hissaam abu lmundir alkalbi »
<b>[Khor03]</b>	-squelettisation, décomposition de graphe de squelette en segments par approximation linéaire	-primitives structurelles (longueur et son angle d'orientation par rapport à l'axe horizontal) extraites à partir des segments	Un modèle de type gauche-droite (caractère).	Analytique	: 72% (sans post-traitement) : 87% (avec post-traitement) -Manuscrit arabe ancien <b>[Khor00]</b>

[Masm00]		-primitives visuelles (type de points diacritique, boucles, occlusions, ascendants, descendants, traits saillants)	Modèle de mot de type droite-gauche à 6 états (empirique).	globale	96%. Base (1100 images de mots écrits par plusieurs scripteurs. 72% d'apprentissage, le reste de test.)
[Al-Ma02b]	-correction de l'inclinaison locale et globale de mots, épaisseur et hauteur de lettres.	-fenêtres glissantes -29 caractéristiques	-Mot est modélisé par HMM1D de type gauche droite (état= caractères)	globale	Presque 45%.  / AHDB
[Al-Ma04]		primitives visuelles (nombre de points hauts, de points bas et de segments) pour le classifieur heuristique		-combinaison séquentielle d' un classifieur fondé sur des règles heuristiques avec plusieurs classifieurs HMM1D fonctionnant en parallèles.	un score : 65% en top 1;  /AHDB
[Dehg01a]	-binarisation  -normalisation	-fenêtres glissantes  -primitives basées sur le contour  -division des fenêtres horizontale en 5 zones.  -20 primitives /zone	-Mot est modélisé par HMM1D de type gauche droite.	-globale	: 65%  (en premier choix)  : 76.1 %  (en second choix)  (17,000 exemples de 198 villes Irlandaises, 60% apprentissage, 40% tests)
[Dehg01b]	-binarisation  -normalisation	Même primitives	-mot est modélisé par un modèle FVQ/HMM de topologie droite-gauche	-globale  QV floue  (C-moyen floue)	: 67 % en top 1  : 96,5 en top 20.  Même base que dans  <b>[Dehg01a]</b>
<i>Siemens</i> [Scha05] [Scha08] [Caes93]	normalisation  (Squelette, taille, largeur, rotation)	-les composantes connexes binaires, les contours noirs et blancs de polygone.  - fenêtres glissantes  -géométrique (20)	-linéaire  -gauche droite (latin)  -droite gauche (arabe)	Analytique  fenêtres glissantes	entre: 88% et 92% (selon la taille de lexique : 100 et 1000 noms de villes issus de la société Siemens).  : 88% / IFN/ENIT

<b>BBN</b> [Nata01] [Nata08]	-binarisation -Correction de l'inclinaison des lignes	-fenêtres glissantes -les centiles noirs (20) -Angle et corrélation	-Bakis à 14 états (empirique) Pour chaque mot	Analytique fenêtres glissantes	59,9% /IAM-DB 89,4% IFN/ENIT 83% (CER) ETL9B
<b>CENPARMI</b> [El-Ya99] [Gran03][Koe02][Koe06]	-Correction d'inclinaison de lignes et de lettres -Normalisation de la ligne de base supérieure -Le lissage	-globale (boucle, ascendants, descendants) -Primitives basées sur l'histogramme des transitions (14 symboles). -Primitives basées sur la segmentation	- modèle spéciale à 8 états (caractères, espace)	-segmentation explicite -graphèmes	78% (4K et 11k images, lexique : de 10 à 1000 entrées). 90% (402 d'images, Lexique : 12 entrées)
<b>IAM</b> [Bert07][Vinc04] [Mart01]	-Correction d'inclinaison des lignes et des caractères -Normalisation de ligne de texte	-fenêtres glissantes -géométrique(9) (poids de fenêtre, centre gravité). - la position et l'orientation de contours et certaines statistiques des pixels.	-linéaire à 14 états (caractère)	Analytique fenêtres glissantes	entre 47% et 62,7% IAM-DB.  91% Cambridge
<b>TUDo</b> [Plöt08][Wien05]	-correction des lignes et de texte -normalisation de la taille	-fenêtres glissantes -géométriques(9) à partir de chaque bande -basées sur l'analyse locale des primitives.	-Bakis (1D) -nombre d'états (dépend de caractère en moyenne 30)	Analytique fenêtres glissantes	-71,1% IAM-DB -60,2 % IAM-OnDB <sup>Cam</sup>
<b>TUM</b> [Brak01] [Brak04]	-binarisation et débruitage. -correction des lignes et des caractères -normalisation de taille	-fenêtres glissantes -division en 5 bandes horizontales -Géométriques (60) (tirets, points, hampes, courbes, ligne verticale ou horizontale -LDA (60→30)	-modèle linéaire à 3 états	Analytique fenêtres glissantes	-86% (SEDAL Noms de villes)  -63,9 (SEDAL Noms de rues)

Tableau 3-2 : Caractéristiques et performances de certains systèmes à base des HMMs 1D de reconnaissance de l'écriture manuscrite

Dans ce type de systèmes, la reconnaissance est effectuée par l'évaluation de la vraisemblance d'une séquence d'observations générées à partir des primitives décrivant la forme à reconnaître par rapport aux modèles. Ce calcul s'effectue efficacement par

l'algorithme de Viterbi ou par les algorithmes Forward et Backward. Cependant, avant d'aborder cette tâche, les modèles doivent être entraînés souvent par de différents critères d'apprentissage. Ces critères s'inscrivent dans le cadre plus général de l'algorithme maximisation expectation (EM). Les émissions dans chaque état peuvent être modélisées par des observations continues (HMM 1D continu) ou discrètes (HMM 1D discret).

Les HMMs 1D continus utilisent les distributions continues pour modéliser les observations, dans la majorité des cas approximatés par des mixtures de gaussiennes. Or, pratiquement l'emploi d'une distribution continue engendre une augmentation très importante du nombre de paramètres à estimer, ce qui nécessite un corpus d'apprentissage très significatif pour l'estimation efficace de ces paramètres.

Les HMMs 1D discrets se basent sur un ensemble de symboles pour coder la séquence d'observation ; Ce codage nécessite un algorithme de quantification vectorielle afin de réduire la variabilité de l'observation réalisée, ce qui peut causer une perte d'information qui peut être décisive pendant la classification. En plus, les HMMs 1D semi-continus qui constituent la combinaison entre les deux cas précédents.

## **4.2. Les modèles HMMs planaires**

Les HMMs 1D précédents sont des modèles statistiques qui analysent uniquement les données séquentielles. En revanche, l'aspect de l'image des mots ou des caractères est bidimensionnel d'où la nécessité d'une transformation de celle-ci en séquence linéaire. Cependant, cette conversion n'est pas évidente, elle se qualifie de l'étape la plus difficile et critique de la chaîne de reconnaissance (cf. Section 2.2).

Les PHMMs de par leurs structures plus ou moins bidimensionnelles (pseudo bidimensionnel) sont une extension des HMMs 1D qui permettent d'opérer directement sur l'image d'écriture ou après que celle-ci ait subi une transformation mineure. Afin de s'affranchir des difficultés causées par le passage de l'image (2D) en séquence (1D). Ces modèles ont été appliqués sur l'imprimé et le manuscrit ([Saon97], [Bena00], [Bena99]). Nous présentons dans ce qui suit les principales approches globales et analytiques fondées sur les PHMMs pour la reconnaissance d'écriture.

### **4.2.1. Approches globales**

Dans [Saon97], Saon a présenté une approche fondée sur les PHMMs pour la reconnaissance des chiffres manuscrits qui se résume par les étapes suivantes :

- Extraction de primitives
  - Primitives basées sur les intensités des pixels.
- Modélisation
  - Le modèle optimal comporte 8 super-états et 4 états par super-état.

- La topologie du modèle principal est de type gauche droit.
- Les structures des modèles secondaires sont strictement de type gauche-droite (transitions autorisées vers l'état courant ou vers l'état suivant).
- L'apprentissage est effectué par un algorithme basé sur le critère de type maximum de vraisemblance de type (MLE: Maximum Likelihood Estimation).
- Les résultats
  - Utilisation de la base de données UNIPEN.
  - Base d'apprentissage (1260 échantillons de chiffres par modèle, c.-à-d. 90 % de toute la base UNIPEN, en totalité : 12600 images de taille 16×16).
  - Base de test (10 % restante de la base, c.-à-d. 14 échantillons par modèle de chiffre).
  - Les scores de reconnaissance moyens obtenus sont 91.36 % en top 1, et 99.64 % en top 5.

L'auteur souligne que les erreurs enregistrées sont due généralement à la confusion entre les chiffres 2 et 0.

- Particularité

Selon la littérature la majorité des systèmes qui utilisent les PHMMs font leur apprentissage par la procédure itérative de Viterbi, quant au système proposé par Saon lui adapte un critère de type maximum de vraisemblance (MLE: Maximum Likelihood Estimation) pour l'apprentissage des modèles de chiffres.

Dans ([Bena99], [Bena00]), Ben Amara et al ont développé une approche globale pour la reconnaissance de l'écriture (pseudo mots) arabe imprimée, cette technique est basée sur l'emploi des PHMMs. Elle comporte les phases ci-dessous :

- Modélisation
  - L'image d'un pseudo-mot est décomposée verticalement en plusieurs zones horizontales.
  - Les zones horizontales correspondent aux bandes de variation morphologique de l'écriture arabe : (les zones des points diacritiques supérieurs, extensions supérieurs, boucles et ligatures horizontales, extensions basses, et diacritiques inférieurs).
  - Chaque zone est modélisée par un modèle secondaire horizontal HMM 1D de type gauche- droite (cf. Figure 3-12 ci-dessous).
  - Les observations dans les HMMs 1D horizontaux sont basées sur des primitives extraites sur la distribution des pixels noirs/blancs et de sa longueur en pixels et de son emplacement relatif par rapport au segment qui se trouve au dessus.
  - La durée dans ces modèles secondaires représente les longueurs des ensembles des pixels voisins de même couleur, appelées segments.
  - Le pseudo-mot est modélisé par un modèle principal vertical spécifique à sa morphologie pilotant les HMMs 1D horizontaux (cf. Figure 3-12 ci-dessous).
  - Le modèle HMM 1D vertical (haut – bas) à durée explicite permet de prendre en compte la hauteur des différentes zones horizontales délimitées.



- La durée dans un super-état est assimilée au nombre de lignes analysées par lui.
- Les résultats
  - Base d'apprentissage (24 000 échantillons de même style (mono-police)).
  - Un vocabulaire (100 pseudo-mots correspondant à 85 noms de villes tunisiennes imprimés).
  - Le score de reconnaissance obtenu est de 99,84%.
- Particularité

Cette modélisation présente certaines spécificités : (1) l'application des PHMMs directement sur l'image d'un mot sans segmentation au préalable; (2) l'auteur considère la topologie utilisée comme solution efficace pour la modélisation des différentes variations de l'écriture arabe, comme les élongations des ligatures horizontales et la présence des ligatures verticales.

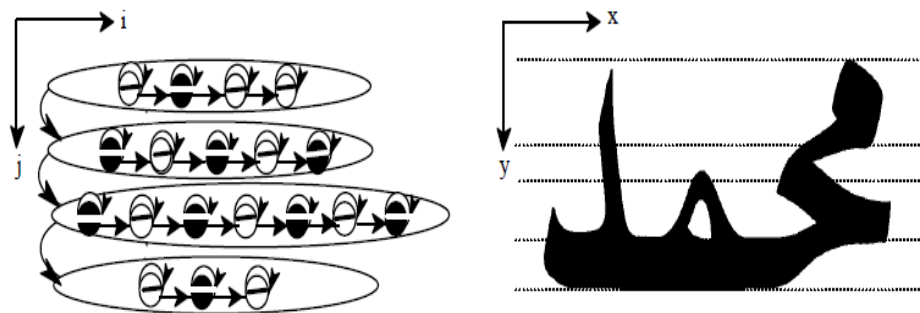


Figure 3-12 : Architecture d'un pseudo mot par PHMM, d'après [Bena99].

#### 4.2.2. Approches Analytiques

Dans ([Kuo94], [Agaz93]), Agazzi et Kuo ont présenté une approche à base des PHMMs pour la reconnaissance des caractères et des mots imprimés latins, comportant les étapes :

- Modélisation
  - Chaque caractère isolé est modélisé par un modèle principal vertical à super-états (cf. Figure 3-13 ci-dessous).
  - Chaque super-état correspond à un modèle horizontale.
  - Le nombre d'états et de super-états dépend de la topologie de la forme considérée (déterminé manuellement en fonction de la présence des zones informatives).
  - Les observations sont basées sur les primitives extraites directement de la matrice des pixels. Elles sont représentées par un vecteur à quatre composantes (la valeur du pixel 0/1, son voisinage horizontal, deux positions relatives du pixel par rapport à sa colonne et sa ligne).
  - L'apprentissage est effectué par l'algorithme k-means qui permet de faire des regroupements des observations en classes correspondantes aux zones des super-états.

- Partitionnement de la séquence d'observations globale de chaque échantillon en une séquence de super-états.
- Utilisation de l'algorithme de Viterbi pour déterminer l'alignement optimal de l'échantillon avec le modèle courant.
- Les paramètres du modèle sont ensuite ré estimés par comptage en fonction des résultats de la segmentation.
- L'initialisation des paramètres des modèles initiaux sont donnés manuellement et de manière arbitraire.
- Les résultats
  - Base de données contient environ 26000 mots synthétiques.
  - Les taux de reconnaissance réalisés varient entre 99% et 96%, selon que les mots de test et ceux utilisés pour l'apprentissage sont dans la même fonte ou non (respectivement).

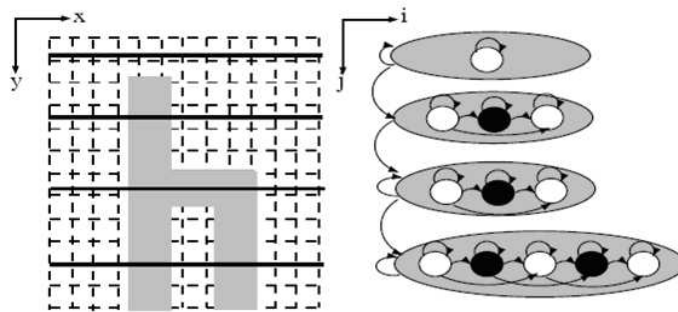


Figure 3-13 : modélisation de caractère imprimé h par un PHMM [Kuo94]

Dans [Bipp97], Bippus et al ont proposé une approche basée sur les PHMMs pour résoudre le problème de la reconnaissance de mots manuscrits issus des chèques bancaires allemands. L'approche proposée se résume en phases ci-dessous :

- Prétraitements et normalisations
  - Détection de la ligne médiane des mots et de l'inclinaison moyenne des caractères.
  - Correction des inclinaisons (rotation, cisaillement) et normalisation de la taille.
  - Squelettisation de l'image
  - Normalisation de l'épaisseur du trait d'écriture sur le squelette (un filtre gaussien convertissant l'image en niveaux de gris).
- Extraction des primitives
  - Primitives basées sur les niveaux de gris de l'image traitée par une transformation Karhunen-Loeve et extraites à partir des fenêtres rectangulaires à double balayage (cf. Figure 1-6 ci-dessus).
- Modélisation

- Le HMM 1D principal, orienté horizontalement, représente une concaténation de modèles de lettres à 3 états, chaque HMM 1D secondaire étant composé de 5 états.
- La topologie du modèle principal est de type Bakis avec possibilité de saut d'un état.
- L'architecture des modèles secondaires étant de type Bakis sans saut d'états.
- L'apprentissage est effectué par l'algorithme k-means de la même manière que Agazzi et Kuo ([Kuo94][Agaz93]).
- La reconnaissance du mot se fait par Viterbi dans un dictionnaire organisé sous forme arborescente.
- Les résultats
  - Base de données contient 1634 montants simulés par 159 scripteurs.
  - un vocabulaire de 47000 entrées.
  - les scores de reconnaissance sont de 84.2% (sans insertion) et 79.9% (avec insertion) par les PHMM.
  - Un gain de 0,7% par les HMMs1D semi-continus utilisant les mêmes types de primitives.
  - les scores sont de 86.8%, (sans insertion) et 83.5% (avec insertion) par la combinaison des deux approches.
- Particularité

L'auteur souligne que l'intérêt de ce système ne réside pas dans l'architecture et la modélisation des PHMMs employée. Mais, réside dans la démarche suivie pendant les phases des prétraitements et d'extraction des primitives.

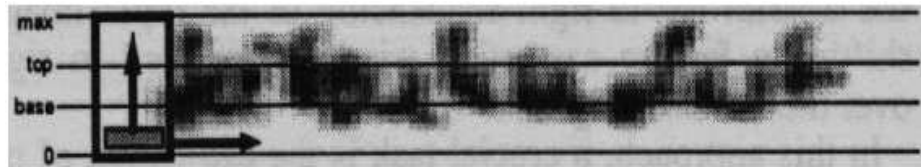


Figure 3-14 : Extraction de primitives par fenêtre glissante [Bipp97]

Le tableau Tableau 3-3 ci-dessous récapitule les principales caractéristiques techniques et les résultats obtenus par des approches et les systèmes à base des modèles de Markov cachés planaires (PHMM) de reconnaissance de l'écriture que nous avons présentée précédemment.

Auteurs	Primitives	Approche	Modélisation	Résultats
[Saon97]	- les intensités (couleurs) des pixels	-globale	-HMM 8 super-états et 4 états par super-état. -Les topologies de type gauche droit. -L'apprentissage le critère (MLE)	-91.36 % en top 1. -99.64 % en top 5. Base (UNIPEN). Base d'apprentissage (90 %). Base de test (10 %).

[Bena99] [Bena00]	-distribution des pixels noirs/blancs. - longueur en pixels. - emplacement relatif.	-globale	-HMM principal vertical (morphologie, durée). - décomposée verticalement en zones horizontales. - zone horizontale est modélisée par 1HMM 1D de type gauche- droite	99,84% Base d'apprentissage (24 000) Un vocabulaire (100 pseudo-mots, 85 noms).
[Kuo94][ Agaz93]	-intensités de pixels -voisinage et positions	-analytique	-HMM principal vertical. -K-mean et Viterbi	-entre 96% et 99% selon les données utilisées. -Base (26000 mots)
[Bipp97]	-transformation Karhunen-Loeve -fenêtres rectangulaires (niveaux de gris)	-analytique	-HMM principal horizontal. -apprentissage (k-mean) -classification(Viterbi) -Bakis modifié	-84.2% (sans insertions) -79.9% (avec) PHMM. -Un gain de 0,7% HMMs1D semi-continu -86.8%, (sans insertions) - 83.5% (avec) PHMM&PHMM -Base (1634). -un vocabulaire (47000)

Tableau 3-3 : Caractéristiques et performances des systèmes à base des PHMM s décrits précédemment pour la reconnaissance de l'écriture

Les modèles pseudo-2D ou planaires (PHMMs) constituent une extension des modèles HMMs 1D au plan en tentant de s'affranchir des inconvénients et difficultés rencontrés au cours de la segmentation et de la conversion 2D en 1D omni présentes dans les modèles HMM1D.

L'avantage majeur des PHMMs réside dans leur structure qui opère sur les axes verticaux et horizontaux et qui permet d'analyser des images de différentes tailles. De ce fait, ils présentent des propriétés d'appariement dans les deux directions. Cependant, les PHMMs échouent face aux problèmes présentant une grande variabilité (ex. écriture manuscrite).

#### 4.3. Modèles HMM bidimensionnels (Champs de Markov ou HMM 2D)

Bien que les modèles HMMs planaires PHMM sont une extension des modèles HMMs 1D au plan, ces modèles ne constituent pas une vraie modélisation de l'aspect 2D de l'image. Par contre, les champs de Markov analysent et opèrent directement sur une vraie structure 2D. L'inconvénient majeur des algorithmes de ce type réside dans leur complexité en terme de temps, leur coût calculatoire exponentiel, ce qui les rendent inapplicables.

Du ce fait, un nombre limité des méthodes à base des champs de Markov (HMM 2D) ont été proposées dans la littérature pour la reconnaissance de l'écriture. Par la suite, nous décrivons les plus marquées.

Dans [Saon97], Saon a proposé deux systèmes, le premier se base sur les champs aléatoires de Pickard cachés (HPRF : Hidden Packard Random Field) pour aborder le problème de la reconnaissance des chiffres manuscrits, il inclut principalement les étapes suivantes:

- Modélisation

- La topologie des modèles est de type ergodique avec initialisation équiprobable des transitions.
- Le nombre d'états a été fixé empiriquement à 16.
- Les probabilités d'observations sont initialisées par comptage dans des décompositions 4×4 des images.
- L'apprentissage des HPRFs est identique aux HMMs 1D discrets appris sur les lignes et les colonnes des images vues comme des séquences d'observations multiples.
- Les résultats
  - Base de données utilisée : UNIPEN.
  - Base d'apprentissage : 9/10 des échantillons.
  - Base de test : 1/10, c.-à-d. 140 échantillons par chiffre.
  - Les scores moyens de reconnaissance qu'ont été enregistrés sont 90% en top 1, et 99.36 en top 5.

Ces résultats ont été jugés décevants par l'auteur.

- Spécificités

Les modèles HPRFs se caractérisent par certaines particularités : (1) Ils sont homogènes et isotropes ; (2) leur comportement peut être simulé par une chaîne de Markov d'ordre 1 dans le cas où ils sont réduits à une ligne ou à une colonne de l'image ; (3) leur estimation de la vraisemblance d'une image se prête à un traitement causal ; (4) leur hypothèse simplificatrice d'indépendance des lignes (ou colonnes) affaiblie leur pouvoir de modélisation bidimensionnelle.

La deuxième proposition de Saon [Saon97], quant à elle est un système basé sur un module hybride combinant les champs de Markov unilatéraux (NSHP : Non-Symmetric Half-Plane Markov Model) et les modèles de Markov cachés HMM (NSHP-HMMs) pour la reconnaissance de l'écriture manuscrite hors-ligne.

Ces modèles présentent des avantages de calcul de la probabilité de masse ce qui correspond bien au sens (gauche-droite) d'écriture manuscrite hors-ligne par des procédures récursives. La composante NSHP de ces modèles possède une grande capacité grâce au 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 variation de l'écriture selon les deux directions. Quant à la composante HMM, elle consiste à modifier dynamiquement les distributions des NSHPs afin de les adapter aux caractéristiques rencontrées dans l'image pour maximiser sa vraisemblance. Ces modèles semblent avoir des capacités impressionnantes pour l'appariement élastique et leur apprentissage est basé sur le critère MLE (Maximum Likelihood Estimation).

Cet apprentissage est effectué par un comptage non supervisé ML (Maximum Likelihood) de configurations de pixels, 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.

- Prétraitements
  - Normalisation en hauteur des images.
- Modélisation
  - Topologie de type gauche-droite avec initialisation équiprobable des transitions (cf. Figure 3-15-a ci-dessous).
  - Nombre d'états proportionnel à la longueur moyenne des échantillons en colonnes de pixels (varie entre 11 et 35 pour une hauteur de 20 lignes).
  - Nombre de lignes varie entre 10, 15 et 20 selon les images utilisées.
  - Ordre des modèles varie entre 0 et 4 suivant les voisinages (cf. Figure-b ci-dessous).
  - Probabilités conditionnelles d'observation de pixels : toutes les images d'un même mot ont été divisées en N bandes verticales de largeur égale. Un comptage normalisé du nombre de configurations de pixels est effectué à l'intérieur de chaque bande pour tous les échantillons.
  - Utilisation de quelques techniques pour améliorer l'apprentissage (Apprentissage correctif, interpolation avec effacement, Liaison de paramètres).
- Les résultats
  - Base de données : SRTP.
  - Base d'apprentissage : 4653 images approximativement 2/3 de la base.
  - Base de test : 2378 images représentant 1/3 de la base.
  - Le taux de reconnaissance est de 90.08% en première réponse.

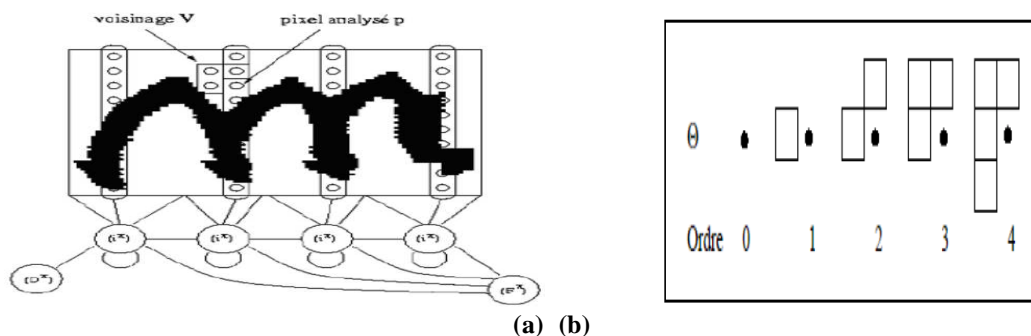


Figure 3-15 : (a) : modélisation utilisée, (b) : voisines utilisés[Saon97]

Dans [Chev04], Chevalier a présenté une approche basée sur les champs de Markov cachés pour le problème de la reconnaissance de l'écriture manuscrite et de chiffres manuscrits. L'approche utilise une analyse entièrement bidimensionnelle de l'écriture fondée sur la combinaison des fenêtres glissantes sur l'image et d'une modélisation markovienne piloté par l'algorithme de la programmation dynamique 2D. Cet algorithme consiste à effectuer un décodage rapide et optimal des champs de Markov. L'importance de la méthode

proposée 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.

- Les résultats
  - Pour les chiffres manuscrits :

Base de données utilisée : MNIST.

Un score de reconnaissance de 86,3 est obtenu avec les paramètres de modélisation (cf. Tableau ci-dessous).

Nombre d'étiquettes	Fenêtres gaussiennes	élagage	Nombre d'observations	Nombre d'itérations
5×7	7×7	30	14×14	12

Tableau 3-4 : paramètres de la modélisation associée au taux 86,3.

- Pour la reconnaissance de mots :

Un taux de 40% avait été obtenu en ne traitant que les mots de taille inférieure ou égale à 4 lettres. Les auteurs estiment que les performances sensiblement accrues sont attendues avec l'amélioration des primitives utilisées et des stratégies d'apprentissage et d'élagage.

Le Tableau 3-5 ci-dessous récapitule les principaux systèmes à base des champs de Markov (HMM 2D) en précisant leurs techniques et les scores enregistrés pour chacun.

Auteurs		Modélisation	Performances
[Saon97]	Système 1	-le modèle HPRF - Architecture ergodique. - nombre d'états = 16. -Apprentissage= appren. HMM1D	-90% en top 1. -99.36 en top 5. Base (UNIPEN).
	Système 2	- Le modèle NSH-PHMMs. -Apprentissage non supervisé ML. -Topologie gauche droite. - $11 \leq$ nombre d'états $\leq 35$ .	-90.08% Base (SRTP).
[Chev04]		- fenêtres glissantes. - champs de Markov cachés. - la programmation dynamique 2D	-86,3 Base (MNIST). -40% taille (mots) $\leq 4$ lettres.

Tableau 3-5 : Caractéristiques et performances des systèmes à base champs de Markov précités.

Contrairement aux modèles HMM1D et PHMM, ceux des champs aléatoires de Markov (HMM 2D) ont naturellement une structure bidimensionnelle (2D). De ce fait, ils s'opèrent directement au plus bas niveau de représentation. C'est à dire sur les intensités de pixels des images et par l'utilisation de procédures de calcul récursives. Leur structure (2D) s'adapte de manière beaucoup plus réaliste avec l'aspect de (2D) des images de l'écriture. Cependant, à cause de leur coût calculatoire prohibitif qui croît de manière exponentielle

avec la taille de l'image et le nombre de paramètres, ces modèles n'ont pas connu en reconnaissance d'écriture le succès escompté.

## 5. Conclusion

Nous avons tracé dans ce chapitre un état de l'art sur l'application des modèles de Markov pour la reconnaissance de l'écriture. Nous avons introduits les différentes étapes principales pilotant un système de référence markovien ainsi que quelques bases de données et systèmes markoviens de reconnaissance données dans la littérature. Nous avons répartis ces systèmes suivant la dimensionnalité des données de leurs modèles de base (HMM 1D, PHMM, HMM 2D).

A l'issue de cette étude nous noterons que l'utilisation de ces différents modèles dans le cadre de la reconnaissance d'écriture a été illustrée à travers la description de plusieurs systèmes de la littérature. Ces systèmes se distinguent entre eux par la dimensionnalité du modèles (1D, pseudo-2D ou 2D), l'unité de base de la modélisation (graphème, lettre ou mot), la nature de la distribution d'observation (discrète, continue ou semi-continue), le critère d'apprentissage (discriminatif ou génératif), la méthode de segmentation (explicite, implicite ou sans segmentation), la topologie du modèle (ergodique, gauche droite), la notion associée aux états du modèle, l'approche de reconnaissance (modèle discriminant ou chemin discriminant) et la modélisation de la durée (explicite ou implicite).

Cependant, la majorité de ces systèmes se croisent tous dans la manière utilisée pour s'affranchir l'étape de transformer l'image en données séquentielles. Ils se servent des techniques des fenêtres glissantes afin d'extraire le vecteur de primitives, généralement pas de théorie qui justifie telle ou telle méthode mais pratiquement ces fenêtres s'adaptent bien avec les modèles HMMs.

Par conséquent, une comparaison objective de ces systèmes semble très difficile. Cette tâche peut se faire uniquement sur des benchmarks bien définies et publiquement accessible, par exemple la base de la graphie arabe IFN/ENIT qui a été bien préparée et subdivisée préalablement en données de d'apprentissage, test et de validation. D'ailleurs, la comparaison de différentes systèmes été l'objet de plusieurs compétitions ICDAR 2005, ICDAR 2007.

Bien que ce domaine a été depuis longtemps étudié (plusieurs publications et compétitions), 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 ou l'information numérique confrontée à celle textuelle permet d'avoir des systèmes plus robustes.



Dans les chapitres suivants, nous présentons nos contributions dans le domaine de la reconnaissance de l'écriture.

En effet et en premier temps, nous avons développé une approche de reconnaissance de caractère isolés indépendante de la nature du script, basée sur les primitives directionnelles obtenues à partir de la transformée de Hough des caractères. L'approche conçue adopte une modélisation markovienne de type modèle discriminant. Nous évaluons le système de reconnaissance proposé sur des bases de données de caractères Arabes et Amazighes. Nous montrons que l'approche proposée donne des résultats encourageants.

Dans le second temps, nous avons proposée un autre système de reconnaissance de caractères imprimés, basé sur une nouvelle approche qui exploite les caractéristiques et les spécificités morphologiques de la langue amazighe par une modélisation markovienne de type chemin discriminant. La solution apportée opère sur un vocabulaire de base formé de différents graphèmes fondamentaux du caractère amazighe. Les scores obtenus sur la base des patterns de la graphie amazighe montrent la robustesse de l'approche proposée.

---

**PARTIE II:**  
CONTRIBUTIONS

---

# Chapitre 4 :Reconnaissance de caractères arabes et amazighes manuscrits isolés par les Modèles de Markov cachés

---

## 6. Introduction

Les systèmes de reconnaissance de caractères manuscrits isolés sont au cœur de la chaîne de reconnaissance de l'écriture manuscrite. Ils ont été considérés comme des systèmes les plus bénéficiés des recherches. En conséquence, plusieurs choix possibles quant à la modélisation et au traitement des caractères. En effet, la majorité de ces systèmes adoptent des approches basées sur des modélisations statistiques modernes telles que les réseaux de neurones, les machines à vecteurs de support ou les modèles de Markov cachés ([Bozi89],[Trie95], [Govi90], [Plam00], [Jain00], [Aric01]).

Dans le même contexte, nous proposons une approche basée sur les modèles de Markov cachés de type modèle discriminant DM-HMM qui s'intéresse aux problèmes de caractères isolés. Ce type de modélisation est largement utilisée dans le domaine de la reconnaissance de la parole. Un processus Markovien met en œuvre des modèles probabilistes spécifiques dans le but de gérer l'incertitude et le manque d'informations qui entachent les formes à reconnaître. Le système développé est évalué et testé sur des images de caractères arabes et amazighes manuscrites isolées. Ce système fait recours à des primitives directionnelles dans la génération des séquences d'observations, qui sont obtenues à l'aide de la technique des fenêtres glissantes opérant sur la transformée standard de Hough des images de caractères.

Dans ce chapitre, nous évoquons premièrement les principales caractéristiques morphologiques de l'écriture arabe et amazighe. Après, nous présentons l'architecture de base de système de reconnaissance automatique de caractères isolés proposé, basé sur les modèles de Markov cachés de type modèle discriminant. Dans la suite, nous décrivons les prétraitements effectués sur les deux types de caractères arabes et amazighes. Ensuite, nous décrivons la phase de l'extraction de primitives directionnelles à partir de l'image de caractère en utilisant la technique des fenêtres glissantes. Les séquences d'observations générées sont utilisées lors des phases de l'apprentissage et de la classification markoviennes pour entraîner les modèles des caractères et pour décider de leurs classes d'appartenance. En fin, nous présentons les expériences effectuées en montrant les résultats encourageants obtenus sur un ensemble de caractères arabes manuscrits isolés et sur une base de données de caractères amazighes manuscrits [Essa11]. Nous concluons par la suggestion de quelques perspectives de travail.

## 7. Présentation de l'écriture arabe et amazighe

Dans cette section, nous présentons un aperçu sur les principales caractéristiques morphologiques de l'écriture arabe et amazighe. Pour une synthèse plus détaillée de ces caractéristiques, les lecteurs peuvent se référer aux travaux ([Bena00], [El-Ha07], [Mena08], [Kess09], [Essa12]).

### 7.1. Caractéristiques morphologiques de l'écriture arabe

L'écriture arabe est semi-cursive aussi bien dans sa forme imprimée que manuscrite. Elle s'écrit de droite à gauche. Elle comprend un alphabet comportant 28 lettres comme le montre la Figure 4-1 ci-dessous.

En écriture arabe, 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. En revanche, la plupart des lettres s'attachent entre elles, même en imprimé. Leurs graphies diffèrent selon qu'elles sont précédées et/ou suivies d'autres lettres ou qu'elles sont isolées. C.-à-d. leur forme dépend de leur position dans le mot. Certaines lettres prennent jusqu'à 4 formes différentes : par exemple la lettre kaf (k) ( ك , ك , ك , ك ) ou le Mim (M) ( م , م , م , م ).

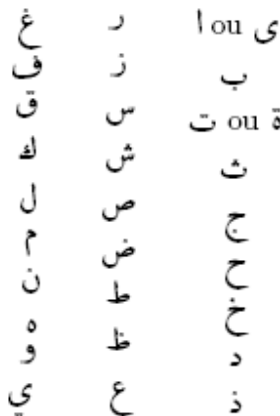


Figure 4-1 : Alphabet arabe

En plus, des spécificités signalées préalablement, l'écriture arabe possède d'autres caractéristiques. Nous citons les plus courantes, notamment les signes diacritiques, ascendants et descendants et la notion de pseudo mot.

#### Signes diacritiques

la majorité des caractères isolés arabes reposent sur le même tracé de base. En effet, dans le but de différencier entre eux, On a recourt à la notion de signe diacritique. Elle constitue une composante secondaire d'une lettre, qui vient la compléter ou en modifier le sens. Dans l'alphabet arabe, 15 lettres parmi les 28 possèdent un ou plusieurs points. La Figure 4-2 ci-dessous illustre un ensemble des caractères avec les signes diacritiques qui sont situés soit

au-dessus, soit en dessous de la forme à laquelle ils sont associés, mais jamais les deux à la fois.

ت ح ي ز ذ ب ش ض خ

Figure 4-2 : Lettres arabes avec signes diacritiques

Au contraire au latin, les caractères arabes sont majoritairement des consonnes, les voyelles ne sont pas des lettres, mais ils sont généralement représentés par ces signes diacritiques.

#### *Ascendants et descendants*

Comme dans l'écriture latine, l'écriture arabe contient des ascendants et des descendants. La Figure 4-3 ci-dessous montre un exemple d'un descendant sur la lettre Mim (M) (cercle en verre), et d'un ascendant de la lettre Lam (L) (cercle en bleu).

ح ل د م ا

Figure 4-3 : Les ascendants et descendants entourés et la ligne de base en rouge

#### *Notion de pseudo mot*

L'alphabet arabe comporte 6 caractères qui ne sont pas connectés à leur successeur : ز, ر, ذ, د, ا, و. Ces caractères introduisent donc une discontinuité dans le mot. Un pseudo-mot est une composante connexe associant un ou plusieurs caractères sous forme d'une séquence. Un mot peut être composé d'un ou plusieurs pseudo-mots (cf. Figure 4-4 ci-dessous).

التكنولوجيا ← التكنولوجيا | ل | ا جيا

Figure 4-4 : Le mot technologie en arabe est composé de 3 pseudos mots

Nous avons donné ci-dessus un survol sur les caractéristiques essentielles de la graphie arabe. Toutefois, Nous noterons que dans l'expérimentation (cf. Section 0) de notre système développé, nous considérons uniquement un ensemble de caractères arabes manuscrits dans leurs formes isolées sans points diacritiques. La Figure 4-5 ci-dessous montre quelques exemples de caractères étudiés.

ك ح ب س ع

Figure 4-5 : Les caractères arabes dans leurs formes isolées

## 7.2. Caractéristiques morphologiques de l'écriture amazighe

L'écriture amazighe est utilisée en Afrique de nord notamment au Maroc [Essa12]. Elle est de nature alphabétique consonantique. A la différence des caractères arabes, l'écriture amazighe n'est pas cursive. Ce qui facilite toute opération de segmentation. De plus, l'écriture amazighe est écrite de gauche à droite. Elle utilise des signes de ponctuation classique acceptés en alphabet latin.

En écriture amazighe, les notions de lettre capitale et lettre minuscule n'existent pas. Il n'y a pas de différence entre la taille verticale de différentes lettres de l'alphabet. C.-à-d., tous les caractères amazighes s'écrivent de même taille verticale sauf le seul caractère ya (ⵢ), qu'est plut petit par rapport aux autres caractères. D'où, en écriture amazighe, il n'y a pas d'ascendants et descendants.

L'écriture amazighe comprend un alphabet comportant 33 lettres comme le montre la Figure 4-6 ci-dessous



ⵍⵔⵓⵔⵕⵍⵏⵎⵉⵙⵉⵢⵙⵉⵢⵙⵉⵢⵙⵉⵢ

ⵔ ⵔⵓ ⵓ ⵓⵓ ⵓⵓⵓ ⵓⵓⵓ ⵓⵓⵓ ⵓⵓⵓ

ⵕ ⵕⵓ ⵕⵓⵕ ⵕⵕⵕ ⵕⵕⵕ ⵕⵕⵕ ⵕⵕⵕ ⵕⵕⵕ

ⵓ ⵓⵓ ⵓⵓⵓ ⵓⵓⵓ ⵓⵓⵓ ⵓⵓⵓ ⵓⵓⵓ ⵓⵓⵓ

ⵔ ⵕⵓ ⵕⵓⵕ ⵕⵕⵕ ⵕⵕⵕ ⵕⵕⵕ ⵕⵕⵕ ⵕⵕⵕ

ⵔ	ⵕⵓ	ⵕⵓⵕ	ⵕⵕⵕ	ⵕ	ⵕⵓ	ⵕ	ⵕⵕ	ⵕ	ⵕⵓ	ⵕⵕ
a	z	e	r	t	y	u	i	g	k <sup>w</sup>	ɾ
ⵕⵕ	ⵕ	ⵕⵕ	ⵕⵕⵕ	ⵕⵕⵕ	ⵕⵕⵕ	ⵕ	ⵕⵕ	ⵕⵓ	ⵕⵕ	ⵕⵕ
ⵔ	p	q	s	f	h	j	k	ā	i	ä
ⵕⵕ	ⵕⵕ	ⵕⵕⵕ	ⵕⵕⵕ	ⵕⵕⵕ	ⵕ	ⵕⵕ	ⵕ	ⵕⵕ	ⵕⵕ	ⵕⵕ
l	m	w	x	v	b	n	d	ṣ	c	z

Figure 4-6 : Alphabet amazighe

En plus, en écriture amazighe, il n'y a pas de notion de pseudo-mot. En revanche, la majorité des modèles graphiques des caractères amazighes est composée de points, de petits cercles, et/ou de segments. Nous distinguons généralement entre deux groupes des caractères amazighes suivants leurs formes:

- Formes circulaires: ⵔ, ⵕⵓ, ⵕⵕⵕ, ⵕⵕⵕ, ⵕⵕⵕ, ⵕⵕⵕ, ⵕⵕⵕ.
- Formes non circulaires : ⵕ, ⵕⵕⵕ, ⵕⵕⵕ, ⵕⵕⵕ, ⵕⵕⵕ, ⵕⵕⵕ, ⵕⵕⵕ, ⵕⵕⵕ, ⵕⵕⵕ, ⵕⵕⵕ, ⵕⵕⵕ.
- Formes avec deux ou trois composantes connexes: ⵕⵓ, ⵕⵕⵕ, ⵕⵕⵕ, ⵕⵕⵕ, ⵕⵕⵕ.

L'alphabet amazighe est formée généralement des caractères sans signes diacritiques. Néanmoins, il comporte seulement deux caractères affectés d'un signe diacritique qui sont situés toujours au-dessus de la forme à laquelle ils sont associés: ⵕⵓ «yakw» et ⵕⵓ «yagw».

La Figure 4-7 ci-dessous montre certaines de ces caractéristiques morphologiques dans un texte amazighe extrait dans un manuel scolaire.

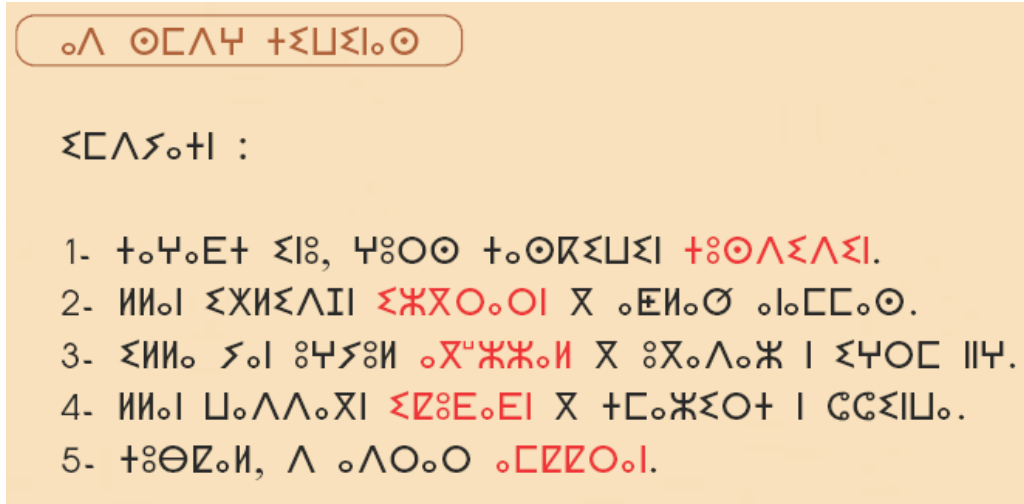


Figure 4-7 : Exemple du texte amazighe dans un manuel scolaire [Essa12]

Nous avons cité ci-dessus quelques caractéristiques morphologiques de l'écriture amazighe incarnées par l'alphabet tifinagh. Ces caractéristiques seront exploitées pendant la mise en ouvre de système de reconnaissance proposé et présenté dans le chapitre 5. Néanmoins, le système de reconnaissance présenté dans ce chapitre, opère sur la forme de caractère par les fenêtres glissantes qu'il que soit son type de graphie (arabe ou amazighe). De ce fait, ce système ne tient pas compte en général des spécificités de l'écriture étudiée.

Nous présentons par la suite, les composantes de l'architecture utilisée par notre système et l'approche utilisée dans ses différentes phases. Les résultats expérimentaux ainsi que la conclusion et les perspectives de la méthode sont donnés dans les deux derniers paragraphes.

## 8. Architecture du système proposé

L'architecture de système de reconnaissance de caractères manuscrits isolés proposé est composé de plusieurs étapes à savoir : les prétraitements, la représentation des données en séquences d'observations et la prise de décision par la modélisation markovienne (cf. Figure 4-8 ci-dessous).

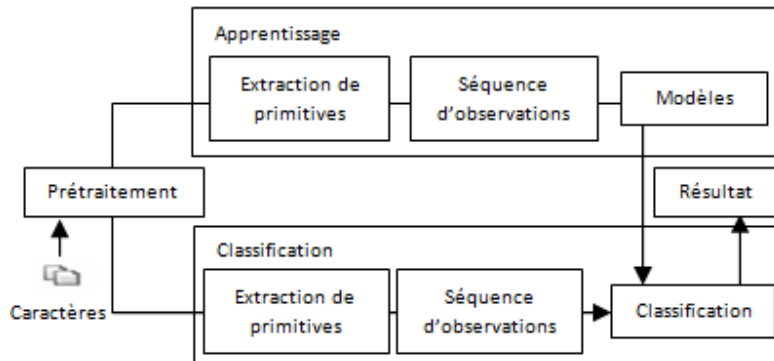


Figure 4-8 : Chaîne de traitement du système proposé

Les prétraitements visent à nettoyer l'image en vue de faciliter les traitements ultérieurs. Ils comprennent les opérations : binarisation, lissage, normalisation de la taille des caractères et la squelettisation.

La représentation des données se traduit par une phase d'extraction de primitives directionnelles générant une séquence d'observations de la forme à reconnaître.

La prise de décision s'effectue en deux étapes essentielles : En premier lieu, l'apprentissage est utilisé pour entraîner les modèles de caractères étudiés. En seconde lieu, la classification est réalisée à l'aide d'un classifieur Markovien qui se prononce sur l'appartenance de la forme à une ou plusieurs classes de caractère.

Dans ce qui suit, nous présentons les étapes constituant cette synopsis de notre système de reconnaissance de caractères manuscrits isolés en donnant simultanément les exemples illustrant les deux types de caractères traités : cas de l'arabe et l'amazighe.

## 8.1. Prétraitements

Cette section met l'accent sur les méthodes de prétraitement que nous avons appliqué en vue de produire une version nettoyée de l'image d'origine. Les images utilisées de la base des caractères arabes et amazighes manuscrits isolés, sont souvent entachées de différents types de bruit engendrés par différentes sources (cf. section 3.3 du chapitre 1). L'intérêt des processus qui sont relatés dans ce qui suit, à savoir : binarisation, lissage, squelettisation et normalisation, est la préparation de l'image à l'étape de l'extraction des primitives. Cet intérêt peut se résumer essentiellement en deux points: (1) réduction de la variabilité de l'écriture manuscrite; (2) atténuation ou suppression des informations indésirables. Cependant, il n'y a pas d'opération de segmentation puisqu'on est dans le cas de traitement des formes isolées.

### 8.1.1. Binarisation

La binarisation est un cas particulier de seuillage (cf. section 3.3.2 du chapitre 1), qui vise en principe à classer les pixels de l'image traitée en deux classes (noire et blanche) : les pixels de premier plan associés au texte et les pixels de l'arrière-plan associés au fond. Les images que nous avons utilisé dans notre système peuvent être en différents modes (niveau



de gris, couleurs...). A ce niveau, nous avons opté pour la méthode d'Otsu [Otsu78], qui consiste à déterminer un seuil unique globalement sur toute l'image. Ce seuil doit minimiser la variance intra classe entre les pixels des deux classes précédentes (noire/blanche). La Figure 4-9 ci-dessous présente l'image du caractère arabe Mim (م) avant et après la binarisation. De plus, la Figure 4-10 ci-dessous présente l'image du caractère yae (ه) avant et après la binarisation.



Figure 4-9 : Image du caractère Mim (م) avant et après binarisation

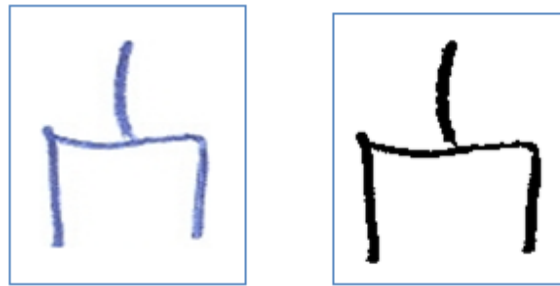


Figure 4-10 : Image du caractère yae (ه) avant et après la binarisation

### 8.1.2. Lissage

Les techniques employées dans cette section consistent à atténuer le bruit et faire disparaître les défauts et perturbations des images des caractères utilisées. En outre, le problème du bruit est très difficile particulièrement dans le cas où l'utilisateur écrit sur un papier ayant un fond complexe. Néanmoins, il semble facile dans notre cas où les images des caractères considérées possèdent toutes un fond blanc. Pratiquement, nous avons implémenté plusieurs méthodes à savoir le moyennage, le médian, le Min/Max [Katk03]. Cependant, nous avons retenu le moyennage qui consiste à améliorer le signal sur le bruit sans perte de résolution. Techniquement cette technique remplace la valeur d'un pixel par la moyenne des valeurs des pixels entourant (et incluant) le pixel d'origine en utilisant le masque :

$$\frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Dans le cas des caractères arabes manuscrits isolés, la Figure 4-11 ci-dessous présente l'image du caractère kaf (ك) avant et après le moyannage.



Figure 4-11 : Image du caractère kaf avant et après le moyannage

Dans le cas des caractères amazighes manuscrits, la Figure 4-12 ci-dessous présente l'image du caractère yae (ⵢ) avant et après le moyannage.

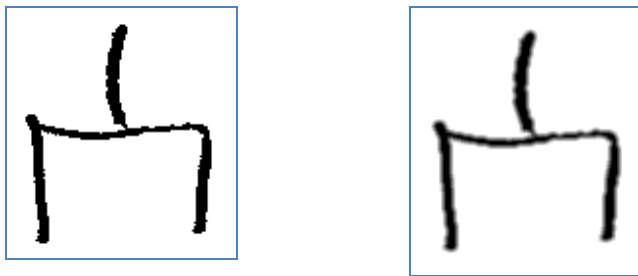


Figure 4-12: Image du caractère yae (ⵢ) avant et après le moyannage

### 8.1.3. Squelettisation

La squelettisation est l'une des techniques les plus utilisées dans la reconnaissance de l'écriture, notamment à des fins de l'extraction de caractéristiques morphologiques. Elle permet de simplifier l'image du caractère en une image plus facile à traiter en la réduisant à une forme avec un épaisseur de 1 pixel tout en conservant ses propriétés topologiques. Il existe de nombreuses méthodes de squelettisation (cf. [section 3.3.3 du chapitre 1](#)). Dans ce travail, nous avons appliqué l'algorithme de Rutovitz [Ruto66], réputé par son efficacité dans la préservation de la connectivité et de la topologie des caractères.

La Figure 4-13 ci-dessous présente quelques exemples de caractères arabes et leurs squelettes obtenus par l'algorithme de Rutovitz. En outre, dans le cas des caractères amazighes, la Figure 4-14 ci-dessous présente les squelettes obtenus pour les lettres yad (ⵢ), yae (ⵢ) et yaw (ⵢ) par le même l'algorithme.



Figure 4-13: Exemples de caractères arabes manuscrits et leurs squelettes obtenus.

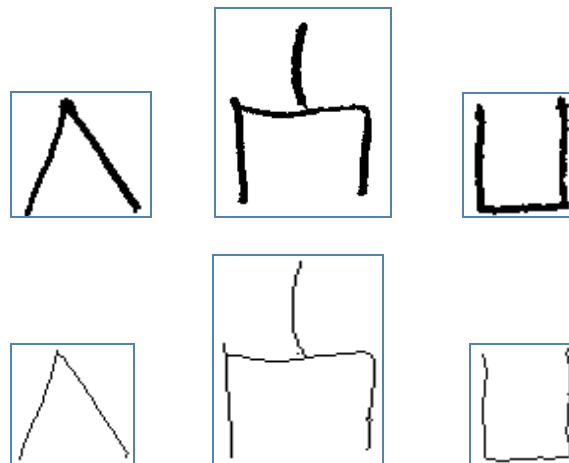


Figure 4-14: Exemples de caractères amazighes manuscrits et leurs squelettes obtenus.

#### 8.1.4. Normalisation de taille

La majorité des images de caractères considérées dans notre travail sont de tailles différentes. Ce qui peut causer une instabilité des paramètres lors de la phase de l'extraction de primitives. Dans le but de minimiser les effets de cette variabilité sur notre système, nous avons appliqué l'algorithme de normalisation de la taille présenté dans [Madh99] (cf. [section 3.3.4 du chapitre 1](#)) pour fixer la taille des images utilisées à 96 x96 pixels. Nous estimons que cette méthode est plus au moins robuste pour l'écriture amazighe.

La Figure 4-15 ci-dessous présente quelques exemples de caractères arabes et leurs tailles normalisées obtenues. La Figure 4-16 ci-dessous présente quelques exemples de caractères

amazighes et leurs tailles normalisées obtenues par l'algorithme de normalisation présenté dans [Madh99].

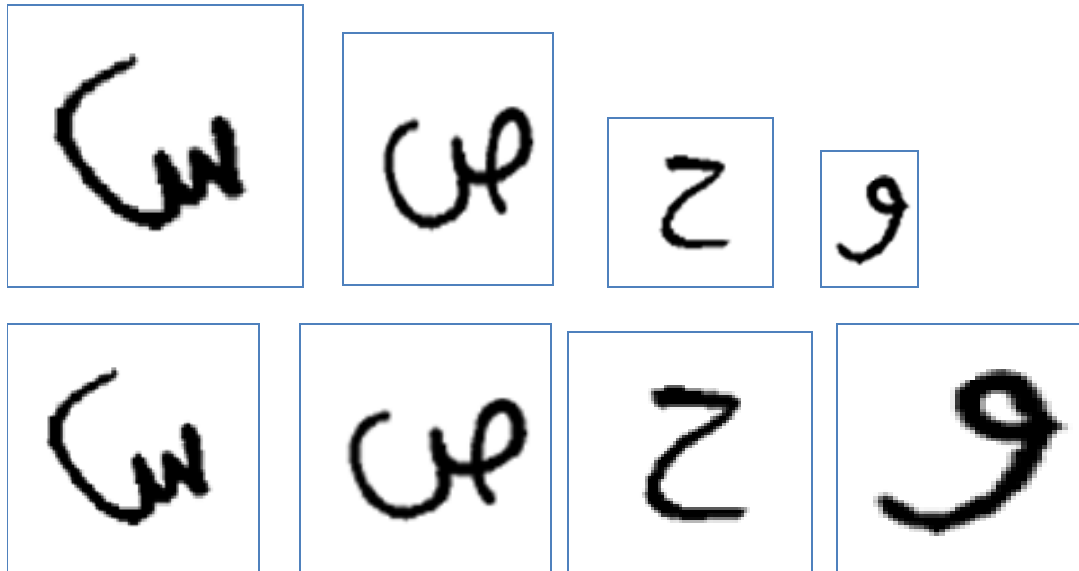


Figure 4-15 : Exemples de caractères arabes isolés manuscrits et leurs tailles normalisées obtenues.

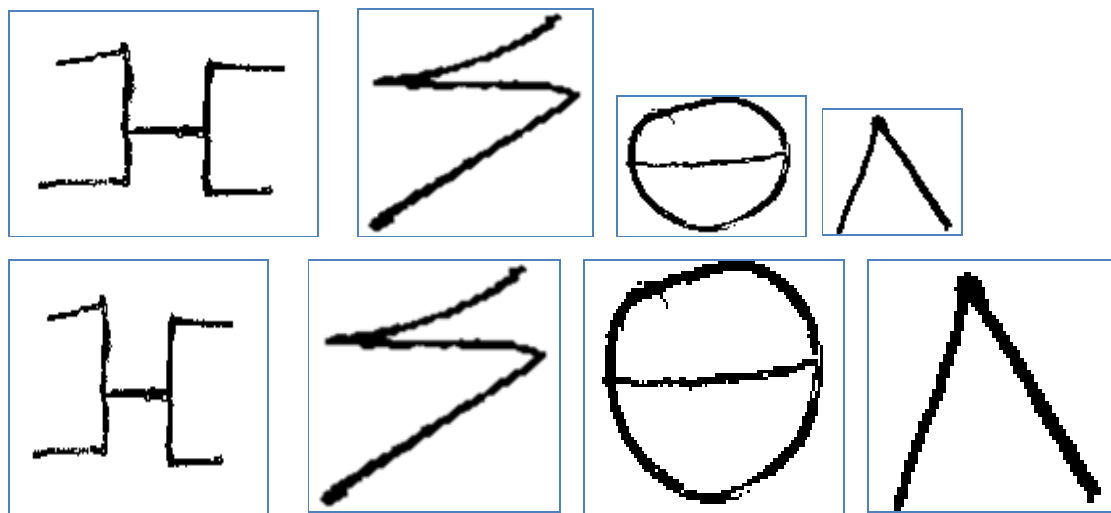


Figure 4-16: Exemples de caractères amazighes manuscrits et leurs tailles normalisées obtenues.

## 8.2. Extraction de primitives

Dans cette phase, on s'intéresse aux techniques transformant l'image de caractère en un vecteur de primitives de dimension fixe. Cependant, l'identification directe du caractère à

partir de son image semble très difficile est même impossible à cause de la morphologie des caractères et de la grande variabilité liée au style d'écriture utilisé et dans certains cas au bruit entachant l'image. D'où la nécessité de sélectionner ou d'obtenir, à partir de la représentation en pixels du caractère, un ensemble de primitives permettant de l'identifier facilement. En outre, ces caractéristiques qui peuvent être existées sous divers types (cf. section 3.5 du chapitre 1), doivent être discriminantes et pertinentes ([Gran00], [Oliv01], [Brit04]).

Dans notre système, nous avons opté pour les primitives directionnelles [Touj02] générées et calculées à partir de la transformation de Hough de l'image de caractère, à l'aide des fenêtres glissantes verticales, divisées horizontalement en zones. Cette transformation a été définie à l'origine par Paul Hough en 1962 [Houg62] pour détecter d'éventuelles lignes droites dans des images en noir et blanc, et donc par conséquent dans des espaces 2D discrets. Elle s'étend cependant facilement à des formes génériques [Duda72], et à des cadres de traitement d'images en niveaux de gris. Les méthodes qui utilisent la transformée de Hough dans le domaine de la reconnaissance d'écriture n'ont cessé d'augmenter depuis deux décennies, car celle-ci est un moyen efficace et robuste d'extraire d'une image des primitives de haut niveau [Gink04].

Nous aborderons alors dans cette section, le principe de fonctionnement de la transformée de Hough, suivi de la démarche que nous avons proposé pour extraire le vecteur de primitives à partir de l'image de caractère.

### 8.2.1. Transformation de Hough

L'intérêt de la transformée de Hough est de détecter dans l'image la présence de courbes paramétriques appartenant à une famille connue (droites, cercles, ellipses ...) en établissant une projection entre l'espace de l'image et un espace représentatif de la forme recherchée, appelé souvent espace de paramètres  $\rho$  et  $\theta$ .

Le fonctionnement de cette transformation repose essentiellement sur l'utilisation d'un accumulateur, qui représente en fait une discrétisation de l'espace de projection précédent. En pratique, cet accumulateur en fait un tableau à  $N$  dimensions où chaque élément va ensuite contenir une valeur qui correspondra au nombre de votes caractérisant alors la présence de la forme concernée dans l'image traitée.

Hough procède alors ainsi : il s'agit de parcourir l'ensemble de pixels de l'image et pour chacun d'entre eux, de déterminer quels sont les vecteurs  $p$  pour lesquels le point (pixel) appartient à la forme correspondante, C.-à-d. qui vérifie son équation mathématique.

Par exemple, dans le cas où la forme paramétrique à détecter correspond à une ligne droite, c'est-à-dire un ensemble de points plus ou moins alignés, cette méthode repose sur le paramétrage de celle-ci par un angle  $\theta$  et une distance  $\rho$  comme définis sur la Figure 4-17 ci-dessous:

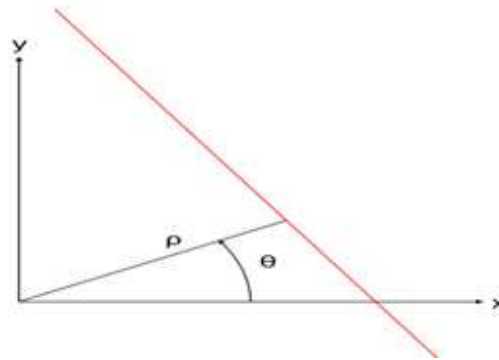


Figure 4-17 : Exemple de ligne droite et les paramètres  $(\rho, \theta)$ .

L'équation cartésienne de la droite est :

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

L'algorithme de Hough utilise une matrice accumulative qui représente le plan  $(\rho, \theta)$ , de dimensions  $(p, q)$  où  $p$  est le nombre de valeurs de  $\rho$  possibles et  $q$  le nombre de valeurs de  $\theta$ . Pour chaque point  $(x, y)$  de l'image binaire traitée, chaque droite  $(\rho, \theta)$  passant par ce point ajoute une unité dans l'élément correspondant de la matrice. À la fin de l'accumulation, les points de la matrice dont la valeur est la plus élevée correspondent à un grand nombre de points alignés sur l'image. Ces droites sont sélectionnées à partir d'un seuil ajustable. La Figure 4-18 ci-dessous montre un exemple de projection de deux points  $p_1(x_1, y_1)$  et  $p_2(x_2, y_2)$  de plan dans l'espace de Hough. L'intersection des deux courbes dans l'espace de Hough démontre l'endroit où une droite passe par ces deux points.

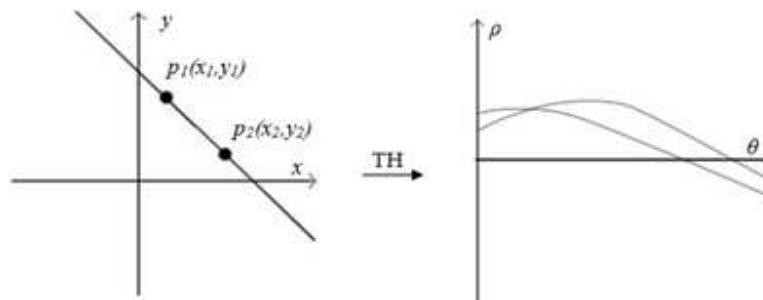


Figure 4-18 : Transformée de Hough de deux points  $p_1$  et  $p_2$  alignés.

La Figure 4-19 ci-dessous montre un exemple de projection d'une ligne de droite dans l'espace de Hough. Ce résultat confirme ce que nous avons mentionné précédemment qu'un point dans l'espace de Hough correspond à un ensemble de points alignés (une droite) dans l'espace d'image.

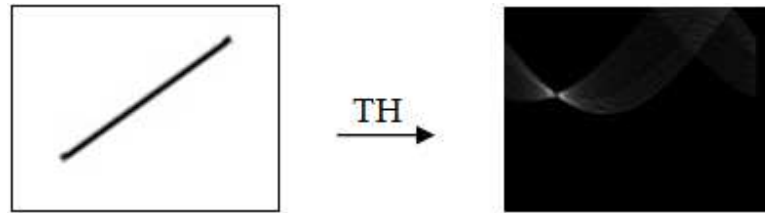


Figure 4-19 : Transformée de Hough d'une ligne droite donnée.

Dans ce travail, les images de caractères étudiées sont binaires (en noir et blanc). Alors, il s'agirait simplement de traiter les pixels représentant la forme de caractère, et d'incrémenter d'une unité chaque cellule de l'accumulateur vérifiant certaines contraintes. Pour se faire, nous avons appliqué sur chaque image de caractère l'algorithme de Hough standard [Mait85] donné ci-dessous :

**Algorithme :**

-Initialisation du tableau d'accumulateur de la Transformée de Hough

-Pour chaque pixel noir de l'image de coordonné  $(x_i, y_i)$   $0 \leq x_i \leq n$  et  $0 \leq y_i \leq m$   
avec  $n$  largeur de l'image et  $m$  sa hauteur

$$0 \leq \theta < 180^\circ$$

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

-Incrémenter la cellule du tableau d'accumulateur correspondant au couple  $(\rho_k, \theta_k)$

La Figure 4-20 ci-dessous présente quelques exemples de caractères arabes et leurs transformations de hough obtenues.

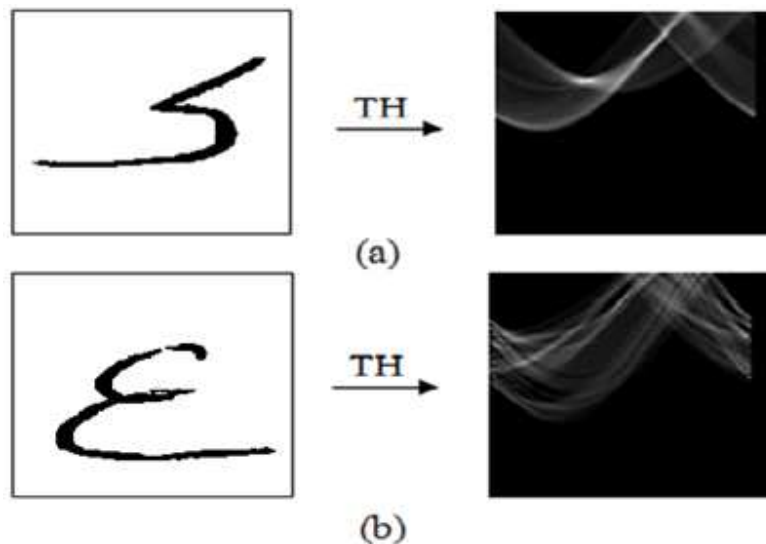


Figure 4-20 : (a) Transformée de Hough de la lettre kaf (b) : Transformée de Hough de la lettre Ain

De plus, la Figure 4-21 ci-dessous présente les transformations de Hough obtenues pour les caractères amazighs (I) et (Y).

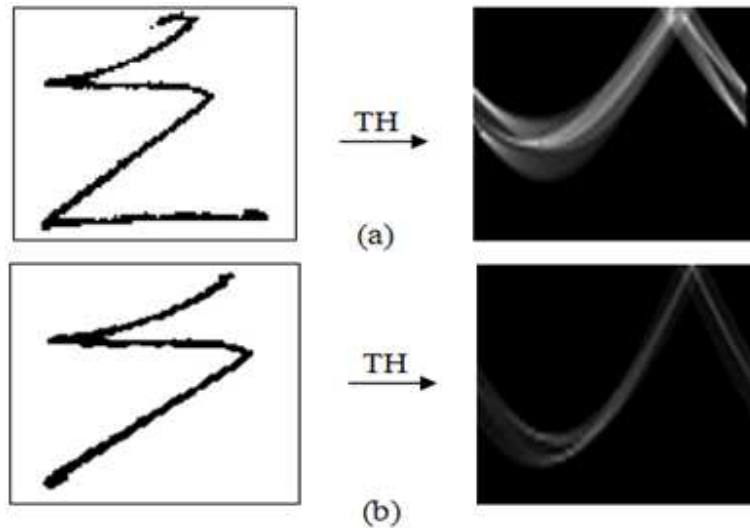


Figure 4-21 : (a) Transformée de Hough de la lettre I (b) : Transformée de Hough de la lettre Y

### 8.2.2. Génération des primitives directionnelles

Afin d'extraire le vecteur de primitives directionnelles, nous avons effectué un ensemble d'opérations sur la transformée de Hough de chaque image obtenue par l'algorithme précédent. Tout d'abord, pour des raisons de simplification, les tailles des images de caractères utilisées ainsi que les tailles de leurs transformées de Hough sont fixées à  $96 \times 96$  pixels. C'est le résultat de la normalisation précédente. Dans les calculs, le pas  $\Delta\theta$  pris est égal 30. Par conséquent, l'accumulateur de Hough contiendra une information sur 6 orientations de ( $0^\circ$ ,  $30^\circ$ ,  $60^\circ$ ,  $90^\circ$ ,  $120^\circ$  et  $150^\circ$ ) (cf. Figure 4-22 ci-dessous).

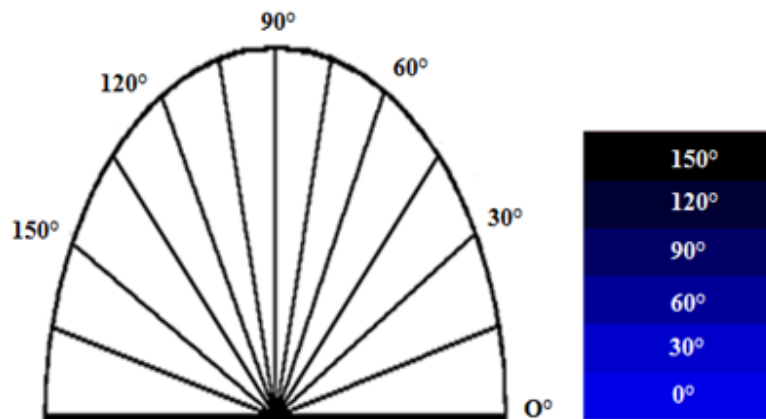


Figure 4-22 : Différentes orientations enregistrées par l'accumulateur de Hough



Par la suite, la représentation en pixel de cet accumulateur pour chaque caractère est analysée en utilisant les fenêtres verticales intercalées dans le sens de l'écriture (de droite à gauche pour l'arabe et de gauche à droite pour la graphie amazighe). Chaque fenêtre est divisée horizontalement en zones. En effet, au total cet accumulateur est partitionné en 16 bandes verticales et 16 bandes horizontales. Les intersections de ces bandes forment ( $16 \times 16 = 256$ ) zones. Chaque zone correspondante à une petite image de taille ( $6 \times 6$ ) pixels. La Figure 4-23 ci-dessous présente un exemple de partitionnement de caractère (Kaf) à l'aide des fenêtres glissantes.

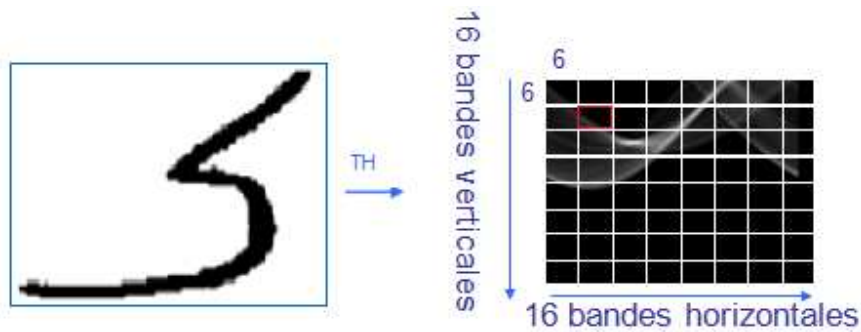


Figure 4-23 : Division de l'image de l'accumulateur de caractère (Kaf) en bandes verticales et zones de taille ( $6 \times 6$ ).

La Figure 4-24 ci-dessous montre le même exemple mais cette fois-ci pour le cas de caractère amazighe yay (y).

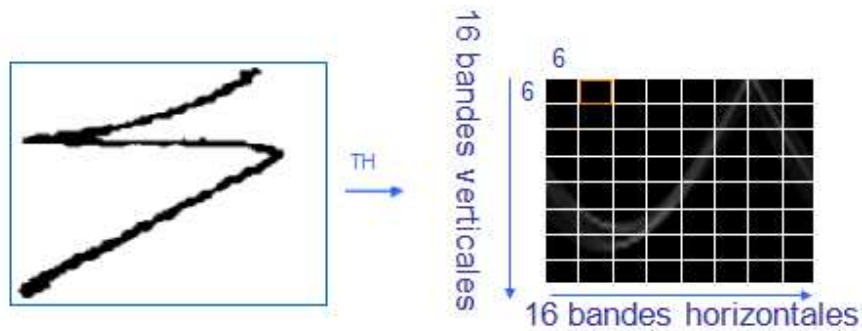


Figure 4-24 : Division de l'image de l'accumulateur de caractère yay (y) en bandes verticales et zones de taille ( $6 \times 6$ ).

La procédure de calcul des primitives continues ainsi, Nous comptabilisons dans chaque zone le taux de présence de chacune des 6 orientations considérées. Ensuite, nous déterminons la direction dominante sur chacune de ces zones c.-à-d. la direction qui possède le taux le plus grand parmi d'autres. Cette information est mémorisée pour la réutiliser dans la génération des observations.

Une fois la direction dominante est retenue, nous allons chercher à l'utiliser comme l'unité de base pour engendrer la séquence d'observations représentative du caractère. En réalité,

cette séquence représentera le vecteur de primitives qui sera utilisé dans les phases d'apprentissage et de classification.

Pour se faire, tout d'abord, dans chaque zone, nous affectons à tous ses pixels la valeur de sa direction dominante déterminée et enregistrée préalablement. Après, les zones sont représentées par leurs taux de direction dominante de haut vers le bas et de gauche à droite dans un vecteur linéaire de longueur 256. C.-à-d. les primitives extraites des 16 zones constituant la première bande verticale seront considérées comme la première sous séquence d'observation dans ce vecteur. En fin, nous parcourons ce vecteur et nous représentons un ensemble de zones adjacentes possédant le même taux par un symbole dans la séquence d'observations. Les symboles qui peuvent être émis sont au nombre de 6 correspondants aux 6 orientations de base étudiées (cf. Tableau 4-1 ci-dessous).

Symbole	Angle
1	0°
2	30°
3	60°
4	90°
5	120°
6	150°

Tableau 4-1 : Correspondance entre symboles émis et angles de base.

La démarche précédente est résumée par le pseudo algorithme ci- après :

**Pseudo algorithme :**

- Calculer la transformée de Hough standard de l'image.
- Partitionner l'image de l'accumulateur en 16 fenêtres verticales et horizontales.
- Comptabiliser dans chaque zone (intersection entre une bande verticale et une bande horizontale) le taux de présence de chacune des 6 orientations considérées.
- Déterminer et mémoriser la direction dominante dans chaque zone.
- Affecter la direction dominante à tous les pixels de la zone.
- Les zones sont considérées de haut en bas et de gauche à droite et représentées par leurs directions dominantes dans un vecteur linéaire.
- Représenter l'ensemble de zones adjacentes possédant le même taux de direction dominante par la valeur de ce score.

La Figure 4-25 ci-dessous illustre cette méthode sur un exemple de caractère donné.

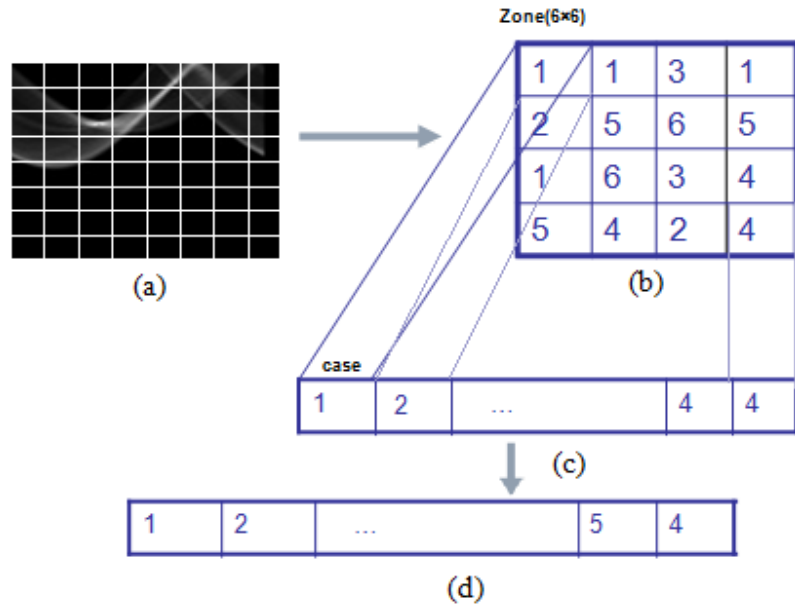


Figure 4-25 : (a) accumulateur de Hough ; (b) : affectation de la direction dominante à tous les pixels de la zone ; (c) : alignement des zone de haut en bas et de gauche à droite dans un vecteur linéaire ; (d) séquence d'observation obtenue.

La Figure 4-26 présente deux exemples de caractères ainsi les séquences d'observations correspondantes générées.

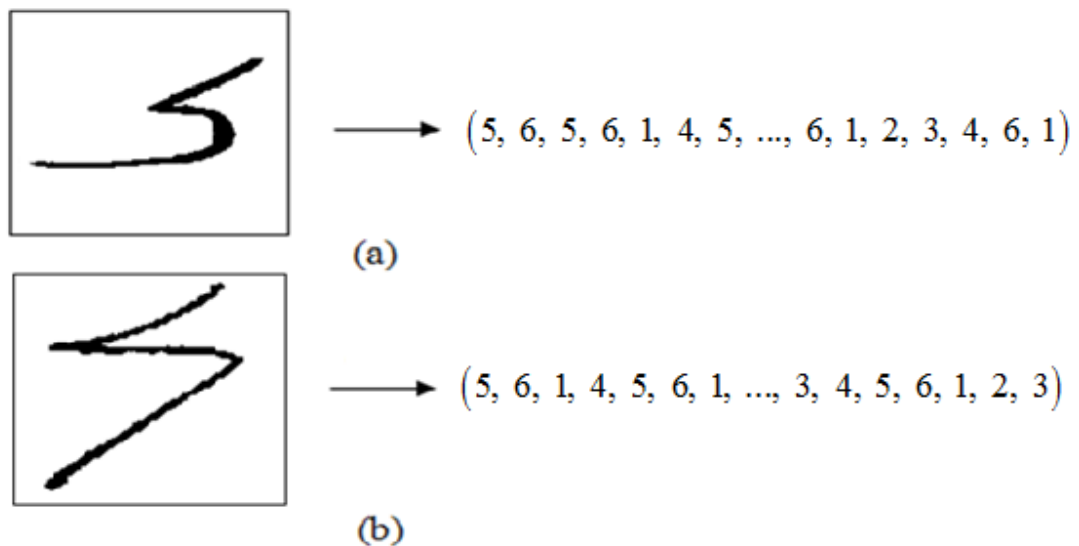


Figure 4-26 : Exemples de caractères et les séquences d'observations générées (a) séquence représentante de la lettre arabe Kaf (b) séquence représentante de la lettre amazighe (y).

## 9. Modélisation markovienne

Grâce à plusieurs avantages au niveau de la modélisation, les modèles de Markov cachés connaissent un grand succès dans la reconnaissance automatique des textes manuscrits. Ils permettent essentiellement d'éviter les problèmes qui sont liés au processus de la segmentation directe des textes en unités plus petites en intégrant la segmentation et la reconnaissance dans un seul module. Ils offrent aussi plusieurs types d'apprentissage, en particulier l'entraînement automatique sans segmentation des caractères à partir des images de mots (apprentissage croisé, en anglais « embedded training ») ([Dupr03], [Beno07]).

Dans notre système, nous avons utilisé une modélisation des caractères par les modèles de Markov cachés de type modèle discriminant. Elle consiste à associer un ou plusieurs modèles par classe. De ce fait, nous obtenons autant de modèles que de classes traitées et La forme à reconnaître est affectée à la classe correspondant au modèle qui maximise la probabilité d'engendrer sa séquence d'observations.

Nous présentons dans ce qui suit, les principales étapes constituant la modélisation markovienne que nous avons développée, qui comprend principalement : le choix des modèles des caractères, leur entraînement, et leur classification.

### 9.1. Choix de modèle

A l'issue de la phase précédente, les images des caractères sont traduites en séquences d'observations pour alimenter les Modèles de Markov cachés modélisant les données d'apprentissage. Or cette modélisation n'est pas toujours une tâche évidente. Elle requiert des connaissances préalables du problème donné. Dans ce sens, on se focalise sur les moyens théoriques permettant de chercher et de sélectionner la topologie des HMMs utilisés afin de modéliser les caractères étudiés. Cette topologie est généralement dépendante de plusieurs aspects à savoir :

- La nature et la taille de l'unité de base. Dans ce cas, nous avons la possibilité de considérer les caractères globalement ou bien d'opérer localement en d'autres unités plus fines C.-à-d., au niveau des bandes ou des graphèmes. Le choix entre l'une ou l'autre est souvent imposé par la taille de vocabulaire de l'application considérée. Or puisque on a affaire à des alphabets de 28 caractères arabes et 33 caractères amazighes, alors cette taille est limitée. De ce fait, il est plus évident d'effectuer une modélisation au niveau lettre, laquelle est relativement plus robuste à la variabilité, et ne requiert pas une segmentation préalable du caractère, qui est souvent très complexe à réaliser.
- Choix des paramètres de modèle HMM : nombre d'états, type de topologie des états et type de fonctions de densités de probabilités. Il n'existe pas une théorie pour les déterminer. La solution est alors empirique ([Kalt93], [Mahj99], [Augu01]). Souvent, le nombre de paramètres dans un HMM doit correspondre aux données d'apprentissage. Ce nombre doit être suffisamment grand pour modéliser la variabilité dans la bas. Mais en revanche, il ne doit pas excéder une limite au delà de laquelle les modèles ne sont plus convenablement appris.

De nombreuses considérations ont donc été prises en compte pour fixer ces paramètres et en particulier, la technique utilisée dans la génération des séquences d'observations. Nous voulons adapter le maximum possible l'architecture des modèles établis au partitionnement adopté pendant cette démarche. Par conséquent, Nous avons opté pour une topologie de type linéaire orientée dans le sens de l'écriture étudiée, 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 il est inférieur ou égale aux nombres de bandes horizontales constituant l'image de caractère sans l'égaliser (fixé à 14 états). En effet certaines bandes surtout celles des extrémités ne fournissent aucune observation.

La Figure 4-27 ci-dessous montre la topologie de type gauche droite dans le cas des caractères arabes.

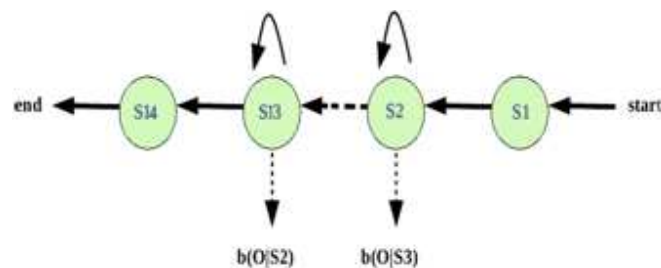


Figure 4-27 : la topologie droite gauche de modèle de caractère arabe

La Figure 4-28 ci-dessous présente la topologie de type gauche droite dans le cas des caractères amazighes.

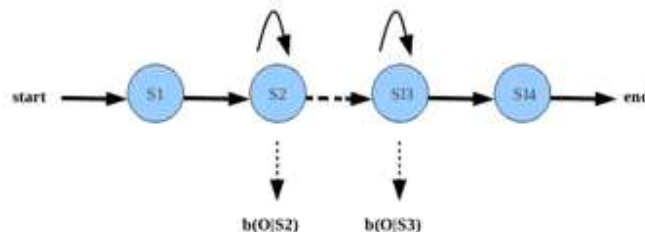


Figure 4-28 : la topologie gauche droite de modèle de caractère amazigh

## 9.2. Le modèle de caractère

L'approche proposée est basée sur la modélisation des caractères par les modèles de Markov cachés (HMMs). Les paramètres décrivant le modèle du caractère  $\lambda_c$  sont: le nombre d'états cachés  $N$ , la matrice des probabilités de transitions entre états  $A$ , la matrice des probabilités d'observations  $B$ . Le modèle de chaque caractère est un HMM discret de 14 états avec une topologie de type linéaire (cf. les Figure 4-27 et Figure 4-28 ci-dessus). Chaque état est associé à une notion logique qui est la bande verticale de l'image. En outre, dans les premières expériences, nous avons utilisé les densités de probabilité des observations discrètes dans chaque état. Cependant, elles sont modélisées par un mélange

de composantes gaussiennes dans les secondes. En utilisant ces paramètres, nous avons déterminé le modèle adéquat du caractère qui a été déterminé par la procédure de Baum-welsh (section suivante).

Au total, pour modéliser l'alphabet arabe, 16 modèles différents ont été construits correspondant aux caractères isolés sans signe diacritique. Dans le cas du script Amazighe, nous avons construit un nombre de modèles qui s'élève à 33 modèles convenant aux lettres de son alphabet.

### 9.3. Apprentissage des modèles HMMs

La problématique majeure d'un système de reconnaissance des caractères manuscrits réside dans l'apprentissage qui consiste à ré-estimer les paramètres des modèles de lettres de l'alphabet utilisé. Cette ré-estimation est réalisée dans le but de capter le maximum possible d'informations relevées de la variabilité des formes représentant chaque classe du caractère manuscrit.

Dans notre système de reconnaissance de caractères manuscrits isolés, tout d'abord, nous avons créé un modèle séparé pour chaque caractère. En suite, les images des caractères de la base d'apprentissage considérée sont traduites en séquences d'observations à l'aide de la phase d'extraction de caractéristiques. De ce fait, nous disposons d'un ensemble de séquences d'observations discrètes  $O = (O^1, O^2, \dots, O^N)$  correspondantes aux caractères des alphabets de reconnaissance (16 classes dans le cas de l'arabe et 33 classes dans le cas de l'amazighe). Le défi relevé à ce niveau, est de déduire les modèles HMMs générant ces séquences d'observations. Une fois la topologie du modèle initial de caractère est choisie, on a recourt à l'apprentissage qui permet l'entraînement de ces modèles initiaux choisis par les séquences  $O$  précédentes afin d'ajuster leurs paramètres. Le modèle de chaque caractère utilise uniquement les échantillons étiquetés de sa classe. Les modèles finaux construits sont associés à l'ensemble des séquences  $O$ .

En pratique, nous avons estimé les probabilités d'entrées, de transitions, d'émissions de chaque modèle de caractère pour modéliser au mieux la base d'échantillons d'apprentissage. Cela, peut se traduire formellement par la recherche des paramètres de chaque modèle  $\lambda_c = \{\pi, A, B\}$  qui maximisent la probabilité  $p(O/\lambda_c)$  de la séquence d'observations donnée  $O = \{o_1, o_2, \dots, o_n\}$ . Pour se faire, nous avons appliqué l'algorithme de Baum-welch (cf.section 3.3.3 du chapitre 2) qui se base sur le principe donné ci-dessous ([Rabi89], [Rabi93]):

- Choisir un ensemble initial de paramètres  $\lambda_0$ .
- Calculer  $\lambda_1$  à partir de  $\lambda_0$ .
- Répéter ce processus jusqu'à un critère de fin.
- Partant de  $\lambda_n$ ,  $\lambda_{n+1}$  doit vérifier :  $\prod_r p(o^r | \lambda_{n+1}) \geq \prod_r p(o^r | \lambda_n)$ .

L'algorithme va donc ré-estimer les paramètres de modèle HMM de chaque caractère, jusqu'à ce que la probabilité  $p(O/\lambda_c)$  de générer la séquence d'observations  $O$  soit maximale.

En fin, le meilleur modèle HMM trouvé est enregistré pour former une base de modèles de référence (cf. Figure 4-29 ci-dessous).

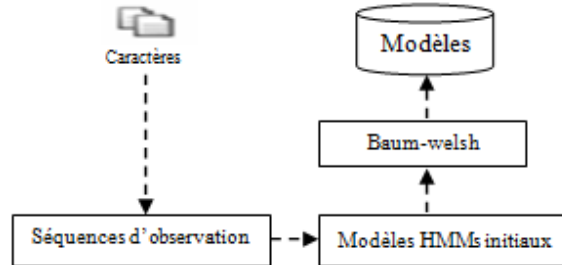


Figure 4-29 : schéma d'apprentissage

A l'aide de la procédure itérative de Baum-welsh, nous avons effectué un apprentissage global, qui est facile à mettre en œuvre. Cette méthode permet de garantir une monotonie croissante de la vraisemblance au cours des itérations, et de ce fait sa convergence vers un point optimal local est évidente. La qualité de la convergence est étroitement liée à l'ensemble initial de paramètres. Nous avons effectué une initialisation uniforme des paramètres de modèles.

#### 9.4. Classification

Dans cette section, la reconnaissance se fait en utilisant l'ensemble des données d'apprentissage. La séquence d'observations décrivant l'image de caractère à reconnaître est alignée sur chaque modèle de la base de référence (résultat de la phase précédente). En effet, nous avons construit durant l'apprentissage autant de modèle HMMs qu'il y avait d'images de caractères dans la base d'apprentissage. Étant donné une séquence d'observations de caractère à tester. Alors la classification se fait par recherche du meilleur modèle engendrant cette séquence parmi tous les modèles HMMs enregistrés dans la base de référence. Pour se faire, nous avons utilisé l'algorithme de Forward pour calculer la vraisemblance de la séquence d'observations de caractère à reconnaître par rapport à chaque modèle de la base [Rabi89]. De ce fait, nous disposons d'un ensemble des scores qui correspondent aux probabilités (*séquence/modèle de classe*). Le modèle élu est celui possédant le plus grand score, c.-à-d., le modèle donnant la vraisemblance maximale identifie le caractère reconnu (cf. Figure 4-30 ci-dessous).

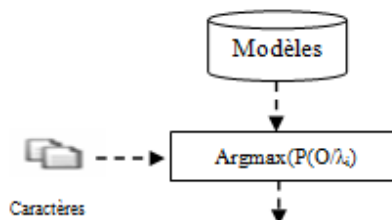


Figure 4-30 : Schéma de classification

## 10. Expérimentations et résultats

Nous avons décrit ci-dessus les différentes phases de notre système de reconnaissance de caractères manuscrits isolés. L'approche développée est indépendante de la nature du script. Elle opère globalement sur la forme du caractère sans segmentation. En effet, nous avons adopté la modélisation de type modèle discriminant basée sur les caractéristiques directionnelles de bas niveaux qui sont extraites sur les images du caractère indifféremment de leur graphie.

Pour évaluer les performances de notre système de reconnaissance, des expériences ont été effectuées sur différentes bases de données représentant les scripts Arabe et Amazighe. Nous présentons dans la suite les bases de données utilisées et les expérimentations effectuées.

### 10.1. Base de donnée utilisée

Pour tester nos algorithmes HMM développés, nous devons disposer de bases de données représentant les objets à étudier. Nous avons utilisé un ensemble de caractères arabes manuscrits isolés et une base de caractères manuscrits amazighes.

#### 10.1.1. Cas de caractères arabes

Nous avons constitué localement une base de caractères arabe manuscrits isolés sans signe diacritique et de différentes tailles. Le nombre de classes considéré est 16. De ce fait, l'ensemble des images construites s'élève à 224 images de caractères (14 caractères pour chaque classe). Un aperçu de certains de ces caractères est présenté dans la Figure 4-31 ci-dessous.

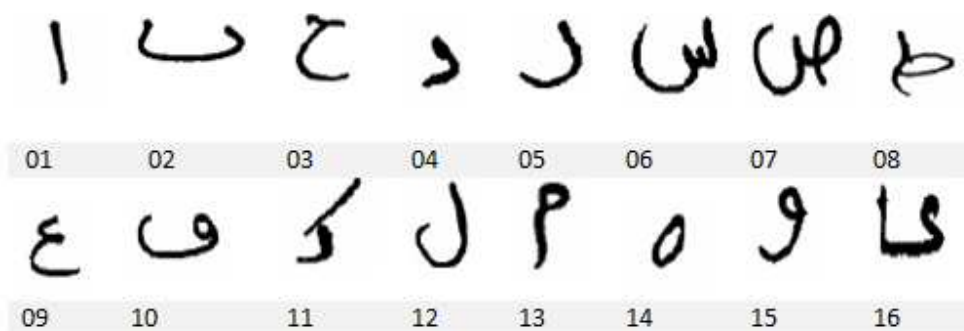


Figure 4-31 : Exemple de modèles de quelques caractères arabes manuscrits isolés sans signes diacritiques

#### 10.1.2. Cas de caractères amazighes

Nous avons utilisé la base de données de caractères amazighes manuscrits (A Database for Amazigh Handwritten Character Recognition Research :AMHCD) créée et développée au sien de notre Laboratoire IRF-SIC de l'Université Ibn Zohr d'Agadir par notre collègue Y.



Es-saady. La base contient 25740 caractères amazighes manuscrits isolés et étiquetés produite par 60 scripteurs. Le lecteur peut trouver une description complète et détaillée sur cette base dans [Essa11].

Jusqu'à présent la base AMHCD est peu utilisée et explorée pour l'évaluation des systèmes de reconnaissance de l'écriture amazighe [Essa12]. En revanche, elle s'est imposée comme la seule et la première base dans sa catégorie (graphie amazighe), grâce à sa taille importante et à sa disponibilité pour les recherches académiques.

Les Tableau 4-2 et Tableau 4-3 ci-dessous présentent des exemples des caractères amazighes manuscrits. Chaque caractère est donné sous forme de quatre variantes qui correspondent aux quatre scripteurs différents.

caractères amazighs imprimés	Scripteur 1	Scripteur 2	Scripteur 3	Scripteur 4

Tableau 4-2 : Exemples de caractères amazighes manuscrits issus de la base AMHCD  
[Essa11]

caractères amazighs imprimés	Scripteur 1	Scripteur 2	Scripteur 3	Scripteur 4
o				
ⵝ				
oio				
ⵉ				
oo				
ⵓ				
ⵔ				
ⵕ				
ⵖ				
ⵗ				
ⵘ				
ⵙ				
ⵚ				
ⵛ				
ⵜ				
ⵝ				

Tableau 4-3 : Exemples de caractères amazighs manuscrits issus de la base AMHCD [Essa11].

## 10.2. Tests et résultats

### 10.2.1. Cas des caractères arabes

Pour évaluer les performances de notre système, nous avons travaillé sur 224 caractères arabes manuscrits isolés (14 pour chaque classe, le nombre de classes est 16) construits localement. 50% des lettres ont servi pour la phase d'apprentissage et 50% ont été utilisées pour les tests. Des résultats encourageants apparaissent. En effet, sur les 112 caractères lus consacrés au test, 96 ont été reconnus, soit un taux de reconnaissance de 85,71%. En ce qui concerne le taux obtenu pour chaque lettre, le meilleur résultat atteint avec cette approche a été de 100 %, pour le caractère (Ha). Le Tableau 4-4 ci-dessous présente le taux de reconnaissance obtenu sur certains caractères.



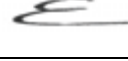
Caractère	Taux de reconnaissance
 (Ha)	100 %
 (KAF)	85,71 %
 (AIN)	71,42 %

Tableau 4-4 : Taux de reconnaissance obtenus sur certains caractères arabes

Les erreurs de reconnaissance sont élevées pour la lettre AIN, ce qui s'explique notamment par l'insuffisance des caractéristiques utilisées pour mieux décrire chaque caractère lors de la phase d'extraction des primitives (très grande variabilité dans les styles d'écriture), et d'autre part, aux données initiales utilisées pendant l'étape d'apprentissage. Une bonne estimation de ces données peut diminuer le taux d'erreur de notre système. Ces résultats ont été publiés dans la revue international Springer [Amro08].

### 10.2.2. Cas des caractères amazighes

Dans cette section, nous avons évalué les performances de notre système sur la base AMHCD avec deux variantes. La première adopte la modélisation discrète des probabilités d'émission, quant à la deuxième utilise les HMMs continus.

A partir de la base AMHCD, nous avons constitué deux parties : apprentissage et test.

Partie de l'apprentissage : nous avons 17160 exemples de caractères, soit 2/3 de la base AMHCD;

Partie de test : 7151 exemples caractères, soit 1/3 de la base AMHCD.

a) *Modélisation par les Modèles de Markov cachés discrets :*

Dans cette expérience, nous avons utilisé seulement un sous ensemble de AMHCD, environ 2220 exemples de caractères. Tout d'abord, nous avons entraîné les modèles initiaux des caractères sur 1480 (2/3) d'exemples de caractères pendant la phase d'apprentissage. Chaque caractère utilise les exemples de sa classe. En suite, notre approche est testée sur le

reste (1/3) des exemples des caractères. Les résultats obtenus semblent satisfaisants. En effet, sur les 740 caractères lus, 669 ont été reconnus, soit un taux de reconnaissance de 90,4%. Le Tableau 4-5 ci-dessous présente les taux des mauvaises affectations et de mauvais rejets.

<b>Taux de mauvaise affectations</b>	7,24%
<b>Taux de mauvaise rejets</b>	2,46%

Tableau 4-5 : Pourcentages des erreurs de reconnaissance

De la même manière que le script arabe, nous estimons que les erreurs proviennent : (1) des primitives directionnelles adoptées lors de l'étape d'extraction. Elles sont moins pertinentes et significatives pour décrire chaque forme ; (2) de la modélisation discrète utilisée. A ce niveau, on a recourt au risque de la perte des informations ; (3) Le choix de modèle initiale de chaque caractère pendant la phase d'apprentissage et aussi la taille des données utilisées pour créer la base des modèles de référence. En fin, une bonne estimation de ces paramètres peut améliorer le taux de reconnaissance de notre approche. Ces résultats ont été publiés dans le journal international ICGST-GVIP [Amro10]. Nous proposons dans ce qui suit une autre expérience en utilisant la deuxième variante continu de notre système avec augmentation de la taille des données utilisées.

*b) Modélisation par les Modèles de Markov cachés continus :*

Dans le cas continu on procède par une modélisation des densités des probabilités par des gaussiennes (cf. [section 3.2 du chapitre 2](#)). Nous avons effectué une série d'expérimentations sur toute la totalité (21450 exemples de caractères) de la base AMHCD. Donc, nous avons employé 17160 (soit 2/3 de AMHCD) exemples d'images de caractères pour l'entraînement et le reste (soit 1/3AMHCD) pour les tests. A l'aide de ces expériences, nous avons évalué le taux de reconnaissance de notre approche en fonction de nombre d'états par modèle HMM et de nombre de gaussiennes. En effet, nous avons utilisé 5 topologies varient entre 6 et 14 états pour étudier l'influence de ce paramètres sur les performances pour une modélisation des émissions par une seule ou deux composantes gaussiennes. Le Tableau 4-6 ci-dessous présente les résultats obtenus sur cette base.

<b>Nombre d'états</b>	6	8	10	12	14
<b>Nombre de gaussiennes</b>	1-2	1-2	1-2	1-2	1-2
<b>Taux de reconnaissance</b>	96,21%	96,56%	96,88%	97,38%	97,89%

Tableau 4-6 : Taux de reconnaissance en fonction des nombre d'états et de nombre de gaussiennes

Nous constatons que le meilleur score obtenu est de 97,89% avec la topologie de nombre d'états égal 14 et le nombre de gaussiennes égal 1 ou 2. Le choix de la topologie influence directement les résultats. En effet, l'augmentation de nombre d'états augmente le taux de reconnaissance de système. En revanche, l'utilisation de deux gaussiennes n'aura pas de

meilleur résultat par rapport au cas mono-gaussien. Néanmoins, l'augmentation de nombre de gaussiennes implique un nombre important de paramètres à calculer.

Les Figure 4-32 et Figure 4-33 ci-dessous montrent l'évolution de taux de reconnaissance en fonction de nombre d'états avec deux ou seule gaussiennes.

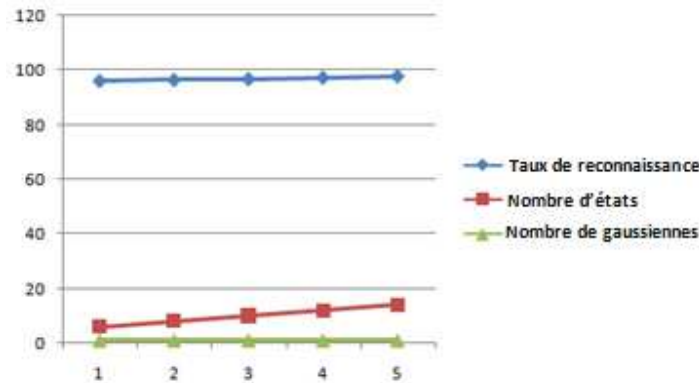


Figure 4-32 : Le taux de reconnaissance par rapport aux états, cas de seule gaussienne

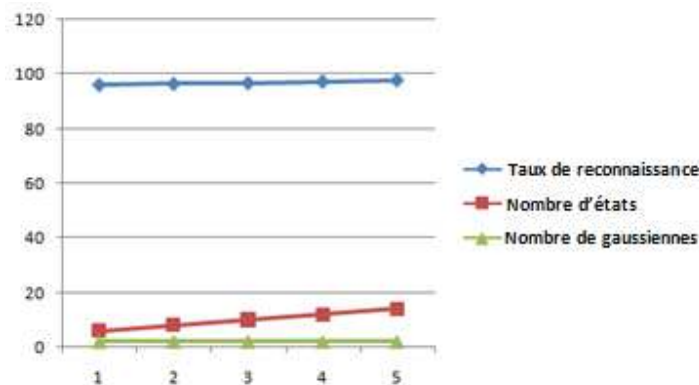


Figure 4-33 : Le taux de reconnaissance par rapport aux états, cas de deux gaussiennes

L'augmentation de la taille des données d'apprentissage et modélisation des densités des probabilités par les gaussiennes ont contribué à diminuer le taux d'erreur commis par notre système. Nous avons passé d'un taux d'erreur global de 9,6% (première expérience avec HMM discrets) à un taux de 2,11% c.-à-d. un gain d'un facteur de 7,49%. De ce fait, les résultats obtenus dans le cas continu sont meilleurs que ceux obtenus dans le cas discret. Ces résultats ont été publiés dans le journal international IJMER [Amro12a].

## 11. Conclusion et perspectives

Dans ce chapitre, nous avons présenté un système pour la reconnaissance de caractères arabes et amazighes manuscrits isolés, en utilisant une approche markovienne de type modèle discriminant basée sur les primitives directionnelles. Après les prétraitements, les

images des caractères étudiées sont transformées en séquences d'observations à l'aide des techniques employées pendant la phase d'extraction de caractéristiques. Le problème que nous avons résolu est le suivant : étant donnée la séquence d'observations représentative d'une image de caractère isolé, nous cherchons à lui attribuer sa classe d'appartenance à l'aide des algorithmes appropriés à base des HMMs, en faisant un alignement de cette séquence sur tous les modèles de la base de référence. Cette dernière a été créée et construite pendant l'apprentissage, en entraînant les modèles initiaux des caractères par les exemples de caractères spécifiques à chaque classes. Les résultats enregistrés montrent l'efficacité de la modélisation proposée, notamment dans le cas continu, où les densités de probabilités sont modélisées par les gaussiennes. En effet, nos expériences qu'ont été menées conduisent à des résultats moins bons dans le cas des caractères arabes, cela revient à la variabilité des formes et les spécificités morphologiques de ses caractères. Cependant, ces résultats ont été obtenus en utilisant une base des données locale, et ne peuvent donc pas se comparer à d'autres méthodes de référence, puisque il faut avoir les mêmes données de test et d'apprentissage. En outre, Dans le cas des caractères amazighes, nous constatons aussi que les scores enregistrés dans le cas des HMMs continus est plus performants que dans le cas des HMMs discrets, puisque dans le cas discret, on perd les informations concernant les images des caractères. Par ailleurs, dans le cas continu, nous avons testé l'influence du nombre d'états et du nombre de gaussiennes sur le taux de reconnaissance : les résultats connaissent une légère croissance avec l'augmentation du nombre d'états. Par contre, les deux modèles mono-gaussiens et deux-gaussiens donnent des résultats semblables.

Les perspectives à envisager pour l'amélioration de ce travail, peuvent se rapportées sur plusieurs aspects, notamment l'intégration d'autres caractéristiques pour enrichir la description des formes pendant l'étape d'extraction. Ainsi, l'augmentation de la taille des données considérées pendant la phase d'apprentissage peut améliorer l'efficacité du système. Ce facteur peut contribuer directement à une amélioration de taux de reconnaissance puisque on étale les chances de représenter les différentes et les variabilités inter et intra formes des caractères.

En outre, dans cette approche et au cours de la génération des séquences d'observations, nous avons adopté une démarche indépendamment de la graphie utilisée. Mais, on peut proposer des techniques exploitants les caractéristiques morphologiques intrinsèques d'une écriture donnée. De plus, nous avons utilisé des algorithmes classiques, en particulier Forward pour la reconnaissance et Baum-welsh pour l'apprentissage. En perspectives, l'application des alternatives plus évoluées de ces procédures, notamment les variantes de Viterbi pour la reconnaissance et l'apprentissage, peuvent améliorer les performances et accélérer le temps de calcul.

Dans le chapitre suivant, nous allons proposer une nouvelle technique basée sur une modélisation markovienne continue de type chemin discriminant conçue spécifiquement pour l'écriture amazighe. En effet, elle se base sur la majorité de ces caractéristiques durant toutes ses phases, à partir de prétraitements jusqu'à la classification.

# Chapitre 5 :Reconnaissance de textes amazighes imprimés par les Modèles de Markov cachés

---

## 1. Introduction

Dans le chapitre précédent nous avons développé une approche de reconnaissance de caractères manuscrits isolés, basée sur une modélisation markovienne de type discriminant. De ce fait, nous avons associé un ou plusieurs modèles par classe. En conséquence, la reconnaissance est généralement s'effectue en estimant les probabilités d'émission de la suite d'observations  $O$  de la forme à reconnaître par les différents modèles préalablement construits. La forme à reconnaître est affectée à la classe dont le modèle qui maximise la probabilité. Bien que, les résultats obtenus par cette approche semblent encourageants, nous soulignons qu'elle ne tient pas en compte des spécificités morphologiques intrinsèques à l'écriture étudiée. En plus, l'approche est sujette au problème de l'insuffisance des caractéristiques utilisées pour décrire chaque forme d'une classe à cause de la technique adoptée, qui se base sur les primitives directionnelles.

Nous présentons dans ce chapitre un système de reconnaissance de caractères imprimés, basé sur une nouvelle approche qui exploite les caractéristiques et les spécificités morphologiques de la langue amazighe. La solution apportée adopte une modélisation markovienne de type chemin discriminant, optimisée par des algorithmes fondés sur la programmation dynamique ([Kim96], [Case96]). L'approche s'appuie sur la proposition d'une nouvelle liste des segments, qui se compose d'un ensemble de traits fondamentaux constituant les caractères amazighes. Ceci permet de mieux exploiter la redondance de ces traits dans les tracés des lettres amazighes. La description de la structure des caractères repose sur ces éléments.

Les caractéristiques exploitées sont extraites à partir des tracés des caractères par une technique de localisation implicite des segments qui le composent. Pour se faire, nous avons utilisé les points d'intérêts des squelettes. Dans la phase de classification nous avons déployé le chemin discriminant fondé sur la programmation dynamique opérant au niveau de graphe des segments.

Notre système est développé sous forme de deux alternatives différentes. L'amélioration apportée au niveau de la deuxième alternative concerne uniquement les phases : pré classification et extraction des primitives.

La première version consiste à adopter un algorithme sélectif basé sur un indice de classification pendant l'étape du pré-classification, et la technique de nombre de transition noire et blanc entre pixels d'un voisinage  $V_8$  pour localiser les points d'intérêts durant



l'extraction de primitives. Les résultats enregistrés par cette variante seront comparés à ceux obtenus par la deuxième version qui utilisera la transformée de hough pendant la pré-classification et une nouvelle technique à base de la déviation maximale pour détecter les points d'intérêts sur les squelettes.

Nous présentons par la suite, la liste des segments de base des caractères tfinaghs déployés par notre système. Nous noterons que les deux termes graphèmes et segments sont employés pour désigner les mêmes notions. Après, nous détaillons les composantes de la première version du système et l'approche utilisée dans ses différentes phases. Nous donnons les premiers résultats expérimentaux obtenus sur la base des patterns de la graphie amazighe [Aito09]. En suite, nous proposons les modules modifiés pour la version améliorée de notre système avec les résultats obtenus sur la même base de données, en les comparant avec ceux obtenus par la variante antécédente. En fin, nous présentons la conclusion et les perspectives à envisager pour notre approche.

## 2. Graphèmes de base du caractère amazighe

La modélisation Markovienne que nous avons mise en place repose sur une description des caractères amazighes en primitives élémentaires. En effet, l'analyse de la morphologie de l'alphabet Tfinagh (cf. [section 7.2 du chapitre précédent](#)) révèle certaines particularités intéressantes, en particulier la redondance des segments horizontaux, verticaux et diagonaux dans la majorité des lettres ; ainsi que la redondance des formes circulaires qui ne se différencient que par la présence et la position d'un trait ou de point. La Figure 5-1 ci-dessous illustre sur quelques caractères les différents segments de base de l'alphabet tfinagh.



Figure 5-1 : les graphèmes fondamentaux observés sur les caractères «ⴰ», «ⴁ», «ⴃ», «ⴄ», «ⴅ» et «ⴆ»

En se basant sur ces spécificités, nous proposons une liste contenant 10 graphèmes fondamentaux qui constituent les traits de la structure de caractère tfinagh (cf. Tableau 5-1 ci-dessous). Cette liste a été construite automatiquement à partir de la base de données des patterns amazighes imprimés [Aito09].

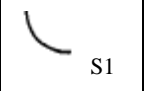
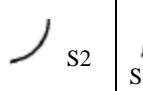
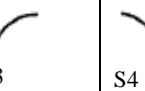
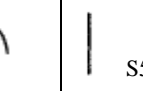

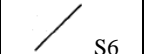
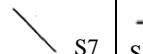



 S1	 S2	 S3	 S4	 S5
 S6	 S7	 S8	 S9	 S10

Tableau 5-1 : Liste de graphèmes de base

Les caractères tfinaghs sont théoriquement et visuellement constitués de segments et arcs élémentaires. L'utilisation de ces graphèmes fondamentaux pour décrire la structure d'un caractère Tfinagh constitue une démarche naturelle. De ce fait, tout caractère de l'alphabet amazighe peut donc être décrit de façon unique et complète par la liste des graphèmes qui le composent. La Figure 5-2 ci-dessous montre les différents segments constituant les tracés des caractères amazighes imprimés « yae », « yaz » et « yak » selon cette liste.

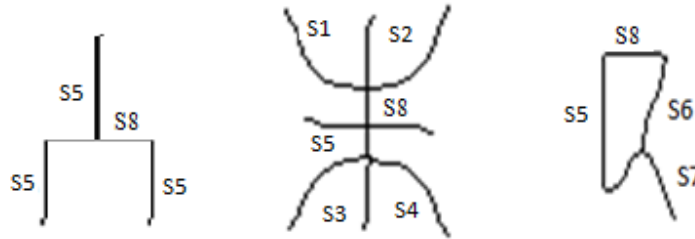


Figure 5-2 : Décomposition des caractères « yae », « yaz » et « yak » en graphèmes (s : segment)

À partir de la liste précédente, nous présentons dans le Tableau 5-2 ci-dessous les décompositions appropriées pour quelques caractères de l'alphabet Tfinagh.

Lettres	décomposition	Lettres	décomposition
○ →	⌒ + ⌒ + ⌒ + ⌒	⦿ →	⌒ + ⌒ + ○ + ○
○ →	○	+ →	— +
⊖ →	○ + —	⌈ →	— +
⊙ →	○ + — + ○	∧ →	/ + \
⊙ →	○ + •	∥ →	+ / +
⋈ →	⌒ + ⌒ +   + ⌒ + ⌒	⋈ →	/ + \ +

Tableau 5-2 : Quelques caractères avec leurs décompositions appropriées

La définition de cette liste est fondamentale pour le bon fonctionnement de notre système de reconnaissance, puisque la description des caractères repose sur ces éléments. Un soin tout particulier a été apporté à l'établissement de ces classes de primitives, notamment pour rendre cette liste aussi exhaustive que possible. C.-à-d., la liste a été établie de façon à couvrir la totalité des formes de traits rencontrés dans l'alphabet tfinaghe. Ce point est crucial, car le principe de définition des primitives, qui permet de se passer de toute segmentation du tracé imprimé à reconnaître, impose que les caractères doivent être composés exclusivement de ces primitives.

La segmentation directe des caractères imprimés en ces graphèmes constitue un problème très complexe. Par ailleurs, nous constatons la stabilité de ces graphèmes dans les tracés des caractères tifinaghs quel que soit le style de l'écriture. En s'appuyant sur ce constat, nous proposons de les considérer directement comme des entités indivisibles de notre modélisation markovienne.

### 3. Synopsi du système développé

Le système de reconnaissance de caractères tifinaghs que nous avons développé est basé sur une architecture simplifiée illustrée par la Figure 5-3 ci-dessous.

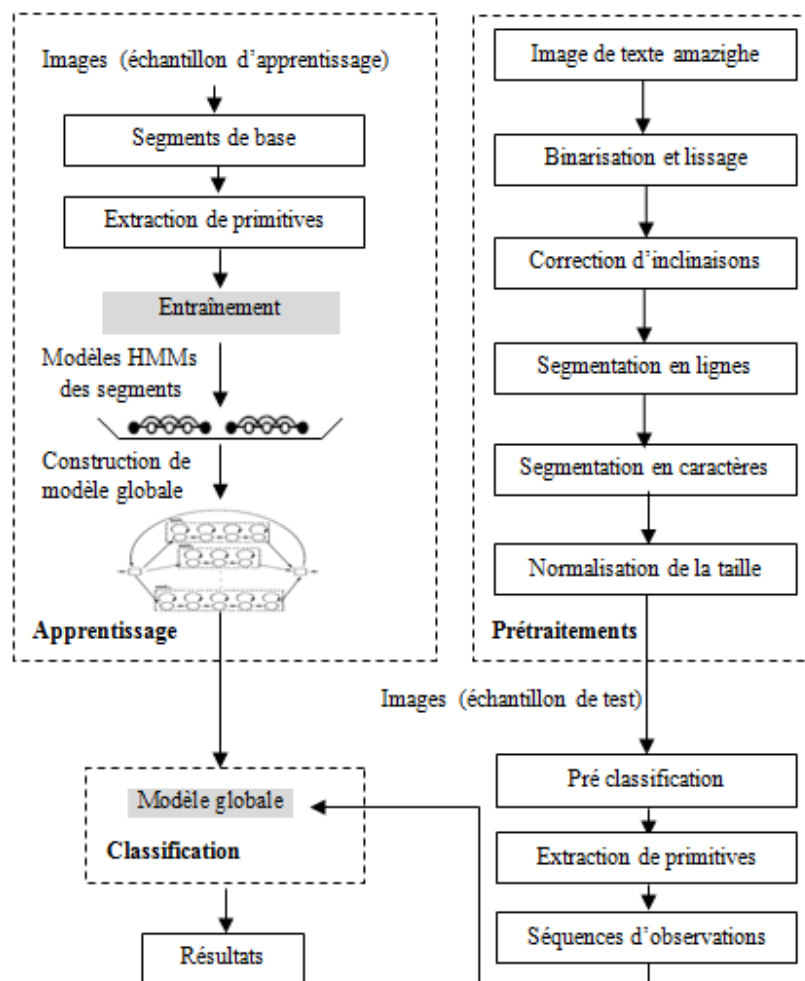


Figure 5-3 : Synopsi de système développé

Cette architecture est de type reconnaissance analytique avec segmentation implicite et apprentissage local au niveau des graphèmes, à travers laquelle le système fonctionne en 2 phases cruciales : l'apprentissage et la reconnaissance [Dupr03]. Chacune d'elles inclue un ensemble d'étapes : prétraitement, normalisation, pré classification et extraction de

primitives. En effet, dans la phase des prétraitements, nous avons adopté plusieurs techniques qui consistent pratiquement à réduire le bruit et la variabilité de l'écriture manuscrite à savoir essentiellement : la technique basée sur l'histogramme de projection horizontale pour corriger l'inclinaison des lignes de texte et les techniques pour segmenter le texte en caractères isolés. Quant à l'étape d'extraction des primitives, notre système utilise les caractéristiques structurelles extraites des graphèmes de chaque caractère. Ces graphèmes sont exprimés ensuite par un ensemble de modèles. Dans l'apprentissage, nous avons construit un modèle globale ergodique de tous ces graphèmes. Le meilleur alignement à travers ce modèle sera déterminé pendant la classification. Nous détaillons les composantes principales de cette architecture dans les sections suivantes.

## 4. Prétraitements

À l'instar des prétraitements présentés dans le chapitre précédent, ceux de cette section visent aussi à se focaliser et à ne garder essentiellement que l'information significative et discriminante pendant la phase de l'extraction de primitives, toute en supprimant le bruit et en réduisant la variabilité des styles de l'écriture. Dans ce cadre, nous avons utilisé une série d'opérations sur l'image de caractère à étudier, comprenant principalement : la binarisation, la réduction de bruit, la correction d'inclinaison des lignes de texte, la segmentation d'un texte en caractères, et enfin la normalisation de la taille et la squelettisation.

### 4.1. Binarisation

De la même manière que le chapitre précédent, nous avons appliqué la méthode d'Otsu [Otsu78], qui consiste en premier lieu à effectuer une analyse pour déterminer la valeur de seuil globalement sur l'image, en seconde lieu, elle classe tous les pixels dont les valeurs sont au-dessus de ce seuil comme étant de pixels d'arrière plan, et tous les autres pixels comme étant de pixels de texte.

La Figure 5-4 ci-dessous présente le résultat obtenu avec la méthode d'Otsu sur un texte amazighe. Ainsi que la Figure 5-5 ci-dessous montre le résultat de binarisation Otsu sur le caractère « $\emptyset$ ».

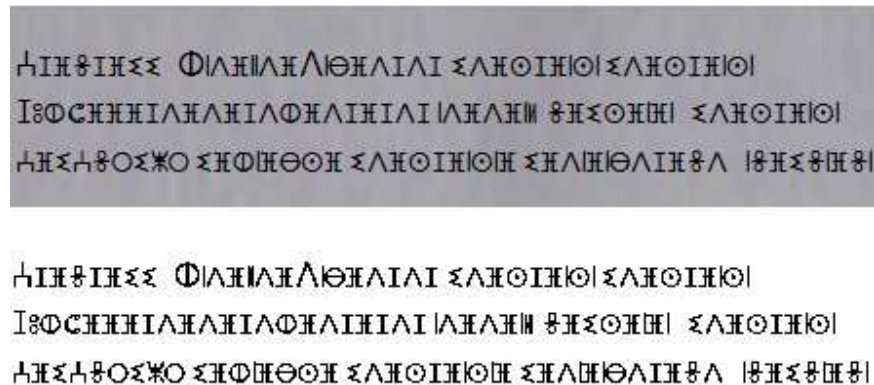


Figure 5-4 : l'image du texte amazighe et sa binarisation avec la méthode d'Otsu

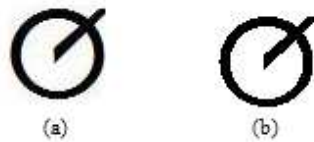


Figure 5-5 (a) le caractère «Ø»; (b): binarisation par la méthode d'Otsu (seuil= 0.8196)

Dans ce travail, le fond des caractères et des textes étudiés est simple et de bonne qualité ce qui convient à la méthode d'Otsu. Néanmoins, elle n'est plus applicable lorsque ce fond est bruité ou non homogène, dans ce cas des taches parasites peuvent apparaître.

#### 4.2. Lissage ou réduction du bruit

Après la binarisation, on peut avoir des points n'appartenant pas aux tracés des caractères. Ces points sont du bruit pouvant provenir par exemple de la binarisation ou bien de l'acquisition. Pour réduire ce bruit, plusieurs techniques existent, dans ce travail nous avons appliqué de manière similaire que le système précédent le filtre moyenneur [Katk03], qui consiste à remplacer la valeur d'un pixel par la moyenne pondérée de ses voisins.

#### 4.3. Correction d'inclinaison des lignes de texte (Skew correction)

L'inclinaison des lignes de textes est définie par une distorsion de ces lignes par rapport l'axe horizontal (cf. section 3.3.4.2 du chapitre 1). De ce fait, les techniques qui visent à corriger cette déformation revient à mettre horizontalement chaque ligne de l'écriture inclinée. Ces algorithmes reposent tous sur le même principe suivant : détection et estimation de l'angle de l'inclinaison puis correction à l'aide de la rotation par cet angle. Dans notre système, nous avons utilisé la technique basée sur les histogrammes de projection horizontale [Vinc01]. En effet, nous avons calculé le nombre des pixels noirs selon des orientations qui sont proches de l'horizontale pour chaque ligne. La qualité de l'histogramme, ou sa pertinence, est estimée par son entropie. La direction la plus probable est celle qui maximise l'entropie. L'histogramme d'entropie maximale est celui dont les extremas sont les plus marqués. L'angle de la ligne  $\theta$  est celui qui correspond à

l'histogramme d'entropie maximale. Pour corriger cette inclinaison, il suffit d'appliquer une rotation de l'image d'angle  $\theta$ .

La Figure 5-6 ci-dessous présente la correction de l'inclinaison des lignes à l'aide de la technique de l'histogramme de projection horizontale pour un texte amazighe incliné.

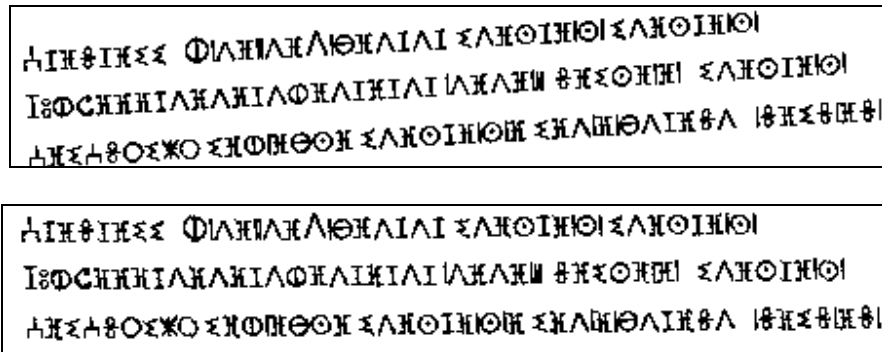
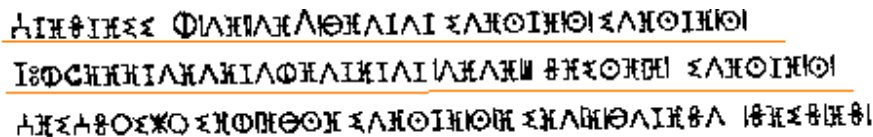


Figure 5-6 : Correction de l'inclinaison des lignes de texte amazighe

#### 4.4. Segmentation d'un bloc de texte amazighe en lignes

Dans cette partie, on s'intéresse aux techniques permettant la séparation des différentes lignes du texte, en vue de faciliter d'autres niveaux de segmentation (cf. [section 3.4 du chapitre 1](#)). La segmentation du texte en lignes n'est pas un problème facile à résoudre, notamment dans certains cas des graphies, où le chevauchement et l'interférence des ascendants et des descendants des lignes voisines sont énormes. Or le texte amazighe ne possède pas d'ascendants et descendants. De ce fait, les lignes en écriture amazighe forment des régions de forte densité séparées par des régions de faible densité. Donc, il est évident dans ce cas qu'une simple application de la technique d'histogramme de projection horizontale [Yin08], permet de localiser chaque ligne du texte.

La Figure 5-7 ci-dessous présente un exemple du texte amazighe avec son histogramme de projection horizontale correspondant. Les maximas et minimas sont déterminés et les composantes connexes entre deux minimas consécutifs forment généralement les lignes cherchées.



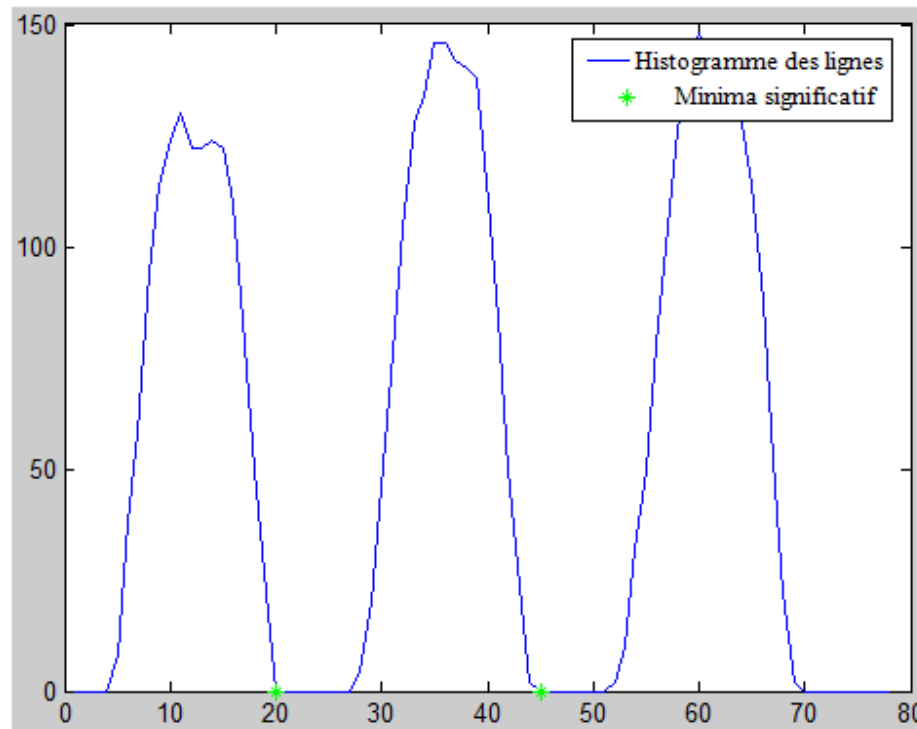


Figure 5-7 : Séparation de ligne du texte par la technique d'histogramme de projection horizontale

#### 4.5. Segmentation d'une ligne de texte amazighe en caractères

À l'issue de l'étape précédente, nous avons obtenu des lignes séparées sous forme des images. En conséquence, nous allons chercher à segmenter chaque ligne du texte en caractères qui le composent, ce qui est difficile dans certains cas du fait que les frontières d'un caractère sont parfois absentes ou floues. Or l'écriture amazighe n'est pas cursif, donc de type scripte c.-à-d., une écriture à lettres séparées ce qui facilite la segmentation d'une ligne de ces lettres.

Dans le cadre de ce travail, nous avons utilisé la technique d'histogramme de projection verticale pour segmenter chaque ligne de texte en caractères.

La Figure 5-8 ci-dessous présente une ligne de texte, son histogramme vertical et le résultat obtenu de la segmentation en caractères.

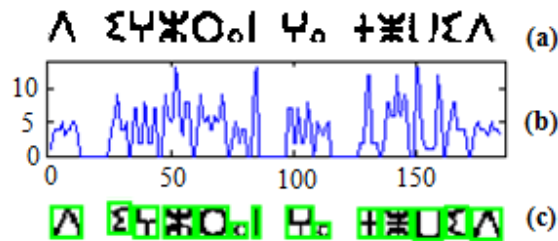


Figure 5-8 : (a) une ligne du texte, (b) Histogramme de projections verticales, (c) le résultat de la segmentation en caractères

#### 4.6. Normalisation de la taille et recadrage

Dans notre système, la normalisation se résume en 2 étapes. Les images de caractères sont subies d'abord une opération qui consiste à supprimer le vide préexistant entre leurs bords et les contours des caractères eux même de telle sorte à avoir des objets recadrés. Ensuite, elles sont transformées en images de même taille (48\*48) pixels, en utilisant l'algorithme de normalisation de la taille proposé dans [Srih97]. La Figure 5-9 ci-dessous présente les résultats obtenus par ces deux algorithmes sur les caractères amazighes imprimés «yaz » et «yak ».

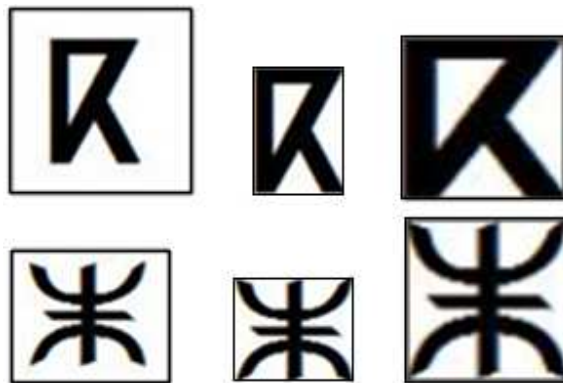


Figure 5-9 : Résultats de l'algorithme de suppression de vide et normalisation de la taille

#### 4.7. Squelettisation

L'approche proposée dans ce travail repose essentiellement sur la détection des points d'intérêts sur le squelette. De ce fait, il est nécessaire de trouver le squelette de l'image avant d'aborder les modules ultérieurs. Généralement, un algorithme de squelettisation à pour principe de ramener l'image du caractère à une écriture linéaire d'une épaisseur égale à un pixel, en préservant la topologie et la connexité du tracé. Il permet d'extraire des caractéristiques importantes, comme les intersections et le nombre de tracés, leurs positions relatives, longueur totale, distance entre parties de la forme. La représentation en squelette facilite l'expression des relations structurelles dans la forme et réduit d'un part, et diminue le temps de traitement en simplifiant la structure de données d'autre part. Nous avons



employé l'algorithme de Hilditch [Hild69] qui exploite le principe d'affinage successifs : la forme est "épluchée" de manière itérative, en maintenant valide le critère de connexité. La Figure 5-10 ci-dessous présente quelques exemples de caractères amazighes et leurs squelettes obtenus par l'algorithme de Hilditch.

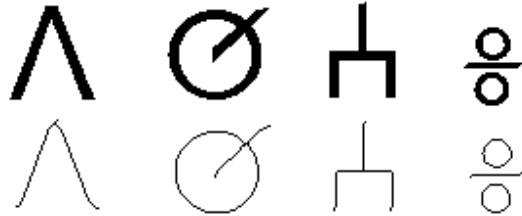


Figure 5-10 : Exemples de squelettisation

## 5. Pré classification des caractères amazighes

Notre méthode est basée sur les points d'intérêts de squelette. Comme certaines caractères ne les possèdent pas, notamment ceux qui contiennent seulement les formes circulaires, alors l'idée principale est de pré-classer les images de caractères en 2 groupes : formes circulaires  $EC=(\circ, \odot, \oplus, \ominus, \otimes, \oslash, \otimes, \Theta, \mathfrak{C}, \mathfrak{Q}, \mathfrak{O})$  et formes non circulaires  $ES=(\varepsilon, \xi, \mathfrak{H}, \mathfrak{L}, \mathfrak{A}, \mathfrak{K}, \mathfrak{M}, \mathfrak{C}, \mathfrak{U}, \mathfrak{T}, \mathfrak{I}, \mathfrak{J}, \mathfrak{E}, \mathfrak{E}, \mathfrak{X}, \mathfrak{I})$ . Dans la littérature, il existe plusieurs méthodes qui répondent à cette question [Duda72][Ety06]. Dans ce cadre, nous avons choisi de résoudre cette problématique par deux méthodes différentes :

la première porte sur un algorithme sélectif basé sur la combinaison des dérivées secondes et le nombre de point d'intérêt de la courbe de tracé des caractères. En revanche, la deuxième méthode repose sur la transformation de Hough [Mait85]. Elle semble efficace et plus usuelle pour ce type de problème. Cette méthode sera présentée par la suite dans la section (11) consacrée à la version améliorée de notre système. En effet, l'algorithme adopté dans la première méthode est donc organisé en niveau successif de décision permettant le filtrage progressif des décisions et réduction de l'ambiguïté on utilisant le nombre de point d'intérêt :  $N_{pi}$  et l'indice de classification obtenu par :

$$\omega = \frac{\sum_{i=1}^{N-1} \sqrt{C_{o u r b e}^2}}{N} \quad (5.1)$$

La Figure 5-11 ci-dessous donne la structure fonctionnelle de notre algorithme. Les caractères de rayon  $R_1$  seront renvoyés dans  $EC_1$  (ensemble circulaire) pendant le niveau 1. En revanche, dans le niveau 2, les caractères de rayon  $R_2$  et qui possèdent au plus deux points d'intérêts seront rassemblés dans  $EC_2$ .

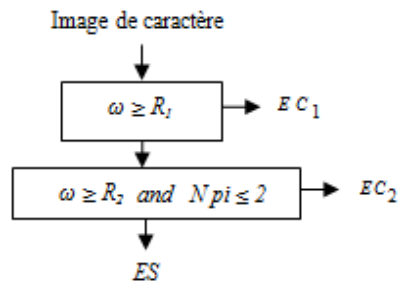


Figure 5-11 : Diagramme de l'algorithme sélectif proposé

L'algorithme fonctionne ainsi : le premier niveau consiste à sélectionner les boucles dotant d'un indice supérieur ou égale au rayon fixé  $R1=Hauteur/2$  qui correspond aux caractères (ⵓ, ⵔ, ⵕ, ⵖ, ⵗ, ⵘ) (cf. Figure 5-12-(a) ci-dessous). Cependant, dans le deuxième niveau, nous avons fixé de la même manière un autre rayon  $R2=Hauteur/4$  pour les caractères (ⵙ, ⵚ, ⵛ, ⵜ), mais l'utilisation de paramètre  $\omega$  uniquement conduit à des confusions entre les formes (ⵙ, ⵚ, ⵛ, ⵜ, ⵝ, ⵞ, ⵟ, ⵠ, ⵡ) (cf. Figure 5-12-(b) ci-dessous). Ceci appuie l'idée de localisation des boucles sur laquelle fondé notre algorithme. Afin de résoudre cette ambiguïté, nous avons proposé de combiner dans ce stade l'indice  $\omega$  et le nombre  $N_{pi}$  total des points d'intérêt de la forme pour filtrer les formes qui possèdent plus que 2 points d'intérêts (cf. Figure 5-12-(c) ci-dessous).

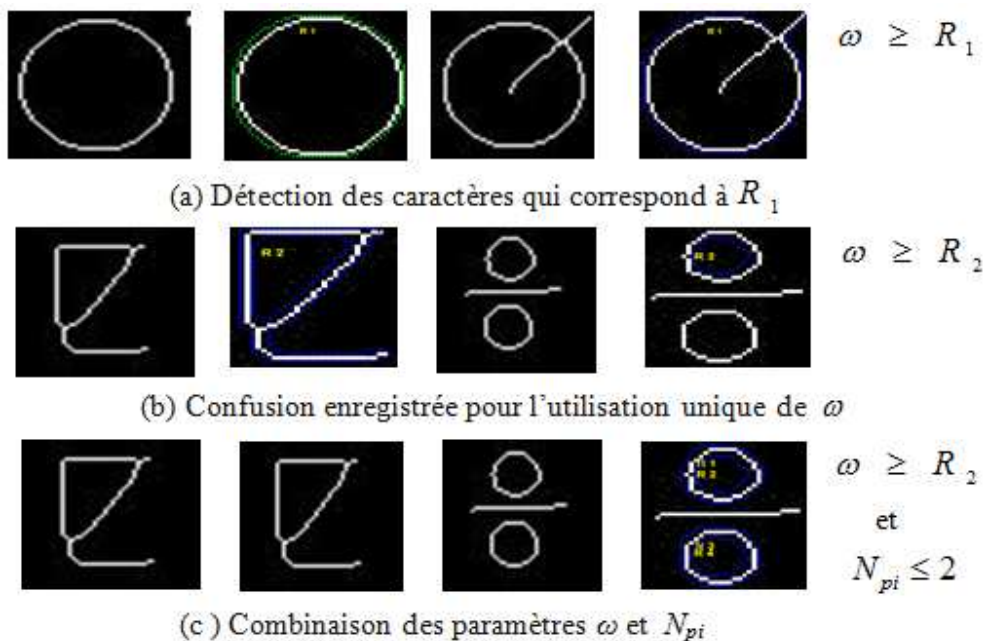


Figure 5-12 : Résultat obtenu par notre algorithme

A l'issue de cet algorithme sélectif, nous avons obtenu deux classes de caractères suivantes :

- $EC = \{EC_1 = (\text{ⵓ, ⵔ, ⵕ, ⵖ, ⵗ, ⵘ}), EC_2 = (\text{ⵙ, ⵚ, ⵛ, ⵜ})\}$ ;

- $ES = \{ \text{ⵎ, ⵏ, ⵐ, ⵑ, ⵒ, ⵓ, ⵔ, ⵕ, ⵖ, ⵗ, ⵘ, ⵙ, ⵚ, ⵛ, ⵜ, ⵝ, ⵞ, ⵟ, ⵠ, ⵡ, ⵢ, ⵣ, ⵤ, ⵥ, ⵦ, ⵧ, ⵨, ⵩} \}$ .

En outre, nous avons poursuivi ce processus de filtrage pour les caractères de groupe  $EC$ . En effet, nous avons proposé une technique qui s'appuie sur les composantes connexes pour classer les lettres selon le nombre  $N_{cc}$  de composantes connexes qui le composent. La Figure 5-13 ci-dessous résume cette démarche. Les caractères avec une seule composante connexe sont regroupés dans  $EC_1\_ICC$  et  $EC_2\_ICC$ . Cependant, les caractères avec deux composantes connexes sont regroupés dans  $EC_1\_2CC$  et  $EC_2\_2CC$ .

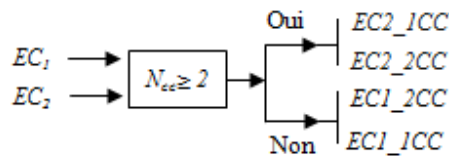


Figure 5-13 : Algorithme de composantes connexes

Finalement, nous avons obtenu les groupes de caractères suivants :

- $EC_1\_1CC = \{ \text{ⵓ, ⵏ, ⵐ, ⵒ, ⵓ} \}$ ;
- $EC_1\_2CC = \{ \text{ⵔ} \}$ ;
- $EC_2\_1CC = \{ \text{ⵖ, ⵗ} \}$ ;
- $EC_2\_2CC = \{ \text{ⵙ, ⵚ} \}$  et  $ES$ .

En outre, nous avons utilisé le nombre de point d'intérêt  $N_{pi}$  pour séparer le groupe  $EC_2\_1CC$  en deux autres groupes  $EC_{21\_1CC} = \{ \text{ⵖ} \}$  et  $EC_{22\_1CC} = \{ \text{ⵗ} \}$ .

Par la suite, chaque classe a subi un traitement approprié à sa morphologie avant d'être soumise au module commun d'extraction. Les traitements qu'ont été effectués sont : l'extraction des composantes connexes pour les caractères de la classe  $EC_1\_2CC$  et ceux de  $EC_2\_2CC$  et une décomposition de l'image du caractère en 4 parties par leurs axes centraux verticale et horizontale pour les groupes  $EC_1\_1CC$ ,  $EC_1\_2CC$  et  $EC_{22\_1CC}$ .

La Figure 5-14 ci-dessous résume les étapes de la pré-classification des caractères amazighes.

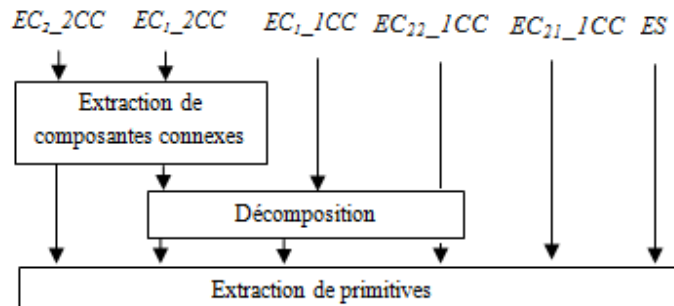


Figure 5-14 : Synopsis de processus de pré-classification

## 6. Extraction de primitives

Cette étape, consiste à représenter les données d'entrée en un vecteur de primitives de dimension fixe. Ces primitives doivent être significatives et représentatives. L'extraction de caractéristiques est une étape cruciale et critique dans les systèmes de reconnaissance. En effet, un mauvais choix de primitives influence négativement et nettement les résultats même si on utilise un classifieur très performant.

Dans la littérature, différentes approches basées sur les primitives statistiques et structurelles pour générer les séquences d'observations ont été proposées pour les écritures latines et arabes ([Oliv01], [Brit01],[Khor03], [El-Ha07], [Mena08]).

Dans notre système, nous avons opté pour les caractéristiques structurelles qui décrivent les propriétés topologiques et géométriques de l'écriture amazighe, souvent extraites directement à partir du squelette de tracé de caractère. Elles représentent globalement :

- Nombre de trous ( Nombre d'Euler) ;
- Longueur minimale et maximale des graphèmes,
- Surfaces, les diamètres et les périmètres ;
- Segments de droite et leurs attributs (position, centre de masse, orientation, ...) ;
- Arcs, boucles, et concavités ;
- Mesures des courbures et orientations principales ;
- Mesure d'excentricité, Solidités et Etendues ;
- Points des jonctions, angularités, et terminaux ;
- Premier et deuxième moment de Hu [Hu61];

Afin de pouvoir compter certaines de ces paramètres, nous avons procédé à analyser le squelette de chaque caractère pour la localisation de ses points d'intérêts. Ces points représentent généralement les extrémités des segments. La Figure 5-15 ci-dessous présente leurs différents types selon la manière dont les segments sont connectés ou agencés.

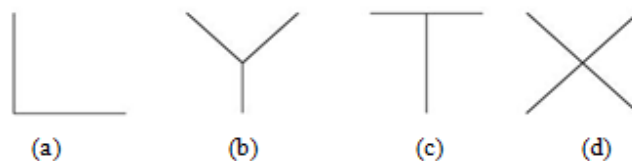


Figure 5-15 : (a) inflexion  $L$ , (b) croisement  $Y$ , (c) intersection  $T$ , (d) croisement  $X$

Notons  $A(P)$  : le nombre de transition d'un pixel noire à d'un pixel blanc ( $0 \rightarrow 1$ ) sur un son 8-voisinage  $P$ . Le Tableau 5-3 ci-dessous présente un exemple 8-voisinage du pixel  $P$  et son organisation sous forme d'une séquence  $p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8$  suivant le sens horaire. Donc, le nombre  $A(P)$  c'est le nombre de transitions ( $0 \rightarrow 1$ ) dans cette séquence.

p1	p2	p3
P8	<b>P</b>	P4
P7	P6	P5

Tableau 5-3 : Voisins de  $P$  (sens horaire de parcours):  $p1, p2, p3, p4, p5, p6, p7, p8$

Selon  $A(P)$ , il existe trois catégories de points caractéristiques [Arri08](cf. Figure 5-16 ci-dessous):

- Si  $A(P)=1$ , alors c'est un point d'extrémité.
- Si  $A(P)\geq 3$ , alors c'est un point d'intersection.
- Si  $A(P)=2$ , alors c'est un point d'inflexion.

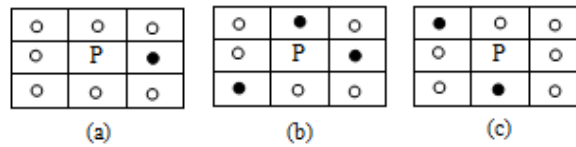


Figure 5-16: (a) points d'extrémité, (b) points d'intersection, (c) points d'inflexion

Si  $A(P)=2$ , alors on est à l'intérieur d'un segment rectiligne sans déviation ou sur un point d'inflexion. Dans ce cas, on vérifie la position des pixels de transition pour déterminer l'angle d'inflexion. Cet angle se calcule à partir de l'écart entre les positions des transitions  $0 \rightarrow 1$  (cf. Tableau 5-4 ci-dessous). Par exemple, dans la Figure 5-16-(c) ci-dessus, les voisins de  $P$  (sens horaire): 1, 0, 0, 1, 0, 0, 0, 0 2 transitions et les positions des transitions sont : 1 et 4, d'où l'angle est donnée par l'écart entre les positions est : 3 ( $3=4-1$ ).

Écart	Angle
1,7	$45^\circ$
2,6	$90^\circ$
3,5	$135^\circ$
4	$180^\circ$

Tableau 5-4 : Correspondance en angle

La Figure 5-17 ci-dessous montre le résultat de cette technique pour la détection des points d'intérêt sur les squelettes des caractères amazighes imprimés « yae » (ⵢ) et « yad » (ⵣ).



Figure 5-17: Localisation des points d'intérêts

Une fois les points d'intérêts sont déterminés sur le squelette du caractère, nous utilisons un algorithme de suivi de contour afin d'extraire les séquences d'observation de chaque caractère [Essa12]. En effet, nous avons fixé le sens de parcours de l'image de haut en bas et de gauche à droite. Le principe de l'algorithme est le suivant :

Nous partons du premier point d'intérêt  $X_i$  rencontré sur l'image pour effectuer un suivi de trait jusqu'à ce que l'on arrive sur un autre point d'intérêt  $X_j$ .

Lorsqu'on trouve un point qui vérifie la propriété  $A(P) \geq 3$ , c.-à-d., un point d'intersection de contour. Dans ce cas, l'algorithme procède à stocker ces points dans une structure pile un par un. Puis, nous empruntons l'un des branches du point immédiat dans la direction que nous avons déterminé auparavant jusqu'au dernier point  $X_i$ . En suite, nous faisons des retours sur les points empilés pour traiter éventuellement les autres branches.

Dès qu'un point d'intérêt  $X_i$  qui ne vérifie pas la propriété  $A(P) \geq 3$  du squelette a été traité, une valeur lui est assignée pour le marqué comme un point déjà visité, de manière à éviter qu'il soit à nouveau pris en considération ultérieurement. La procédure de recherche continue jusqu'à ce que toutes les extrémités du squelette soient traitées. En plus, un point d'intersection doit rester présent dans l'image jusqu'à ce que toutes ses branches aient été traitées.

la Figure 5-18 ci-dessous illustre un exemple d'application de la procédure de suivi de contour sur le caractère yav (v).

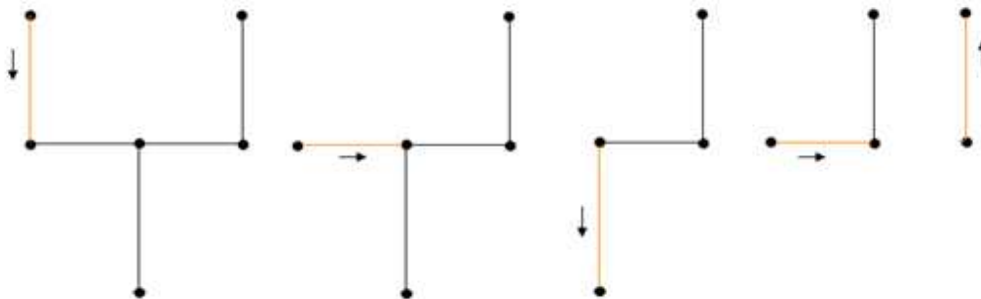


Figure 5-18 : Illustration de l'algorithme de suivi de contour sur le caractère yav (v)

Un point d'intérêt  $X_i$  est définie par ses coordonnées  $(x_i, y_i)$ . Chaque segment est représenté par une suite de pixels de squelette délimité par deux points  $(x_{i1}, y_{i1})$  et  $(x_{i2}, y_{i2})$ .

Au cours de suivi de contour expliqué précédemment, nous calculons un ensemble d'indices structurels sur chacun des différents segments qui composent le caractère, en se basant sur les boîtes englobantes de hauteur  $h=|y_{i2}-y_{i1}|$  et de largeur  $l=|x_{i2}-x_{i1}|$ . Cependant, dans le cas des segments verticaux où la hauteur est nulle ( $h=0$ ) et les segments horizontaux où la largeur est nulle ( $l=0$ ), nous avons considéré des fenêtres avec des dimensions  $h=10$  et  $l=10$  respectivement. En effet, les segments sont entourés par des boîtes adaptées et ajustées à leurs dimensions (cf. Figure 5-19 ci-dessous).

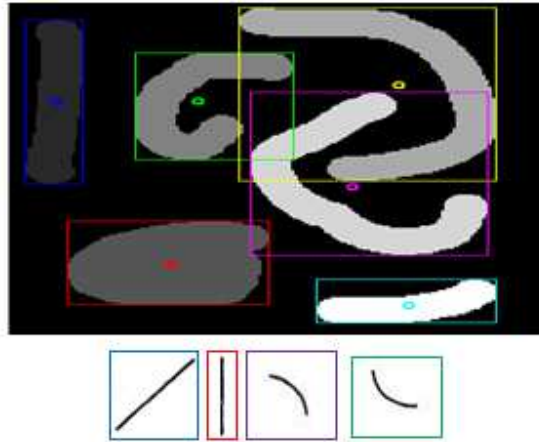


Figure 5-19 : Exemples de boîtes englobantes

Après que les segments ont été localisés implicitement sur le caractère étudié par des boîtes englobantes, chacune d'entre elles est décrite par 10 primitives qui sont illustrées par le Tableau 5-5 ci-dessous :



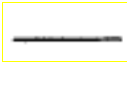

Segments primitives				
Excentricité	0.8346	0.6217	0.9347	0.3118
Centre de masse	0.1505	0.1914	0.3604	0.2052
	0.2701	0.1495	0.1300	0.2149
Orientation	0.8999	0.0118	-1.4397e-004	0.7914
Etendue	0.9740	0.9720	0.9696	0.9768
Diamètre	0.4366	0.3644	0.4681	0.4571
Longueur d'axe principal	0.6155	0.4297	0.8250	0.4866
Longueur d'axe minimal	0.3391	0.3366	0.2932	0.4623
Moment d'ordre 1	0.2355	0.1971	0.3231	0.1844
Moment d'ordre 2	0.0148	0.0023	0.0609	0.0002

Tableau 5-5: Exemples de primitives calculées

En fin, pour chaque caractère, nous obtenons un vecteur de primitives qui regroupe les séquences générées à partir de chacun des segments qui le composent.

## 7. Modèles HMMs de segments

L'approche proposée est analytique. Elle est basée sur la modélisation des segments de la liste présentée préalablement par des modèles de Markov cachés (HMMs) ([Rabi89],

[Rabi93]. Ce qui donne en totalité 10 modèles. Le modèle  $\lambda = \{N=3, M, \Pi, A, B\}$  d'un segment est de type gauche droite (cf. Figure 5-20 ci-dessous) et a les paramètres suivants :

$N$  : Le nombre d'états cachés ;

$T$  : Le nombre d'observations ou symboles possibles ;

$q_t$  : L'état du système au temps  $t$  ;

$M$  : Taille de la séquence observée ;

$$A = \{a_{ij} = p(s_j / s_i)\}; \sum_{j=1}^N a_{ij} = 1 \quad (5.2)$$

Représente les probabilités de transitions entre états ;

$$\Pi = \{\pi_i = p(s_i)\}; \sum_{i=1}^N \pi_i = 1 \quad (5.3)$$

C'est le vecteur des probabilités initiales ;

$$B = \{b_i(o_k) = p(o_k / s_i)\}; \sum_{k=1}^T b_i(o_k) = 1 \quad (5.4)$$

Représente les probabilités d'observations.

$b_j(o_k)$  sont pratiquement estimées par un mélange de  $M$  lois gaussiennes multi variables :

$$b_j(o_t) = \sum_{k=1}^M C_{jk} \mathbb{N}(o_t, \mu_{jk}, \sigma_{jk}) \quad 1 \leq j \leq M \quad \text{avec} \quad \sum_{k=1}^M C_{jk} = 1 \quad (5.5)$$

$$\text{Avec} \quad \mathbb{N}(o_t, \mu_{jk}, \sigma_{jk}) = \frac{1}{2\pi \sqrt{|\sigma_{jk}|}} \exp\left(-\frac{1}{2}(o_t - \mu_{jk}) \sigma_{jk}^{-1} (o_t - \mu_{jk})^T\right) \quad (5.6)$$

dénote une distribution gaussienne normale.

- $C_{jk}$  : le coefficient de pondération de la de  $k^{\text{ème}}$  composante de mélange.
- $\mu_{jk}$  : vecteur moyen associé.
- $\sigma_{jk}$  : matrice de covariance associée.

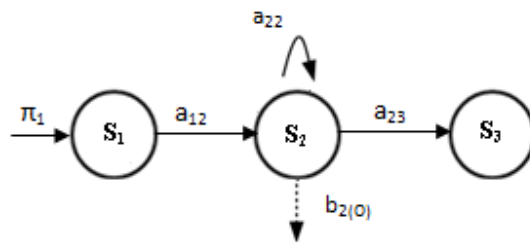


Figure 5-20 : Le modèle HMM gauche droite des segments

En pratique, nous avons utilisé la boîte à outils Hidden Markov Model Toolkit (HTK) de l'Université de Cambridge [Youn06], pour créer le modèle initial de chaque segment. Nous reviendrons sur cette plate forme en détail dans la section des expérimentations.



La Figure 5-21 ci-dessous montre le prototype HMM initial représentant le segment S1 de la liste précédente. Ce prototype est édité manuellement en se basant sur le formalisme adopté par la plate forme HTK. Le modèle HMM initial est de type gauche droite et mono-gaussien à 3 états.

```

~h "S1"
<BeginHMM>
<NumStates> 3
<State> 2 <NumMixes> 1
<Stream> 1
<Mixture> 1 1.0000
  <Mean>6
  | 0.0 0.0 0.0 0.0 0.0 0.0
  <Variance>6
  | 1.0 1.0 1.0 1.0 1.0 1.0
<TransP> 3
  | 0.000e+0 1.000e+0 0.000e+0
  | 0.000e+0 6.000e-1 4.000e-1
  | 0.000e+0 0.000e+0 0.000e+0
<EndHMM>

```

Figure 5-21 : le modèle HMM initial correspondant au segment S1

## 8. Apprentissage

Dans cette section, on cherche à modéliser au mieux les données d'apprentissage par un seul modèle global HMMs. En effet, à l'aide de module d'extraction de primitives, nous avons converti les images des segments en espace des observations. En suite, les prototypes des segments  $\lambda_s$  créés lors de la phase précédente ont été entraînés par la procédure de Baum-welsh [Rabi89], en utilisant les séquences d'observations correspondantes à la base d'entraînement. Chaque prototype de segment utilise uniquement les échantillons de sa classe.

De manière similaire, l'apprentissage (cf. [section9.3](#)) utilisé est celui du système de type modèle discriminant présenté dans le chapitre précédent. Celui-ci permet aussi de ré-estimer le modèle HMM de chaque segment, jusqu'à ce que la vraisemblance de générer les séquences d'observations de sa classe soit maximale. Le meilleur modèle HMM trouvé est enregistré. Par la suite, l'ensemble des modèles obtenus sont concaténés pour former le modèle globale ergodique de notre système. La Figure 5-22 ci-dessous illustre les différentes étapes de l'apprentissage.

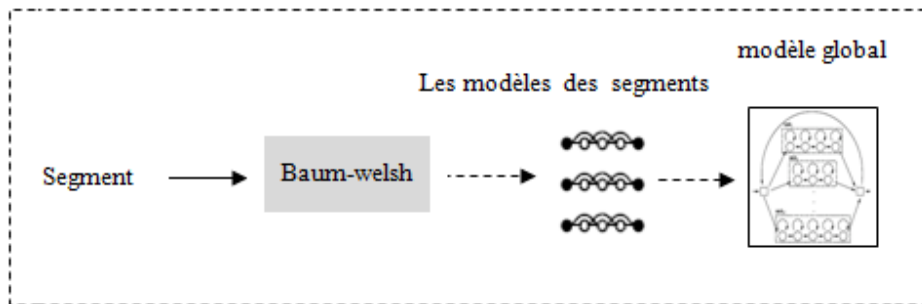


Figure 5-22 : Schéma d'apprentissage

La Figure 5-23 ci-dessous présente le meilleur modèle HMM représentant le segment S1. Ce modèle est obtenu à l'aide des outils HInit et HVite adoptées par HTK. Les paramètres : les probabilités d'entrées, de transitions, d'émissions ont été réajustés.

```

~h "S1"
<BEGINHMM>
<NUMSTATES> 3
<STATE> 2
<MEAN> 6
  2.974781e-001 2.665662e-001 2.620118e-001 1.978344e-001 1.664846e-001 2.620613e-001
<VARIANCE> 6
  7.467990e-002 9.069875e-002 8.275853e-002 5.000000e-002 5.000000e-002 5.633270e-002
<GCONST> -5.327267e+000
<TRANSP> 3
  0.000000e+000 1.000000e+000 0.000000e+000
  0.000000e+000 8.888889e-001 1.111111e-001
  0.000000e+000 0.000000e+000 0.000000e+000
<ENDHMM>

```

Figure 5-23 : Meilleur modèle HMM enregistré pour S1

## 9. Classification

A l'issue de l'étape d'apprentissage, nous avons obtenu un modèle globale ergodique modélisant l'ensemble des graphèmes de dictionnaire proposé. Chaque chemin dans ce modèle représente une séquence de segments. Par conséquent, la reconnaissance d'un caractère se fait par la recherche de meilleur chemin dans ce treillis des segments. Nous avons employé l'algorithme Viterbi, qui est fondé sur la programmation dynamique, pour décoder la meilleure séquence d'états candidates en se basant sur un critère de maximum de vraisemblance (cf. Figure 5-24 ci-dessous) [Beno07]. Pratiquement, il prend le caractère à reconnaître comme étant une séquence d'observations  $O = \{o_1, o_2, \dots, o_n\}$  extraite de son image et détermine la suite des états  $Q = \{q_1, q_2, \dots, q_n\}$  qui a la probabilité maximale d'avoir générée  $O$ .

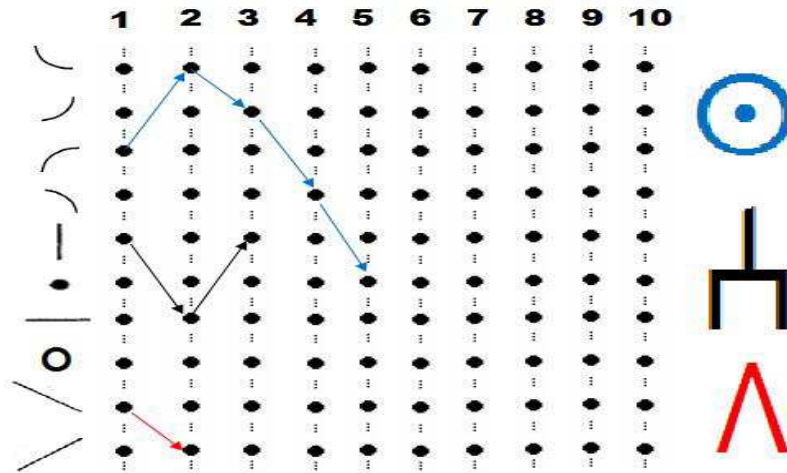


Figure 5-24 : Exemples des meilleurs chemins par le réseau de Viterbi pour les caractères

## 10. Expériences et résultats

Nous avons présenté ci-dessus la démarche déployée dans notre approche de reconnaissance de caractères amazighe imprimés. L'approche développée adopte une modélisation de type chemin discriminant qui exploite les spécificités intrinsèques de la graphie amazighe.

Pour tester la robustesse de notre système de reconnaissance, des expériences ont été effectuées sur la base des patterns de la graphie amazighe [Aito09]. Toutes ces expériences sont effectuées à l'aide de la boîte à outils des modèles de Markov cachés (HTK : Hidden Markov Models Toolkit) [Youn01]. Nous présentons dans la suite la base de données utilisée, la plate forme logicielle HTK et les expérimentations effectuées.

### 10.1. Base de données utilisée

Pour tester notre approche, nous devons disposer des images représentant les caractères à étudier. Donc, nous avons utilisé la base de caractères amazighes imprimés. Il s'agit d'une base des patterns de différentes fontes amazighes et de tailles variées [Aito09]. La base contient vingt mille patterns. Elle contient au total 12 polices de caractères et les tailles du 10 points au 28 points pour chaque modèle. Les patterns sont fournis sous forme d'images bitonales de tailles variées. Le Tableau 5-6 ci-dessous donne une liste de quelques patterns dans cette base.



Tableau 5-6 : Exemples de quelques caractères dans la base des patterns de la graphie amazighe

## 10.2. La plate forme logicielle HTK

Les Modèles de Markov cachés s'appuient sur un ensemble d'algorithmes standards et éprouvés. L'implémentation de ces algorithmes s'appuie sur des techniques bien maîtrisées de la programmation dynamique.

Actuellement, de nombreuses implémentations de ces algorithmes sont disponibles au téléchargement. En plus, un certain nombre de bibliothèques HTK, GHMM et ESMERALDA et de modules sont également publiquement accessibles pour l'apprentissage et le décodage des HMMs [Youn93].

Nous avons choisi de mener cette étude en utilisant un logiciel de développement de systèmes à base de modèles de Markov cachés le plus complet possible au sens des tâches à réaliser et dont le programme source est ouvert pour d'éventuelles fonctions à mettre en œuvre. Après une analyse des caractéristiques de chacun de ces logiciels, notre choix s'est

finalement porté sur la plate-forme logicielle HTK (Hidden Markov ToolKit). Elle était originalement mise au point à l'Université de Cambridge, dédiée au développement de systèmes à base de HMMs, principalement pour la reconnaissance de parole [Youn02]. Mais, elle à été également utilisée dans un certain nombre de publications de recherche pour la tâche reconnaissance de caractères [Khor07], de mots [Slim09]. Le Tableau 5-7 ci-dessous présente quelques caractéristiques de la librairie HTK.

	<b>HTK</b>
Organisme	Microsoft et Cambridge University
URL	<a href="http://htk.eng.cam.ac.uk/">http://htk.eng.cam.ac.uk/</a>
Langage	C
Environnement	Unix, Linux, Windows
Support	Excellent
Date de la première version	1993
Disponibilité de la source	Sous licence

*Tableau 5-7 : Quelques caractéristiques de la librairie HTK*

La boîte à outils HTK a connu une période de commercialisation. De ce fait, HTK est passée par les différents cycles de perfectionnement nécessaires au logiciel commercial. Elle est par conséquent efficace, flexible (liberté du choix des options et possibilité d'ajout d'autres modules), plus convivial et plus souple que les autres logiciels. En outre, HTK est complète dans le sens où elle fournit une documentation très détaillée (le livre HTK [Youn06] est une encyclopédie dans le domaine).

HTK se compose d'un ensemble de modules sous forme des codes sources c. L'exécution de ces modules se fait par des lignes de commande comportant des options et des arguments.

#### **Les paramètres HTK utilisés:**

Nous utilisons la boîte à outils HTK [Youn01] pour l'évaluation de notre approche. HTK permet de spécifier l'architecture des HMMs à utiliser, le nombre d'états, le nombre de gaussiennes et le dictionnaire de la reconnaissance, etc. L'utilisation de HTK pour une tâche de reconnaissance de caractères isolés, comprend trois phases : la préparation des données, l'apprentissage et la reconnaissance.

- La phase de préparation des données est effectuée par l'outil **HInit** qui vise à initialiser les modèles par un algorithme de type k-moyennes (Viterbi).
- Durant la phase d'apprentissage, l'outil **HRest** effectue la ré-estimation Baum-welsh des paramètres des modèles.
- La reconnaissance se fait en appliquant l'algorithme de Viterbi qui cherche la meilleure séquence des sous-modèles dans le HMM global par l'outil **HVite**.

- Les performances du système sont évaluées et analysées en termes de taux de reconnaissance par l'outil **HResult**.

### 10.3. Tests et résultats

Pour valider le système proposé, nous avons effectué des expérimentations significatives à l'aide de Toolkit (HTK) sur la base de données de patterns de la graphie amazighe [Aito09]. Nous avons constitué à partir de cette base deux ensembles distincts de données, un ensemble  $A$  ( $A=2/3$ ) pour l'apprentissage et un ensemble  $B$  ( $B=1/3$ ) pour les tests.

Plusieurs tests ont été effectués, pour évaluer le taux de reconnaissance du système en fonction de : nombre d'états et de nombre de mélange de gaussienne. Par ailleurs, nous avons effectué les premiers tests sur toute la base de patterns de la graphie amazighe (BD1 : contient 19437 caractères multi fonts c.-à-d. 627 échantillons x 31 classes). Le Tableau 5-8 ci-dessous présente les résultats obtenus de ces tests sur la base BD1, en utilisant les modèles mono-gaussien, les modèles à deux gaussiens et les modèles à trois gaussiens.

<b>Nombre d'états</b>	3	5
<b>Nombre de gaussiennes</b>	1-2-3	1-2-3
<b>Taux de reconnaissance</b>	98.21%	98.52%

Tableau 5-8 : Taux de reconnaissance sur BD1

La Figure 5-25 ci-dessous illustre la sortie renvoyée par la plate forme logicielle HTK sur la base BD1

```

:===== HTK Results Analysis =====
Date: Thu Mar 01 18:34:50 2012
Ref : donnees/Labels
Rec : resultats
----- Overall Results -----
SENT: %Correct=98.21 [H=1204, S=22, N=1226]
WORD: %Corr=98.21, Acc=98.21 [H=1204, D=0, S=22, I=0, N=1226]
:=====
----- Overall Results -----
SENT: %Correct=98.52 [H=5714, S=86, N=5800]
WORD: %Corr=98.52, Acc=98.52 [H=5714, D=0, S=86, I=0, N=5800]
:=====

```

Figure 5-25: Résultats retournés par HTK sur BD1

Ces résultats montrent un taux d'erreur de 1,48% avec un modèle de topologie de 5 états. Ce fait peut se justifier par la déformation de certains caractères dans certaines fontes, notamment dans les fontes « Tassafut » et « Taromeit ». La Figure 5-26 et la Figure 5-27 ci-dessous illustrent quelques exemples de caractères de la fonte « Tassafut » et dans la fonte « Taromeit » respectivement.



Figure 5-26 : Exemples de caractères de la base, de la fonte « Tassafut »



Figure 5-27 : Exemples de caractères de la base, de la fonte « Taromeit »

En effet, les traits constituant certains caractères de ces fontes ne respectent pas les éléments de la liste précédente que nous avons définie dans la [section 2](#). Ce choix de description impose que les caractères doivent être composés exclusivement avec ces primitives, donc une certaine contrainte sur le style d'écriture admissible par notre système.

En revanche, cette contrainte est relativement réaliste et peu contraignante, puisque le dictionnaire a été établi de façon à couvrir l'essentiel des formes de traits rencontrés l'alphabet tifinagh.

Afin de valider notre hypothèse nous avons procédé à un deuxième test sur un sous ensemble BD2 de la base BD1 précédente. En effet, La base BD2 est constituée à partir de BD1. Elle comporte 456 x 31 classes, c.-à-d., 14136 caractères. Nous avons exclu les deux fontes Tromeit et Tassafut précédentes. La Figure 5-28 ci-dessous présente quelques exemples de caractères de la base BD2. On constate que les traits constituant les tracés de ces caractères respecte les segments de la liste précédente.



Figure 5-28 : Exemples de caractères de la base, dont la fonte est dérivée de deux fontes Tromeit et Tassafut

Nous avons obtenus des résultats très encourageants présentés par le Tableau 5-9 ci-dessous. Le meilleur score enregistré est de 99,76%, avec les modèles de topologie de 5 états. Nous constatons que le nombre de gaussienne utilisé n'influence pas les résultats, tandis que son augmentation implique un nombre important de paramètres à calculer. Cependant, le choix de la topologie influence directement les résultats. En effet, l'augmentation de nombre d'états augmente le taux de reconnaissance de système.

Nombre d'états	3	5
Nombre de mélange de gaussienne	1-2-3	1-2-3
Taux de reconnaissance	99,38%	99,76%

Tableau 5-9 : Taux de reconnaissance sur BD2



La Figure 5-29 ci-dessous montre le résultat renvoyé par la boîte à outils HTK sur BD2

```

:===== HTK Results Analysis =====:
Date: Thu Mar 01 16:34:50 2012
Ref : donnees/Labels
Rec : resultats
----- Overall Results -----
SENT: %Correct=99.38 [H=5764, S=36, N=5800]
WORD: %Corr=99.38, Acc=99.38 [H=5764, D=0, S=36, I=0, N=5800]
:=====
----- Overall Results -----
SENT: %Correct=99.72 [H=5784, S=16, N=5800]
WORD: %Corr=99.72, Acc=99.72 [H=5784, D=0, S=16, I=0, N=5800]
:=====

```

*Figure 5-29 : Résultats retournés par HTK sur BD2*

Les résultats obtenus sur la BD2 sont plus prometteurs que ceux obtenus sur la base BD1. Cela confirme notre hypothèse. En effet, le gain le plus important en taux de reconnaissance est de l'ordre de 1,24% en Tableau 5-9 ci-dessus avec une topologie de 5 états par modèle. Ces tests ont montré que les HMMs continus sont plus efficaces. Ces résultats ont été publiés dans le journal international IRECOS [Amro12b].

Enfin, nous estimons que les autres erreurs de reconnaissance sont attribuées, d'une part, aux méthodes utilisées pour la pré-classification et à la détection des points d'intérêts, et d'autre part, à l'insuffisance des caractéristiques utilisées pour mieux décrire chaque segment. En effet, nous avons déployé un algorithme sélectif, qui s'appuie sur un indice de classification pendant la pré-classification. En outre, nous avons employé la technique qui repose sur le nombre de transition noire et blanche entre les pixels d'un voisinage 8, pour localiser les points d'intérêts. Cette méthode dépend fortement de la topologie de la forme traitée. Si le squelette obtenu de la forme possède des déformations, il influence la détection des ces points.

En vue de remédier à ces problèmes pour diminuer le taux d'erreur de notre système et par conséquent avoir un système fiable, nous proposons dans ce qui suit, une autre variante de ce système, qui consiste à adopter la transformation de Hough pour séparer les lettres circulaires et les non circulaires. En plus, cette version s'appuie sur une autre technique pour déterminer les points d'intérêts de la forme, la technique utilise la notion de déviation maximale au niveau de la courbe du caractère.

## **11. Version améliorée du système proposé**

Nous avons décrit ci-dessus une méthode de reconnaissance de l'écriture amazighe qui adopte une modélisation markovienne de type chemin discriminant, basée sur une phase de pré classification des caractères amazighes et combinée avec une technique de localisation des points d'intérêt sur le squelette. Les résultats obtenus ont été jugés intéressants. Cependant, nous estimons que les erreurs sont attribuées aux démarches adoptées dans la pré classification et l'extraction de primitives. Nous proposons dans cette partie, une version améliorée de notre système de reconnaissance.

Cette variante comprend les mêmes étapes que la précédente. La différence réside donc au niveau des techniques employées dans les phases : pré-classification et extraction de primitives. En effet, contrairement à l'approche présentée précédemment, nous avons déployé la transformée de hough pendant la pré-classification et une nouvelle technique à base de déviation maximale pour trouver les points d'intérêts de la forme, et donc localiser les segments composant le caractère. La méthode présentée dans ce qui suit consiste à améliorer le taux de reconnaissance obtenu par rapport au système de reconnaissance décrit précédemment en lui perfectionnant les étapes pré classification et génération d'observations. La Figure 5-30 ci-dessous montre le diagramme de la version améliorée de notre système.

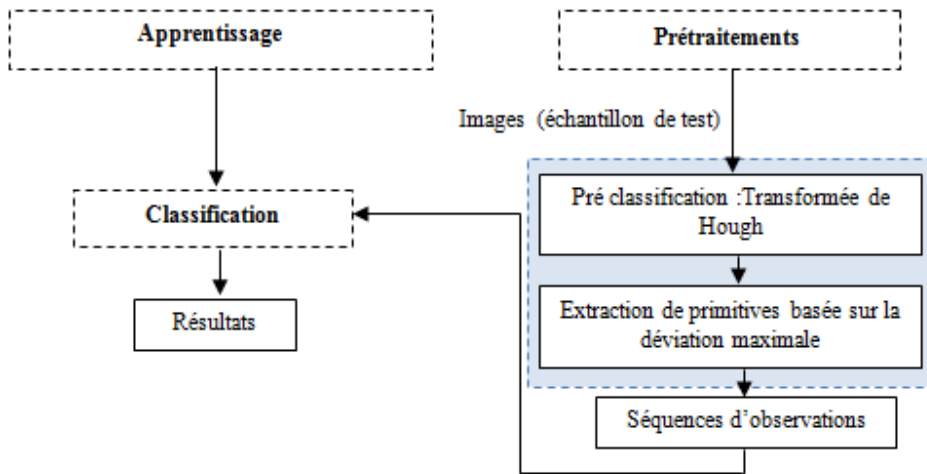


Figure 5-30 : Architecture du système amélioré

### 11.1. Pré classification des caractères amazighes : version améliorée

L'idée de base est d'exploiter l'efficacité et la robustesse de la transformation de Hough dans la détection des formes paramétriques (droite, cercle, ellipse...) [Mait85]. En effet, le problème qu'on cherche à résoudre est le suivant :

pré classer les images de caractères amazighes en 2 groupes : formes circulaires (ⵝ, ⵏ, ⵝ, ⵏ, ⵝ, ⵏ, ⵝ, ⵏ) et formes non circulaires (ⵉ, ⵏ, ⵉ, ⵏ, ⵉ, ⵏ). D'où l'intérêt d'utiliser cette transformée. Le principe de Hough repose principalement de la projection des pixels de l'image dans un espace paramétrique, appelé espace de Hough. Algorithmiquement, il s'agit simplement de parcourir les pixels de l'image ainsi que les éléments de l'accumulateur, et d'effectuer ensuite un calcul simple à partir des équations de cercles cherchés ( $R1=Hauteur/2$  et  $R2=Hauteur/4$ ). Ainsi, les valeurs contenues dans l'accumulateur constitueront les premiers indices de présence des cercles dans l'image. Cependant, ces valeurs ne suffisent pas à désigner les cercles qui nous intéressent pour la suite des traitements. Mais, un simple seuillage permet ensuite de ne conserver que les vecteurs de paramètres dont les valeurs dans l'accumulateur sont élevées, Ces valeurs candidats sont en réalité des cercles cherchés. La Figure 5-31 ci-dessous présente des exemples des caractères circulaires

détectés. Dans le cas de  $R1=Hauteur/2=20.8$  c.-à-d., les caractères ( $\mathcal{Q}$ ,  $\mathcal{O}$ ,  $\mathcal{E}$ ,  $\mathcal{O}$ ,  $\mathcal{O}$ ,  $\mathcal{O}$ ), le seuil fixé doit être proche de la valeur maximale 47. Par contre, dans le cas de  $R2=Hauteur/4=7.03$ , c.-à-d., les caractères ( $\mathcal{O}$ ,  $\mathcal{O}$ ,  $\mathcal{C}$ ,  $\mathcal{O}$ ), le seuil fixé doit être proche de la valeur maximale 27.

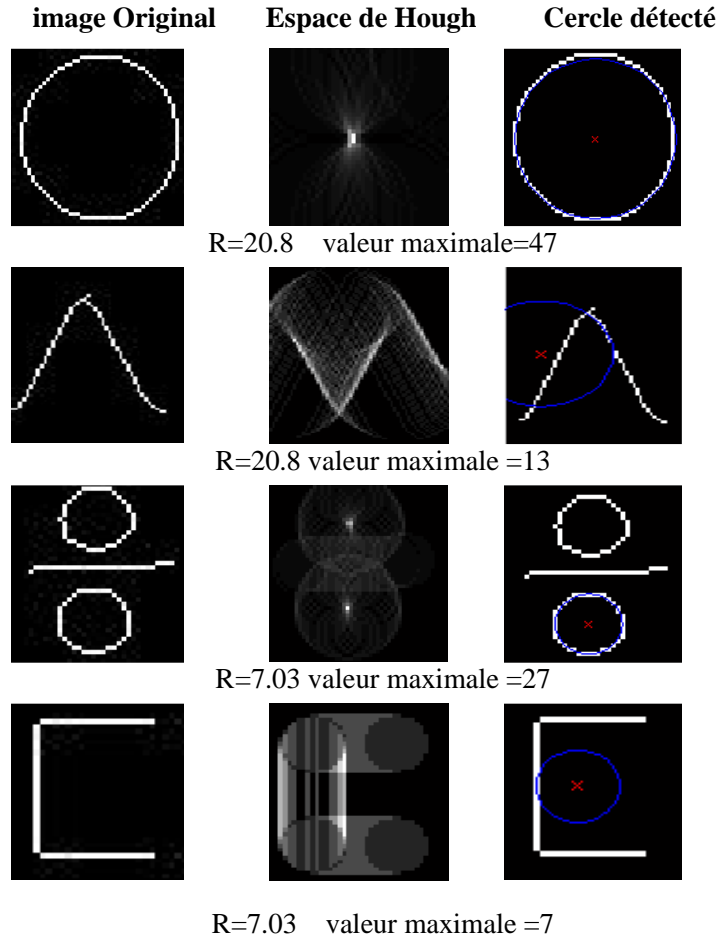


Figure 5-31 : Détection des cercles par la transformation de Hough

En conséquence, nous obtenons les mêmes groupes de caractères que la version de référence présentée auparavant, qui sont :

-EC={EC1=( $\mathcal{Q}$ ,  $\mathcal{O}$ ,  $\mathcal{E}$ ,  $\mathcal{O}$ ,  $\mathcal{O}$ ,  $\mathcal{O}$ ), EC2=( $\mathcal{O}$ ,  $\mathcal{O}$ ,  $\mathcal{C}$ ,  $\mathcal{O}$ )};

-ES={ $\mathcal{K}$ ,  $\mathcal{+}$ ,  $\mathcal{z}$ ,  $\mathcal{z}$ ,  $\mathcal{r}$ ,  $\mathcal{A}$ ,  $\mathcal{A}$ ,  $\mathcal{H}$ ,  $\mathcal{H}$ ,  $\mathcal{C}$ ,  $\mathcal{U}$ ,  $\mathcal{r}$ ,  $\mathcal{l}$ ,  $\mathcal{K}$ ,  $\mathcal{E}$ ,  $\mathcal{E}$ ,  $\mathcal{X}$ ,  $\mathcal{I}$ }.

A ce niveau, ces groupes de caractères seront soumis les mêmes processus de filtrage que précédemment, afin d'aboutir à un ensemble de classes. Par la suite, chaque classe à subi un traitement approprié à sa morphologie avant d'être soumise au module commun d'extraction.

### 11.2. Extraction de primitives

L'efficacité de la méthode adoptée pour de la détection des points d'intérêts, qui repose sur le nombre  $A(P)$  de transition d'un pixel noire à d'un pixel blanc ( $0 \rightarrow 1$ ) sur les voisins de pixel central  $P$ , dépend fortement du squelette de caractère étudié. En effet, la squelettisation cause souvent plusieurs déformations de type intersection ou barbules qui sont apparus au niveau de la forme. Si le squelette généré a subi des perturbations, cela influence négativement la localisation de ces points. Par conséquent, dans le but d'améliorer le processus d'extraction de segment, nous avons proposé une autre technique basée sur la notion de déviation maximale au lieu de nombre  $A(P)$  [Kove96], pour localiser les points d'intérêts.

La déviation maximale entre deux points  $(x_1, y_1)$  et  $(x_2, y_2)$  de contour de la forme est donnée par l'équation :

$$x(y_2 - y_1) + y(x_2 - x_1) + y_2x_1 - y_1x_2 = 0 \quad (ax + by + c = 0) \quad (5.7)$$

Nous déterminons la position et la valeur de déviation maximale en fonction d'un seuil de tolérance fixé empiriquement (seuil = [2..4]). Si la déviation maximale dépasse la tolérance admissible, le contour est réduit dans ce point et l'algorithme se poursuit. Sinon on refait la même procédure pour les pixels successeurs au long du contour de caractère étudié. De cette manière chaque contour est décomposé en segments de ligne. La Figure 5-32 ci-dessous illustre le résultat obtenu par la technique de déviation maximale sur les caractères amazighes « yae », « yam », « yad » et « yaq ».

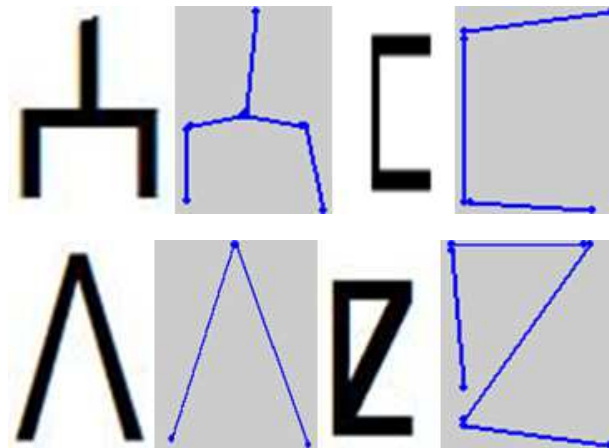


Figure 5-32 : Exemples de localisation des segments par la technique de déviation maximale

A ce niveau, nous appliquons la même procédure de suivi du contour précédente en utilisant de la même manière la technique des boîtes englobantes décrite ci-dessus (cf.

Figure 5-33 ci-dessus). Nous obtiendrons le vecteur de primitives structurales de chaque caractère étudié, en explorant tous ses segments.

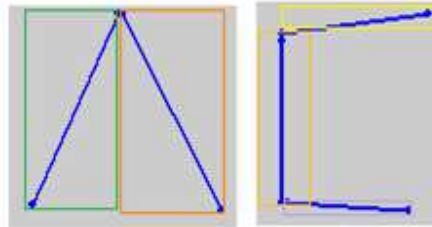


Figure 5-33 : Exemples des boîtes englobantes

### 11.3. Analyse des résultats

De la même façon, pour évaluer la version améliorée de notre système, on recommence la première expérience précédente. Nous avons effectué, à nouveau, l'expérimentation de notre système amélioré sur la totalité de la base de données de patterns de la graphie amazighe (*BDI*), en utilisant les mêmes parties déjà constituées :  $A$  ( $A=2/3$ ) pour l'apprentissage et un ensemble  $B$  ( $B=1/3$ ) pour les tests.

D'une manière similaire, on procède par une série de tests pour mesurer le taux de reconnaissance de notre système en fonction de : nombre d'états et de nombre de mélange de gaussiennes. Le Tableau 5-10 ci-dessous montre les résultats obtenus par la version améliorée sur la base *BDI*.

<b>Nombre d'états</b>	3	5
<b>Nombre de gaussienne</b>	1-2-3	1-2-3
<b>Taux de reconnaissance</b>	98,41%	98,76%

Tableau 5-10: Taux de reconnaissance sur *BDI*

Dans ce cas, on constate que le taux est de 98,41% par une topologie de type gauche droite à 3 états et le même pour tous les modèles gaussiens utilisés (1,2 et 3). Soit un gain de 0,20% sur le score avec la nouvelle version. En outre ce taux s'élève à 98,76 par la topologie de type gauche droite à 5 états et le même pour tous les modèles gaussiens utilisés (1,2 et 3). C.-à-d. cette fois-ci un gain de 0,24. En comparant ces taux de reconnaissance obtenus par la version améliorée de notre système avec ceux obtenus par la première version, nous soulignons une amélioration qui s'échelonne entre 0,2% et 0,25%. Les résultats de cette version ont été publiés dans le journal international *IJCA* [Amro12c].

De ce fait, les techniques déployées lors des phases pré-classification et extraction de primitives, notamment celle qui concerne la localisation des segments de caractères, ont une influence significative sur la performance de système. Par ailleurs, cela démontre que

l'emploi de la technique de déviation maximale au lieu de la méthode classique pour détecter les points d'intérêts permet une augmentation au niveau des scores obtenus.

## **12. Conclusion et perspectives**

Dans ce chapitre, nous avons proposé une solution au problème de la reconnaissance automatique de caractères Tifinaghs, fondée sur une modélisation markovienne continue de type chemin discriminant. La solution apportée exploite les caractéristiques morphologiques du caractère tifnagh, en proposant une liste limitée de segments. Cette modélisation présente l'avantage d'être indépendante du nombre de classes de reconnaissance (en termes de mémoire et de vitesse) puisque le modèle est construit pour toutes les classes. Deux variantes de ce système ont été proposées. Chacune d'elles a été évaluée plusieurs fois avec différents paramètres (nombre d'états, nombre de gaussiens....) sur la base des patterns de la graphie amazighe. Les résultats obtenus globalement par les deux versions sont tout à fait encourageants. En outre, ceux obtenus sur les modèles mono gaussiens, deux gaussiens et trois gaussiens se ressemblent. Par contre, on constate une évolution de taux de reconnaissance avec l'augmentation de nombre d'états.

L'analyse des résultats obtenus par la version améliorée du système montre une amélioration significative du taux de reconnaissance par rapport à la première alternative. Ce qui montre que l'amélioration proposée apporte une augmentation des performances de système. Ces résultats montrent que les HMMs continus sont plus robustes. Cependant les inconvénients de cette approche résident dans : (1) la détection des points d'intérêts pendant l'extraction des caractéristiques semble contraignante pour certains styles d'écriture utilisés ; (2) le choix des éléments de vocabulaire, qu'ont été établi de façon à couvrir l'essentiel des formes de segments rencontrés dans l'alphabet tifinagh, impose une contrainte sur les styles admissibles par le système. En effet, un caractère dont un de ses segments ne correspondrait à aucun élément de ce vocabulaire ne pourra pas être reconnu par le système. Pour remédier à ces problèmes, l'intégration d'autres caractéristiques, augmentation de taille de vocabulaire et la proposition d'une autre méthode de segmentation des caractères en graphèmes peuvent constituer une perspective que nous envisageons pour l'amélioration de notre approche. En plus, l'utilisation des HMMs continus (CHMMs) en combinaison avec d'autres classifieurs notamment les réseaux de neurones de type perceptron multicouches (Multilayer Perceptron : MLP ) peut diminuer le taux d'erreur de notre système et par conséquent avoir un système fiable.

---

## Conclusion générale et perspectives

---

Le travail présenté dans cette thèse s'inscrit dans le cadre de la reconnaissance automatique hors ligne de l'écriture manuscrite et imprimée. Dans ce contexte, nous avons tracé un état de l'art sur les généralités et les préliminaires associés à la chaîne de reconnaissance d'une part, et une étude bibliographique sur les différentes approches markoviennes de reconnaissance automatique de l'écriture proposées dans la littérature d'autre part. En outre, nous avons développé deux systèmes de reconnaissance de caractères manuscrits et imprimés.

La première contribution porte sur un système pour la reconnaissance de caractères arabes et amazighes manuscrits isolés, en utilisant une approche markovienne de type modèle discriminant basée sur les primitives directionnelles. Nous avons utilisé des modèles de Markov cachés de type gauche droite du premier ordre dans le cas discret et le cas continu. Après les prétraitements, les images des caractères étudiées sont transformées en séquences d'observations à l'aide des techniques employées pendant la phase d'extraction de caractéristiques. Nous avons déployé les algorithmes Forward et Baum-welsh durant la classification et l'apprentissage. Nous avons effectué nos expériences sur des ensembles distincts : une base locale des caractères arabes manuscrits isolés, un sous ensemble de la base AMHCD et sur la totalité de AMHCD. Les résultats enregistrés montrent l'efficacité de la modélisation proposée, notamment dans le cas continu, où les densités de probabilités sont modélisées par les gaussiennes. Comme limite de cette approche, elle ne tient pas en compte de la morphologie de l'écriture traitée. Dans le but de proposer un système de reconnaissance robuste et spécifique à une écriture donnée, nous avons décidé et fait le choix d'étudier succinctement les spécificités de la langue amazighe.

En effet, dans la deuxième contribution, nous avons proposé une solution au problème de la reconnaissance automatique de caractères amazighes, fondée sur une modélisation markovienne continue de type chemin discriminant. La solution apportée exploite les caractéristiques morphologiques et intrinsèques des caractères tifnaghes, en proposant une liste limitée de segments. Chaque caractère s'écrit en combinaison des éléments de cette liste. Les caractéristiques exploitées sont extraites à partir des tracés des caractères par une technique de localisation implicite des segments qui le composent. Pour se faire, nous avons utilisé les points d'intérêts des squelettes. Dans la phase de classification nous avons déployé le chemin discriminant fondé sur la programmation dynamique opérant au niveau de graphe des segments. Deux variantes de ce système ont été proposées. Chacune d'elles a été évaluée plusieurs fois avec différentes paramètres (nombre d'états, nombre de gaussiens....) sur la base des patterns de la graphie amazighe. Les résultats obtenus globalement par les deux versions sont tout à fait encourageants. En outre, ceux obtenus sur les modèles mono gaussiens et deux gaussiens et trois gaussiens se ressemblent. Par contre,

on constate une évolution de taux de reconnaissance avec l'augmentation de nombre d'états.

L'analyse des résultats obtenus par la version améliorée du système montre une amélioration significative du taux de reconnaissance par rapport à la première alternative. Ce qui montre que l'amélioration proposée apporte une augmentation des performances du système. Ils montrent que l'utilisation de la technique de localisation des points d'intérêts par la déviation maximale durant l'extraction des primitives, en combinaison avec l'algorithme à base de la transformée de hough pendant la pré-classification des caractères amazighes, améliore légèrement les résultats obtenus.

Contrairement à la première contribution qui consiste à associer un ou plusieurs modèles par classe, elle est pratiquement utilisée dans le cas où le nombre de classes à reconnaître est relativement limité ( application à vocabulaire limité). Toutefois elle devient coûteuse en temps de calcul et place mémoire quand ce nombre dépasse le millier puisque chaque classe possède au moins un modèle qui lui est propre. La modélisation proposée dans la deuxième contribution présente l'avantage d'être indépendante du nombre de classes de reconnaissance (en terme de mémoire et de vitesse) puisque le modèle est construit pour toutes les classes.

Malgré que les résultats obtenus par les deux systèmes aient été jugés encourageants, plusieurs améliorations et perspectives sont envisageables pour poursuite de nos travaux, en particulier :

Dans le cas du premier système qui adopte la modélisation markovienne de type modèle discriminant, parmi les perspectives qui pourraient être envisagées : L'intégration d'autres caractéristiques pour enrichir la description des formes pendant l'étape d'extraction. Ainsi, l'augmentation de la taille des données considérées pendant la phase d'apprentissage. Ce facteur peut contribuer directement à une amélioration de taux de reconnaissance puisque on étale les chances de représenter les différentes et les variabilités inter et intra formes des caractères. En outre, dans cette approche et au cours de la génération des séquences d'observations, nous avons adopté une démarche indépendamment de la graphie utilisée. Mais, on peut proposer des techniques exploitant les caractéristiques morphologiques intrinsèques d'une écriture donnée.

Quant au deuxième système, nous pouvons estimer que les faiblesses proviennent essentiellement, de la phase de la détection des points d'intérêts pendant l'extraction des caractéristiques qui semblent contraignantes pour certains styles d'écriture utilisés d'une part. En plus, le choix des éléments de vocabulaire qu'ont été établi de façon à couvrir l'essentiel des formes de segments rencontrés dans l'alphabet tifinagh, impose une contrainte sur les styles admissibles par le système d'autre part. En effet, un caractère dont un de ses segments ne correspondrait à aucun élément de ce vocabulaire ne pourra pas être reconnu par le système. Pour remédier à ces problèmes, l'augmentation de taille de liste et la proposition d'une autre méthode de segmentation des caractères en graphèmes qui serait



appliquée dans le cas où un trait d'un autre style d'écriture ne correspondrait à aucune des segments répertoriées, peuvent constituer une perspective que nous envisageons pour l'amélioration des performances de notre système.

Pour les deux systèmes, nous avons utilisé des algorithmes classiques, en particulier Forward et Viterbi pour la reconnaissance et Baum-welsh pour l'apprentissage. En perspectives, l'application des alternatives plus évoluées de ces procédures, notamment les variantes de Viterbi pour la reconnaissance et l'apprentissage, peuvent améliorer les performances et accélérer le temps de calcul. Par ailleurs, l'utilisation des modèles de Markov cachés en combinaison avec d'autres classifieurs notamment les réseaux de neurones peut diminuer le taux d'erreur de notre système et par conséquent avoir un système fiable.

## Bibliographie

- [Adam06] T. Adamek, E. O'Connor, F. Smeaton, Word matching using single closed contours for indexing handwritten historical documents IJDAR, Springer-Verlag, 2006.
- [Agaz93] O. Agazi & S. Kuo. Hidden Markov model based optical character recognition in the presence of deterministic transformations. Pattern Recognition, Vol. 26, N° 12, pp. 1813–1826, 1993.
- [Aito09] Y. Ait Ouguengay, M. Taalabi, Elaboration d'un réseau de neurones artificiels pour la reconnaissance optique de la graphie amazighe: Phase d'apprentissage, Systèmes intelligents-Théories et applications, Paris : Europa, cop. (impr. au Maroc), ISBN-102909285553, Avril 2009.
- [Al-Em90] S. Al-Emami and M. Usher. On-line recognition of handwritten arabic characters. IEEE Trans. Pattern Anal. Mach. Intell., Vol. 12, N° 7, pp. 704–710, 1990.
- [Alim97] Adel M. Alimi. An evolutionary neuro-fuzzy approach to recognize on-line Arabic handwriting. In ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition, IEEE Computer Society, pp. 382–386, 1997.
- [Alle03] F. Alleau, E. Poisson, C.V. Gaudin, and P. Le Callet, Tdnn with masked inputs. Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on, 2 pp. 989–993, vol. 2, 15-18 December 2003.
- [Al-Ma02a] Al-Ma'adeed S., Higgins C., Elliman D., "A database for Arabic handwritten text recognition research", Proceedings of IWFHR'02, 8th International Workshop on Frontiers in Handwriting Recognition, pp. 485-489, Ontario, Canada, August 2002.
- [Al-Ma02b] Al-Ma'adeed S., Higgins C., Elliman D., "Recognition of off-line handwritten Arabic words using hidden Markov model approach", Proceedings of ICPR'02, 16th International Conference on Pattern Recognition, Vol. 3, pp. 481-484, Quebec City, Canada, August 2002.
- [Al-Ma04] Al-Ma'adeed S., Elliman D., Higgins C., "Off-line recognition of handwritten Arabic words using multiple hidden Markov models", Knowledge-Based Systems, Vol. 17, N° 2-4, pp. 75-79, May 2004.
- [Al-Ma06] Al-Ma'adeed S., "Recognition of off-line handwritten Arabic words using neural network", Proceeding of GMAI'06, International Conference on Geometric Modeling and Imaging, pp. 141-114, London, England, July 2006.
- [Al-Oh02] Al-Ouali Y., "Handwritten Word Recognition – Application to Arabic Cheque Processing", PhD Thesis, Concordia University, Montreal, Quebec, Canada, February 2002.

- [Al-Ra06] Al-Rashaideh H., "Preprocessing phase for Arabic word handwritten recognition", *Information Transmissions in Computer Networks*, Vol. 6, N°1, pp. 11-19, 2006. (Disponible sur la toile : [www.jip.ru/2006/11-19-2006.pdf](http://www.jip.ru/2006/11-19-2006.pdf)).
- [Alsa06] Alsallakh B., Safadi H., "AraPen: an Arabic online handwriting recognition system", *Proceeding of ICTTA'06, 2nd IEEE International Conference on Information & Communication Technologies: from Theory to Applications*, Vol. 1, pp. 1844-1849, Damascus, Syria, April 2006.
- [Amin03] Amin A., "Recognition of hand-printed characters based on structural description and inductive logic programming", *Pattern Recognition Letters*, Vol. 24, pp. 3187-3196, 2003.
- [Amin80] A. Amin, A. Kaced, J.P. Haton, and R. Mohr. Handwritten arabic character recognition by the irac system. In *ICPR*, pp.729–731, 1980.
- [Amro08] M. Amrouch, M. Elyassa, A. Rachidi, D. Mammass : "Off-line Arabic Handwritten Characters Recognition Based on a Hidden Markov Models", , *Proceedings of the 3rd ICISP, Cherbourg-Octeville, Normandy, France*, pp.447-454, Juillet 2008.
- [Amro09] M. Amrouch, Y. Es Saady, A. Rachidi, M. El Yassa, D. Mammass, "Printed Amazigh Character Recognition by a Hybrid Approach Based on Hidden Markov Models and the Hough Transform", *ICMCS'09, Ouarzazate*, 2009.
- [Amro10] M. Amrouch, A. Rachidi, M. El Yassa, D. Mammass, "Handwritten Amazigh Character Recognition Based On Hidden Markov Models", *ICGST-GVIP Journal*, vol.10, Issue 5, pp.11-18, December 2010.
- [Amro12a] M. Amrouch, Y. Es Saady, A. Rachidi, M. El Yassa, D. Mammass "Handwritten Amazigh Character Recognition System Based on Continuous HMMs and Directional Features", *IJMER journal*, Vol.2, Issue 2, pp.436-441, Mar.-Apr. 2012.
- [Amro12b] M. Amrouch, Y. Es Saady, A. Rachidi, M. El Yassa, D. Mammass "A New Approach Based on Strokes for Printed Tifinagh Character Recognition Using Discriminating Path-HMM ", *IRECOS journal*, Vol.7, N°.2, Mars 2012.
- [Amro12c] M. Amrouch, Y. Es Saady, A. Rachidi, M. El Yassa, D. Mammass "A Novel Feature Set for Recognition of Printed Amazigh Text Using Maximum Deviation and HMM", *IJCA journal*, Vol.44, N°.12, pp.23-30, April 2012.
- [Anta03] Antanacopoulos A., Gatos B., Karatzas D., Page Segmentation Competition, *ICDAR'2003*, pp.688-692, 2003.
- [Aric01] Nafiz Arica and Fatos T. Yarman-Vural. "An overview of character recognition focused on off-line handwriting", *IEEE Transactions on Systems, Man and Cybernetics - part C : Applications and Reviews*, Vol.31, N°.2, pp.216–233, May 2001.
- [Ariv07] M. Arivazhagan, H. Srinivasan, S.N. Srihari, A statistical approach to handwritten line segmentation, in: *Document Recognition and Retrieval XIV, Proceedings of SPIE, San Jose, CA, USA*, pp. 6500T-1-11, February 2007.

- [Arri08] Denis ARRIVAUL, Apport des Graphes dans la Reconnaissance Non-Contrainte de Caractères Manuscrits Anciens, Thèse de doctorat, Traitement du Signal et des Images, Université de Poitiers, 2008.
- [Assr07] S. Assrvatham « Apprentissage à base de Noyaux Sémantiques pour le Traitement de Données Textuelles » thèse de Doctorat, Paris, 2007.
- [Augu00] Augustin E., Price D., Baret O., "Le système de reconnaissance de documents A2IA Check Reader™", CIFED'00, 6ème Colloque International Francophone sur l'Écrit et le Document, pp.241-250, Lyon, 2000.
- [Augu01] E. Augustin. Reconnaissance de mots manuscrits par systèmes hybrides Réseaux de Neurones et Modèles de Markov Cachés, these de doctorat, Université RENE DESCARTES - PARIS V, 2001.
- [Aziz02a] Azizi N., Sari T., Souici-Meslati L., Sellami. M., "Une architecture de combinaison floue de classifieurs neuronaux pour la reconnaissance de mots arabes manuscrits", CIFED'02, 7ème Colloque International Francophone sur l'Écrit et le Document, pp. 89-96, Hammamet, Tunisie, Octobre 2002.
- [Aziz02b] Azizi N., "Combinaison de classifieurs neuronaux basée sur la logique floue : application à la reconnaissance des mots arabes manuscrits", Mémoire de magister, Laboratoire LRI, Département d'informatique, Université d'Annaba, Algérie, 2002.
- [Bagh05] Baghshah M. S., Shouraki S. B., Kasaei S., "A novel fuzzy approach to recognition of online Persian handwriting", Proceedings of the ISDA'05, 5th International Conference on Intelligent Systems Design and Applications, pp.268-273, Wroclaw, Poland, September 2005.
- [Bahl01] Claus Bahlmann and Hans Burkhardt, Measuring HMM similarity with the Bayes probability of error and its application to online handwriting recognition. In Proc. of the 6th ICDAR, pp.406-411, 2001.
- [Bald95] P. Baldi, Y. Chauvin, T. Hunkapiller & M. Mc. Clure. Hidden Markov models of biological primary sequence information. Proceedings of Nature Academic Science, Vol. 91, N°. 3, pp.1059-1063, 1995.
- [Ball06] Ball G., Srihari S. N., Srinivasan H, "Segmentation-free and segmentation dependent approaches to Arabic word spotting", IWFHR'06, 10th International Workshop on Frontiers in Handwriting Recognition, pp. 53-58, La Baule, France, October 2006.
- [Bela92] Belaid A., Belaid Y., Reconnaissance de formes : méthodes et applications, Interéditions, 1992.
- [Bela97] A. Belaid & G. Saon. Utilisation des processus markoviens en reconnaissance de l'écriture. Traitement du signal, vol.14, N°. 2, pp.161-177, 1997.
- [Bena00] Ben Amara N., Belaïd A., Ellouze N., "Utilisation des modèles Markoviens en reconnaissance de l'écriture arabe état de l'art", Colloque International Francophone sur l'Écrit et le Document (CIFED'00), Lyon, France, pp.181-191, 2000.

- [Bena99] 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.
- [Beng94] Bengio Y. and LeCun Y. "Word Normalization for On-Line Handwritten Word Recognition", IAPR, editor, proc. Of the International Conference on pattern Recognition, vol. 2, pp.409 – 413, Jerusalem, October 1994.
- [Beno06a] A. Benouareth, A. Ennaji, and M. Sellami. Hmms with explicit state duration applied to handwritten arabic word recognition. In ICPR06, pp. 897–900, 2006.
- [Beno06b] Benouareth A., Ennaji A., Sellami M. "Reconnaissance des Mots manuscrits Arabes par Combinaison d'une Approche Globale et une Approche Analytique", IWFHR, pp.265-270, 2006.
- [Beno07] A. Benouareth. Reconnaissance de Mots Arabes Manuscrits par Modèles de Markov Cachés à Durée d'Etat Explicite, Thèse de doctorat, Université Badji Mokhtar -Annaba V, 2007.
- [Bens04] Bensefia A., "Analyse des documents manuscrits : identification et vérification du scripteur", Thèse de Doctorat, Laboratoire LITIS, Université de Rouen, Mai 2004.
- [Bern86] J. Bernsen. Dynamic thresholding of grey-level images. In Proc. Eighth Int 'l Conf. on Pattern Recognition, pp.1251–1255, 1986.
- [Bert07] Bertolami, R., Uchida, S., Zimmermann, M., Bunke, H.: Non-uniform slant correction for handwritten text line recognition, In: Proceedings of the International Conference on Document Analysis and Recognition, vol.1, pp. 18–22. Curitiba, Brazil 2007.
- [Biad06] Biadisy F., El-Sana S., Habash N., "Online Arabic handwriting recognition using Hidden Markov Models", Proceeding of IWFHR'06, 10th International Workshop on Frontiers in Handwriting Recognition, pp. 85-90, La Baule, France 2006.
- [Bipp97] Bippus R., "1-dimensional and pseudo2-dimensional HMMs for the recognition of German literal amounts", Proceedings of the ICDAR'97, 4th International Conference on Document Analysis and Recognition, Vol. 2, pp. 487-490, Ulm, Germany, August 1997.
- [Bish06] Bishop C. M., "Pattern recognition and machine learning", Springer, 2006.
- [Blac99] M.J. Black, G. Sapiro, "Edges as outliers: Anisotropic smoothing using local image statistics", In Scale-Space Theories in Computer Vision, pp. 259-270, Kerkyra, Greece, 1999.
- [Blum02] Blumenstein M., Cheng C. K., Liu X. Y., "New Preprocessing techniques for Handwritten Word Recognition", Proc. of the 2nd IASTED Conf. on Visualization, Imaging and Image Processing , pp. 480-484, 2002.
- [Boug09] Bougamouza Fateh, Hazmoune Samira et Benmohammed Mohammed, "Nouveaux prétraitements et extraction des caractéristiques pour la

- reconnaissance de mots manuscrits arabes, Image'09 Biskra, Université Mentouri de Constantine, Algérie.
- [Bous06] Boussellaa W., Zahour A., Taconet B., Alimi A., "Segmentation texte/graphique : application au manuscrits arabes anciens", CIFED'06, 9ème Colloque International Francophone sur l'Écrit et le Document, pp.139-144, Fribourg, Suisse, 18-21 Septembre 2006.
- [Bozi89] Bozinovic, R.M., Srihari, S.N.: Off-line cursive script word recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* Vol.11, N°.1, pp.69– 83, 1989.
- [Brak01] Brakensiek, A., Rottland, J., Wallhoff, F., Rigoll, G.: Adaptation of an address reading system to local mail streams. In: *Proceedings of the International Conference on Document Analysis and Recognition*, pp.872–876. Seattle 2001.
- [Brak04] Brakensiek, A., Rigoll, G.: Handwritten address recognition using hidden Markov models. In: Dengel, A., Junker, M., Weisbecker, A. (eds.) *Reading and Learning—Adaptive Content Recognition*, Lecture Notes in Computer Science, Vol. 2956, pp.103–122, Springer, Berlin 2004.
- [Brit01] Britto, A.D.S., Sabourin, R., Bortolozzi, F., Suen, C.Y.: A two-stage HMM-based system for recognizing handwritten numeral strings. In: *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 396–400. Seattle 2001.
- [Brit04] Britto, A.S., Sabourin R., Bortolozzi F. and Suen C.Y., "Foreground and Background Information in an HMM-Based Method for Recognition of Isolated Characters and Numeral Strings", 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR-9), pp.371-376, Tokyo, Japan, October, 2004.
- [Bunk03] Bunke, H.: Recognition of cursive Roman handwriting—Past, present and future. In: *Proceedings of the International Conference on Document Analysis and Recognition*, vol. 1, pp. 448–459 (2003).
- [Burg98] C.J.C. Burges. "A Tutorial on Support Vector Machines for Pattern Recognition". *Knowledge Discovery and Data Mining*, Vol.2, N°.2, 1998.
- [Caes93] Caesar, T., Gloger, J.M., Mandler, E.: Preprocessing and feature extraction for a handwriting recognition system. In: *Proceedings of the International Conference on Document Analysis and Recognition*, pp.408–411, Tsukuba Science City, Japan 1993.
- [Case96] R.G. Casey and E. Lecolinet. A survey of methods and strategies in character segmentation. *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on*, Vol.18, N°.7, pp.690–706, Jul 1996.
- [Chen94] Chen M. Y., Kundu A., Zhou J., "Off-line handwritten word recognition using hidden markov model type stochastic network", *IEEE Transactions on PAMI*, Vol. 16, N°.5, pp. 481-496, May 1994.

- [Chen98] H.D. Cheng, Jim-Rong Chen and Jiguang Li. « Threshold selection based on fuzzy c-partition entropy approach ». *Pattern Recognition*, vol.31, No.7, pp.857-870, 1998.
- [Chev04] S. Chevalier. Reconnaissance d'écriture manuscrite par des techniques markoviennes : une approche bidimensionnelle et générique. Thèse de Doctorat, Université René Descartes - Paris 5, 2004.
- [Choi03] Choisy C., "Modélisation analytique de l'écriture manuscrite par une approche optimal sans segmentation basée sur des champs de Markov", Thèse de Doctorat, Université de Nancy 2, 2003.
- [Chri99] Christian Viard-Gaudin, Pierre Michel Iallican, Stefan Knerr, Philippe Binter: Ireste, The IRESTE On/Off (IRONOFF) Dual Handwriting Database, laboratoire SEI/EP CNRS ,Nantes, France, 1999.
- [Coet04] Coetzer, J., Herbst, B.M., du Preez, J.A.: Offline signature verification using the discrete Radon transform and a hiddenMarkovmodels. *EURASIP J. Appl. Signal Process.* 4, pp.559–571 2004.
- [Coüa02] Coüasnon B., Camillerapp J., "DMOS. Une méthode générique de reconnaissance d'images de documents : évaluation sur 60 000 formulaire du XIXe siècle", CIFED'02, 7ème Colloque International Francophone sur l'Ecrit et le Document, pp.225-234, Hammamet, Tunisie, Octobre 2002.
- [Dehg01a] Dehghan M., Faez K., Ahmadi M., Shridhar M., "Handwritten Farsi (Arabic) word recognition: a holistic approach using discrete HMM", *Pattern Recognition*, Vol. 34, N°. 5, pp. 1057-1065, May 2001.
- [Dehg01b] Dehghan M., Faez K., Ahmadi M., Shridhar M., "Unconstrained Farsi handwritten word recognition using fuzzy vector quantization and hidden Markov models", *Pattern Recognition letters*, Vol.22, N°.2, pp.209-214, February 2001.
- [Dile00] Di Lecce V., Dimauro G., Guerriero A., Impedovo S., Pirlo G., Salzo A., "A New Hybrid Approach for Legal Amount Recognition ", 7th International Workshop on Frontiers in Handwriting Recognition, Amsterdam, The Netherlands, IWFHR'2000, pp. 199-208, September 2000.
- [Ding00] Ding,Y., Kimura, F., Miyake, Y., Shridhar, M, Accuracy improvement of slant estimation for handwritten words. In: *Proceedings of the International Conference on Pattern Recognition*, vol. 4, pp.527–530, Barcelona 2000.
- [Doer95] D. S. Doermann and A. Rosenfeld. Recovery of temporal information from static images handwriting. *International Journal of Computer Vision*, Vol.15, pp.142-164, 1995.
- [Duda72] R.O. Duda and P.E. Hart, Use of the hough transform to detect lines and curves in pictures, N°. 1, pp.11–15, 1972.
- [Dupr03] X. Dupre. Contributions à la reconnaissance de l'écriture cursive `a l'aide de modèles de Markov cachés, Thèse de doctorat, Université Rene Descartes - Paris V, 2003.

- [El-Ab07] El Abed, H., Märgner, V.: Comparison of different preprocessing and feature extraction methods for offline recognition of handwritten Arabic words. In: Proceedings of the International Conference on Document Analysis and Recognition, vol. 2, pp. 974–978. Curitiba, Brazil 2007.
- [El-Ha05] El-Hajj R., Mokbel C. and Likforman L., "HMM-based Arabic Cursive Handwritten Recognition System", The RTST conference (Int'l. Conference on Research Trends in Science and Technology). LAU University Beirut Lebanon, March 2005.
- [El-Ha06] El-Hajj R., Mokbel C., Likforman L., "Reconnaissance de l'écriture Arabe cursive: Combinaison de classifieurs MMCs à fenêtres orientées", actes de CIFED 2006, pp: 271 – 276, Fribourg Suisse, 2006.
- [El-Ha07] 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.
- [El-Ha09] El-Hajj, R., Likforman-Sulem, L., Mokbel, C.: Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition. IEEE Trans. Pattern Anal. Mach. Intell. Vol.31, Vol.7, 2009.
- [El-Ya02] El-Yacoubi M. A., Gilloux M., Bertille J. M., "A statistical approach for phrase location and recognition within a text line: an application to street name recognition", IEEE Transactions on PAMI, Vol.24, N°.2, pp.172-188, February, 2002.
- [El-Ya99] El-Yacoubi, A., Gilloux, M., Sabourin, R., Suen, C.Y.: An HMM-based approach for off-line unconstrained handwritten word modeling and recognition. IEEE Trans. Pattern Anal. Mach. Intell. Vol.21, N°.8, pp.752–760 1999.
- [Essa11] Youssef Es Saady, Ali Rachidi, Mostafa El Yassa and Driss Mammass. Article: AMHCD: A Database for Amazigh Handwritten Character Recognition Research. International Journal of Computer Applications, Vol.27, N°.4, pp:44-48, August 2011. Published by Foundation of Computer Science, New York, USA.
- [Essa12] Y. Es-Saady, Contribution au développement d'approches de reconnaissance automatique de caractères imprimés et manuscrits, de textes et de documents amazighes, Thèse de doctorat, Université Ibn zohr, Agadir, 2012.
- [Etyng06] P. Etyngier, N. Paragios, J.-Y. Audibert., and R. Keriven, "Radon/hough space for pose estimation", Rapport de recherche 06-22, Centre d'Enseignement et de Recherche en Technologies de l'Information et Systèmes, Ecole Nationale des Ponts et Chaussées, Janvier 2006.
- [Fara06] Farah N., Souici L., Sellami M., "Classifiers combination and syntax analysis for Arabic literal amount recognition", Engineering Applications of Artificial Intelligence, Vol. 19, N°. 1, pp. 29-39, February 2006.



- [Feng02] Feng, B., Ding, X., Wu, Y.: Chinese handwriting recognition using hidden Markov models. In: Proceedings of the International Conference on Pattern Recognition, Vol. 3, pp. 212–215, Québec 2002.
- [Forn73] Forney G. D., "The Viterbi Algorithm", Proceedings of IEEE, Vol. 61, N° 3, pp.268-278, March 1973.
- [Fuji07] Fujisawa, H.: Robustness design of industrial strength recognition systems. In: Chaudhuri, B. (ed.) Digital Document Processing: Major Directions and Recent Advances, pp. 185–212. Springer, London, 2007.
- [Gema84] Geman, S., and Geman, D. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence Vol.6, N° 6, pp.721-741, 1984.
- [Gill96] Gilloux(M.). – Real-Time Handwritten Word Recognition Within Large Lexicon. Proc. IWFHR5, p. 301-304, Colchester, sept. 1996.
- [Gink04] M. van Ginkel, C.L. Luengo Hendriks, and L.J. van Vliet, A short introduction to the radon and hough transforms and how they relate to each other, 2004.
- [Gors99] Gorski N., Anisimov V., Augustin E., Baret O., Price D., Simon J, "A2iA check reader: a family of bank check recognition systems", ICDAR'99, 5th International Conference on Document Analysis and Recognition, pp.523-526, Bangalore, India, September 1999.
- [Govi90] Govindan V. K., Shivaprasad A. P., "Character recognition: a review", Pattern Recognition, Vol. 23, N° 7, pp 671-683, 1990.
- [Gran00] Grandidier F., Sabourin R., Suen C.Y., and Gilloux M., "Une nouvelle stratégie pour l'amélioration des jeux de primitives d'un système de reconnaissance de l'écriture", Colloque International Francophone sur l'Écrit et le Document, (CIFED'2000), pp. 111-120, July 2000, Lyon , France.
- [Gran03] Grandidier F., "Un nouvel algorithme de sélection des caractéristiques – Application à la lecture automatique de l'écriture manuscrite", Thèse de doctorat, École de Technologie Supérieure, Université du Québec, Canada, Janvier 2003.
- [Günt03] Günter, S., Bunke, H.: Optimizing the number of states, training iterations and Gaussians in an HMM-based handwritten word recognizer. In: Proceedings of the International Conference on Document Analysis and Recognition, Vol. 1, pp. 472–476. Edinburgh 2003.
- [Guil98] D. Guillevic and C.Y. Suen. HMM-KNN word recognition engine for bank cheque processing. In Proc. of the 13th International Conference on Pattern Recognition, pp.1526–1529, Brisbane, Australia, August 16-20 1998.
- [Guyo95] 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, pp.454–457, 1995.
- [Guyon] I. Guyon, J. Bromley, N. Mati, M. Schenkel, and H. Weissman. Penactive : A neural net system for recognizing on-line handwriting.

- [Hach04] O. Hachour, « Reconnaissance hybride des caractères Arabes imprimé », JEP-TALN, Fès, Avril 2004.
- [Hadj01] K. Hadjar, O. Hitz, R. Ingold. Newspaper page decomposition using split and merge approach. Proceedings of the 5th International Conference on Document Analysis and Recognition. pp.1186-1191, 2001.
- [Hadj03] Hadjar K., Ingold R., Arabic Newspaper Page Segmentation, ICDAR'03, pp. 895-899, Août 2003.
- [Haji05] M.M. Haji "Farsi Handwritten Word Recognition Using Continuous Hidden Markov Models And Structural Features", MSC, Computer Engineering Shiraz University Shiraz, Iran, 2005.
- [Harv03] A. Harvey C. Welwitage and A. Jennings. Whole of word recognition methods for cursive script. APRS Workshop on Digital Image Computing (WDIC), february 2003.
- [Henn01] A. Hennig, N. Sherkat, and R. J. Whitrow. Zone-estimation for multiple lines of handwriting using approximating spline functions.
- [Henn02] Andreas Hennig and Nasser Sherkat. Exploiting zoning based on approximating splines in cursive script recognition. Pattern Recognition, Vol.35,N°2, pp.445-454, 2002.
- [Héro98] Héroux P., Diana S., Ribert A., Trupin E., "Etude de méthodes de classification pour l'identification automatique de classes de formulaires", Actes CIFED'98, 5ème Colloque International Francophone sur l'Ecrit et le Document, pp.463- 472, Québec, Canada, Mai 1998.
- [Hild69] J. Hilditch. Linear skeletons from square cupboards. Machine Intelligence 4 (B.Meltzer and D. Michie, Eds.), pp.404-420, 1969.
- [Houg62] P.V.C. Hough, Method and means for recognizing complex patterns, 1962.
- [Hu61] M.K. Hu. Pattern recognition by moment invariants. In Proc. IRE, page 1428, Sept. 1961.
- [Huan89] Huang, X.D., Jack, M.A.: Semi continuous hidden Markov models for speech signals. Comput. Speech Lang. Vol.3, N°3, pp.239- 251, 1989.
- [Huan90] Huang, X.D., Ariki, Y., Jack, M.A.: Hidden Markov Models for Speech Recognition. No.7 in Information Technology Series. Edinburgh University Press, Edinburgh 1990.
- [Huan93] Huang X.D., Hon H.W., Hwang M.Y. and Lee K.F. "A comparative study of discrete, semi continuous, and continuous hidden Markov models". Computer Speech and Language , Vol.7, pp. 359-368, 1993.
- [Jain00] Jain A.K., Duin R.P.W., Mao J., "Statistical pattern recognition: A review", IEEE Transactions on PAMI, Vol. 22, N°1, pp. 4-37, January 2000.
- [Jang92] B.K. Jang and R.T. Chin. One-pass parallel thinning : analysis, properties, and quantitative evaluation. Pattern Analysis and Machine Intelligence, IEEE Transactions on, Vol.14,N°11, pp.1129-1140, Nov 1992.

- [Jian97] Jiang, H. F., Han, C. C. and Fan, K. C. "A Fast Approach to the Detection and Correction of Skewed Documents", *Pattern Recognition Letters*, vol. 18, N°7, pp. 675-686, 1997.
- [Juan91] B. H. Juang & L. R. Rabiner. Hidden Markov models for speech recognition. *Technometrics*, Vol. 33, N° 3, pp.251–272, 1991.
- [Kalt93] Kaltenmeier, A., Caesar, T., Gloger, J.M., Mandler, E.: Sophisticated topology of hidden Markov models for cursive script recognition. In: *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 139–142 1993.
- [Kapo03] Kapoor, R., Bagai, D., and Kamal, T. Representation and extraction of nodal features of DevNagri letters. *ICVGIP*, 2003.
- [Kapu85] J.N. Kapur, P.K. Sahoo, A.K.C. Wong. « A New method for gray-level picture threshold using the entropy of the histogram ». *Graphical Models and Image Processing*, 29, 1985
- [Katk03] V. Katkovnik, K. Egiazarian, J. Astola, Application of the ICI principle to window size adaptive median filtering, *Signal Processing*, N. 83, pp. 251-257, 2003.
- [Kauf00] Kaufmann G., Bunke H., "Detection and correction of recognition errors in check reading", *International Journal on Document Analysis and Recognition, IJDAR*, Vol. 2, N° 4, pp. 211-221, June 2000.
- [Kava00] Kavallieratou, E., Fakotakis, N. and Kokkinakis, G. "A Slant Removal Algorithm", *Pattern Recognition*, pp. 1261-1262, 2000.
- [Kava01] Ergina Kavallieratou, Nikos Fakotakis, and George K. Kokkinakis. Slant estimation algorithm for ocr systems. *Pattern Recognition*, Vol.34, N°12 pp.2515–2522, 2001.
- [Kefa10] A.Kefali, T.Sari, M.Sellami "Evaluation of several binarization techniques for old Arabic documents images" *Laboratoire de Recherche en Informatique (LRI), Université Badji Mokhtar - Annaba, Algérie*, 2010.
- [Kess09] Y.Kessentini, « Modèles de Markov multi-flux pour la reconnaissance de l'écriture manuscrite multi-scripts », *Thèse de doctorat, Université de Rouen*, 2009.
- [Khor00] Khorsheed M. S., "Automatic recognition of words in Arabic manuscripts", PhD thesis, Churchill college, University of Cambridge, England, Also available as University of Cambridge, Computer Laboratory Technical Report N°. 495, June 2000.
- [Khor02] Khorsheed M.S., "Off-line arabic character recognition-a review", *Pattern Analysis Applications*, Vol. 5, N°.1, pp.31-45, May 2002.
- [Khor03] Khorsheed M.S., "Recognising handwritten Arabic manuscripts using a single hidden Markov model", *Pattern Recognition Letters*, Vol. 24, N°.14, pp. 2235-2242, October 2003.

- [Khor07] M. S. Khorshed, "Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK)", *Pattern Recognition Letters*, vol. 28, pp. 1563–1571, September 2007.
- [Kim96] H.J. Kim, J.W. Jung, and S.K. Kim. On-line Chinese character recognition using ARTbased stroke classification. *Pattern Recognition Letters*, Vol.17, N°.12, pp.1311–1322, 1996.
- [Kim98] G. Kim and V. Govindaraju. Handwritten phrase recognition as applied to street name images. *Pattern Recognition*, Vol.31, N°.1, pp.41–51, 1998.
- [Klas02] Klassen T. J., "Towards the on-line recognition of Arabic characters", *Proceedings of IJCNN'02, International Joint Conference on Neural Networks*, pp. 1900-1905, Honolulu, Hawaii, USA, May 2002.
- [Koer02] Koerich, A.L., Leydier, Y., Sabourin, R., Suen, C.Y.: A hybrid large vocabulary handwritten word recognition system using neuronal networks with hidden Markov models. In: *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, pp. 99–104. Niagara on the Lake, Canada 2002.
- [Koer06] Koerich, A.L., Britto, A.S., de Oliviera, L.E.S., Sabourin, R.: Fusing high- and low-level features for handwritten word recognition. In: *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, pp. 151–156. La Baule, France 2006.
- [Korn98] P. Kornprobst, *Contributions à la Restauration d'Images et à l'Analyse de Séquences: Approches Variationnelles et Solutions de Viscosité*, Thèse de Doctorat, Université de Nice-Sophia Antipolis, 1998.
- [Kuhn94] Kuhn (T.), Niemann (H.) et Schnkat-Talamazzini(E.G.). – Ergodic Hidden Markov Models and Polygrams for Language Modeling. *Proc. ICASSP-94*, Vol.1, pp. 357-360, 1994.
- [Kund07] Kundu, A., Hines, T., Phillips, J., Huyck, B.D., Van Guilder, L.C.: Arabic handwriting recognition using variable duration HMM. In: *Proceedings of the International Conference on Document Analysis and Recognition*, Vol. 2, pp. 644–648. Curitiba, Brazil 2007.
- [Kund98] A. Kundu, H. He, and M.Y. Chen. Alternative to variable duration HMM in handwriting recognition. *IEEE Trans. on Pattern Analysis and Machine Recognition*, Vol.20, N°.11, pp.1275–1281, 1998.
- [Kuo94] 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, N°. 8, pp.842–848, 1994.
- [Laak98] J. Laaksonen and E. Oja. Learning subspace classifiers and error-corrective feature extraction. *International Journal of Pattern Recognition and Artificial Intelligence*, Vol.12, N°.4, pp.423–436, 1998.
- [Lam92] L. Lam, S.W. Lee, and C.Y. Suen. Thinning methodologies: A comprehensive survey. *PAMI*, Vol.14, N°.9, pp:869-885, September 1992.

- [Lamo96] Lamouche I., Bellissant C., «Séparation recto/verso d'images de manuscrits anciens », Actes de CNED'96, Colloque National sur l'Écrit et le Document, pp. 199-206, Nantes, juillet 1996.
- [Lebo00] Lebourgeois, F., Emptoz, H., Vigne, H., "RASADE / Reconnaissance Automatique des Structures Associées aux Documents Ecrits", Conf.Int. Francophone sur l'Écrit et le Document (CIFED'00), Lyon, pp. 281-294, 2000.
- [Lebo07] LeBourgeois F., Emptoz H., "DEBORA: Digital accEs to BOoks of the RenaissAncE", International Journal on Document Analysis and Recognition, IJDAR , Vol. 9, N°. 2, pp.193-221, April 2007.
- [Leco90] Eric Lecolinet. Segmentation d'images de mots manuscrits : Application à la lecture de chaîne de caractères majuscules alphanumériques et à la lecture de l'écriture manuscrites. Thèse de doctorat, Université Paris 6, Mars 1990.
- [Lecu07] Y. LeCun, S. Chopra, M. Ranzato, and F.-J. Huang. Energy-based models in document recognition and computer vision. In ICDAR '07 : Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR), IEEE Computer Society, Vol.1, pp.337–341, 2007.
- [Lecu98] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition Proceedings of the IEEE, Vol.86, N°.11, pp.2278–2324, 1998.
- [Lecu98b] LeCun, Y. The MNIST database. <http://yann.lecun.com/exdb/mnist/index.html>, 1998.
- [Lema07] Lemaître M., "Approche markovienne bidimensionnelle d'analyse et de reconnaissance de documents manuscrits", Thèse de Doctorat, Université Paris 5 René Descartes, 2007.
- [Levi92] E. Levin & R. Pieraccini. Dynamic planar warping for optical character recognition. IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 3, pp.149–152, 1992.
- [Li92] Y. Li. Reforming the theory of invariant moments for pattern recognition. Pattern Recognition, Vol.25, pp.723–730, 1992.
- [Likf07] L. Likforman-Sulem, A. Zahour, B. Taconet, Text line segmentation of historical documents: a survey, International Journal on Document Analysis and Recognition , Vol.9, N°.2, pp. 123–138, 2007.
- [Lins94] Lins R.D., Guimaraes Neto M., França Neto L., Galdino Rosa L., «An Environment for Processing Images of Historical Documents», Microprocessing and Microprogramming, Vol.40, pp. 939-942, 1994.
- [Liwi05] Liwicki, M., Bunke, H.: Handwriting recognition of whiteboard notes. In: Proceedings of the 12th Conference of the International Graphonomics Society, pp.118–122, 2005.
- [Liwi05a] Liwicki, M., Bunke, H.: IAM-OnDB—an on-line English sentence database acquired from handwritten text on a whiteboard. In: Proceedings of the

- International Conference on Document Analysis and Recognition, Vol. 2, pp. 956–961. Seoul, Korea 2005.
- [Lori05] Lorigo L., Govindaraju V. "Segmentation and Pre-Recognition of Arabic Handwriting", ICDAR 05, Corée du Sud, pp. 605-609 2005.
- [Lori06] Lorigo L.M., Govindaraju V., "Offline Arabic handwriting recognition: a survey", IEEE Transactions on PAMI, Vol. 28, N°5, pp. 712-724, May 2006.
- [Loul09] G. Louloudi, B.Gatos, I.Pratikakis, C.Halatsis, "Text line and word segmentation of handwritten documents", Pattern Recognition, Vol. 42, pp. 3169 – 3183, Department of Informatics and Telecommunications, University of Athens, Greece 2009.
- [Lu96] Y. Lu and M. Shridhar. Character segmentation in handwritten words : An overview. PR, Vol.29, N°1, pp.77–96, January 1996.
- [Madh99] Madhvanath, S., Kim, G., Govindaraju, V., Chaincode contour processing for handwritten word recognition. IEEE Trans. Pattern Anal. Mach. Intell. Vol. 21, N°9, pp.928–932, 1999.
- [Mahj99] M.A.Mahjoub, « Application des modèles de Markov Cachés stationnaires et non stationnaires et non stationnaires à la reconnaissance en ligne de l'écriture arabe » Thèse de doctorat, ENIT, 1999.
- [Mait85] Henri MAITRE « Un panorama de la transformation de Hough » École Nationale Supérieure des Télécommunications, Laboratoire Image, Département Images, Sons et Vidéo, traitement de signal, vol.2, N°4 -1985.
- [Marg05] Margner V., Pechwitz M., El Abed H., Arabic handwriting recognition competition, 8th International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 70–74, 2005.
- [Märg06] Märgner V., El Abed H., Pechwitz M., "Off-line handwritten word recognition using HMM- a character approach without explicit segmentation", Actes CIFED'06, 9ème Colloque International Francophone sur l'Écrit et le Document, pp. 259-264, Fribourg, Suisse, September 2006.
- [Marg07] MÄRGNER V., EL ABED H., ICDAR 2007 Arabic Handwriting Recognition Competition, 9th International Conference on Document Analysis and Recognition (ICDAR), vol. 2, pp. 1274–1278, 2007.
- [Mart01] Marti, U.V., Bunke, H.: Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition systems. Int. J. Pattern Recogn. Artif.Intell. Vol.15, N°1, pp.65–90 2001.
- [Mart02] Marti, U.V., Bunke, H.: The IAM-database: an English sentence database for offline handwriting recognition. Int. J.Doc. Anal. Recogn. Vol.5, N°1, pp.39–46, 2002.
- [Masm00] Masmoudi S., Amiri H., "Reconnaissance de mots arabes manuscrits par modélisation markovienne", Actes CIFED'00, 6ème Colloque International Francophone sur l'Écrit et le Document, pp.473-482, Lyon, France, 2000.

- [Mena08] F. Menasri. Contributions à la reconnaissance de l'écriture arabe manuscrite, Thèse de doctorat, Université Descartes- Paris, 2008.
- [Meng00] Mengucci M., Granado I., «Morphological Segmentation of text and figures in Renaissance books (XVI Century) », in *Mathematical Morphology and its applications to image processing*, J. Goutsias, L. Vincent, D. Bloomberg(eds), pp. 397-404, Kluwer, 2000.
- [Mezg02] Neila Mezghani, Amar Mitiche, and Mohamed Cheriet, On-line recognition of handwritten arabic characters using a kohonen neural network. In *IWFHR '02 : Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02)*, IEEE Computer Society, page 490, 2002.
- [Mezg03] Neila Mezghani, Mohamed Cheriet, and Amar Mitiche, Combination of pruned kohonen maps for on-line arabic characters recognition, In *ICDAR '03 : Proceedings of the Seventh International Conference on Document Analysis and Recognition*, IEEE Computer Society, page 900, 2003.
- [Mezg05a] Mezghani N., Mitiche A., Cheriet M., "A new representation of shape and its use for high performance in online Arabic character recognition by an associative memory", *International Journal on Document Analysis and Recognition, IJDAR*, Vol. 7, N°. 4, pp. 201-210, September 2005.
- [Mezg05b] Mezghani N., "Densités de probabilité d'entropie maximale et memoires associatives pour la reconnaissance en ligne de caractères Arabes", Thèse de doctorat, Institut national de la recherche scientifique (INRS), centre Énergie, Matériaux et Télécommunications, Canada, Juin 2005.
- [Mile97] Miled H., Olivier C., Cheriet M., Lecourtier Y., "Coupling observation/letter for a Markovian Modelisation applied to the recognition of arabic handwriting", *IEEE Proc. 4th International conference on document analysis and recognition (ICDAR'97)*, pp.580-583, Ulm, Germany, 1997.
- [Miti02] A. Mitiche N. Mezghani and M. Cheriet, Reconnaissance en-ligne de caracteres arabes manuscrits par un réseau de kohonen, In *Vision Interface*, IEEE Computer Society, pp.86–191, Calgary, Canada, 2002.
- [Mori02] Morita, M., Sabourin, R., Bortolozzi, F., Suen, C.Y.: Segmentation and recognition of handwritten dates. In: *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, pp. 105–110. Niagara on the Lake, Canada 2002.
- [Mota97] Motawa D., Amin A., Sabourin R., "Segmentation of Arabic cursive script", *Proceedings of the ICDAR'97, the 4th Conference on Document Analysis and Recognition*, Vol. 2, pp. 625-628, Ulm, Germany, August 1997.
- [Moza05] S.Mozaffari, K.Faez, and M.Ziaratban, "Structural Decomposition and Statistical Description of Farsi/Arabic Handwritten Numeric character," in *Proc. International Conference on document analysis and Recognition*. Seoul, Korea, pp.237-241, 2005.

- [Naga07] P. Nagabhushan, S.A. Angadi, and B.S. Anami. Geometric model and projection based algorithms for tilt correction and extraction of ascenders / descenders for cursive word recognition. *Signal Processing, Communications and Networking, ICSCN '07. International Conference on*, pp.488–491, 22-24 February 2007.
- [Nata01] Natarajan, P., Lu, Z., Schwartz, R., Bazzi, I., Makhoul, J.: Multilingual machine printed OCR. *Int. J. Pattern Recog. Artif.Intell*, Vol.15, N°.1, pp.43–63 2001.
- [Nata08] Natarajan, P., Saleem, S., Prasad, R., MacRostie, E., Subramanian, K. : Multilingual offline handwriting recognition using hidden Markov models: a script-independent approach. In: Doermann, D.S., Jaeger, S. (eds.) *Arabic and Chinese Handwriting Recognition: SACH 2006 Selected Papers, Selected Papers, Lecture Notes in Computer Science*, vol. 4768, pp. 231–250, Springer, Berlin 2008.
- [Nibl00] W. Niblack. *An introduction to digital image processing*. Prentice Hall, July 1986.
- [Nico06] Nicolas S., "Segmentation par champs aléatoires pour l'indexation d'images de documents", Thèse de Doctorat, Laboratoire LITIS, Université de Rouen, Décembre 2006.
- [O'boy94] O'Boyle(P.), Owens(M.) et Smith (F.J.). –A weighted average n-gram model of natural language. *Computer Speech and Language*, 8, pp. 337-349, 1994.
- [Oliv01] Oliveira L.S., Benahmed N., Sabourin R., Bortolozzi F., and Suen C.Y., "Feature Subset Selection Using Genetic Algorithm for Handwritten Digit Recognition", 14th Brazilian Symposium on Computer Graphics and Image Processing, (SIBGRAP'2001), pp. 362-369, October 2001, Brazil.
- [Olsz01] Robert T. Olszewski, "Generalized Feature Extraction for Structural Pattern Recognition in Time Series Data", PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2001.
- [Otsu78] N.Otsu. « A threshold selection method from grey-level histograms ». *IEEE Trans. Syst. Man. Cybern*, vol.SMC-8, 1978.
- [Pal01] Pal, U., Belaïd, A., and Choisy, C. Water reservoir based approach for touching numeral segmentation. *International Conference on Document Analysis and Recognition*, 892, 2001
- [Park96] H. Park & S. Lee. An HMMRF-based statistical approach for off-line handwritten character recognition. *International Conference on Pattern Recognition*, Vol. 2, pp.320–324, 1996.
- [Pech02] Pechwitz M., Maddouri S. S., Margner V., Ellouze N., Amiri H., IFN/ENIT-Database of Handwritten Arabic Words, *Colloque International Francophone sur l'Écrit et le Document (CIFED)*, pp. 127-136, 2002.
- [Pech03] M. Pechwitz, V. Märgner, "HMM based approach for handwritten Arabic word recognition using the IFN/ENIT– database", *Proceeding of ICDAR'03*, 7th



- International Conference on Document Analysis and Recognition, Vol. 2, pp. 890-894, Edinburgh, Scotland, 2003.
- [Pech06] Pechwitz M., Märgner V., El Abed H, "Comparison of two different feature set for off-line recognition of handwritten Arabic words", Proceedings of IWFHR'06, 10th International Workshop on Frontiers in Handwriting Recognition, pp.109-114, La Baule, France, October 2006.
- [Pete04] Peter B., "Arabic handwriting recognition", Master Thesis, School of Informatics, University of Edinburgh, Scotland, 2004.
- [Kove96] Perter Kovesi, Invariant Mesures of Image Features from phase Information, rapport thèse, département de psychologie, université de Western Australia, May 1996.
- [Pinal01] Pinales J.R., "Reconnaissance hors-ligne de l'écriture cursive par l'utilisation de modèles perceptifs et neuronaux", Thèse de doctorat, ENST Paris, 2001.
- [Plam00] Plamondon R., Srihari S. N. "On-Line and Off-Line Handwriting Recognition: A comprehensive survey", IEEE Trans on Pattern Analysis and Machine Intelligence, Vol. 22, N°1, pp.63-84, 2000.
- [Plöt08] Plötz, T., Thurau, C., Fink, G.A.: Camera-based whiteboard reading: new approaches to a challenging task. In: Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition, pp. 385–390, Montreal, Canada 2008.
- [Pois05] Poisson E., "Architecture et apprentissage d'un système hybride neuro markovien pour la reconnaissance de l'écriture manuscrite en-ligne", Thèse de Doctorat, Université de Nantes, 2005.
- [Pun80] T.Pun. « A New method for gray-level picture threshold using the entropy of the histogram ». Signal processing, vol.2, N°3, 1980.
- [Rabi89] Rabiner L. R., "A tutorial on hidden markov models and selected applications in speech recognition", Proceedings of IEEE, Vol. 77, N°. 2, pp. 257-286, February 1989.
- [Rabi93] Rabiner, L., and Juang, B. Fundamentals of speech recognition. Prentice Hall,1993.
- [Rein71] Ch. H. Reinsch. Smoothing by spline functions ii, March 1971.
- [Roch95] Rocha J, Pavlidis T. "Character Recognition without segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, pp. 903-909, 1995.
- [Ruto66] D. Rutovitz: Pattern recognition. Journal of Royal Statistical Society, 129:530, 1966.
- [Safa05] R.Safabakhsh and P.Adibi, "Nastaaligh Handwritten Word Recognition Using a Continuous-density Variable-duration HMM", The Arabian Journal For Science and Engineering, Vol.30, pp.95-118, 2005.

- [Saon97] Saon, G. Modèles markoviens uni- et bidimensionnels pour la reconnaissance de l'écriture manuscrite hors-ligne. Thèse de doctorat, Université Henri Poincaré – Nancy 1, 1997.
- [Sauv00] J. Sauvola and M. Pietikäinen. Adaptive document image binarization. *Pattern Recognition*, Vol.33,N°.2, pp.225 – 236, 2000.
- [Sayr73] K. Sayre. Machine recognition of handwritten words : A project report. *Pattern Recognition*, Vol.5, pp.213–228, 1973.
- [Scha03] Schambach, M.P.: Determination of the number of writing variants with an HMM based cursive word recognition system. In: *Proceedings of the International Conference on Document Analysis and Recognition*, Vol. 1, pp. 119–123, Edinburgh 2003.
- [Scha05] Schambach, M.P.: Fast script word recognition with very large vocabulary. In: *Proceedings of the International Conference on Document Analysis and Recognition*, vol. 1, pp. 9–13, Seoul, Korea 2005.
- [Scha08] Schambach, M.P., Rottland, J., Alary, T.: How to convert a Latin handwriting recognition system to Arabic. In: *Proceedings of the International Conference on Document Analysis and Recognition*, 2008.
- [Schw96] H. Schwenk and M. Milgram. Constraint tangent distance for on-line character recognition. In *International Conference on Pattern Recognition*, vol.D, pp.515-519, aout 1996.
- [Sell05] Sari T., Sellami M., "Cursive Arabic script segmentation and recognition system", *International Journal of Computers and Applications*, Vol. 27, N°. 3, 2005.
- [Seni98] Senior, A.W., Robinson, A.J.: An off-line cursive handwriting recognition system. *IEEE Trans. Pattern Anal. Mach.Intell.* Vol.20, N°.3, pp.309–321 1998.
- [Shah06] Shahabi S., Rahmati M., "Comparison of Gabor-based features for writer identification of Farsi/Arabic handwriting", *IWFHR'06, 10th International Workshop on Frontiers in Handwriting Recognition*, pp. 545-550, La Baule, France, October 2006.
- [Shen99] D. Shen and H.H.S. Ip. Discriminative wavelet shape descriptors for recognition of 2-D patterns. *Pattern Recognition*, Vol.32, pp.151–165, 1999.
- [Sher05] N. Sherkat. Influence of zoning on whole word recognition. *Document Analysis and Recognition, Proceedings, Eighth International Conference on*, Vol. 2, pp.1085– 1089, 29 Aug.-1 Sept. 2005.
- [Shi04] Z. Shi and V.Govindaraju. line separation for complex document images using Fuzzy run length. In *Int. Xorkshop on Document Image Analysis for Librairies*, 2004.
- [Shri95] Shridhar, M. and Kimura, F. "Handwritten Address Interpretation Using Word Recognition with and without Lexicon", *IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp.2341-2346, 1995.

- [Slim09] F.Slimane, S.Kanoun, J.Hennebert, Adel M. Alimi et R.Ingold, "Modèles de Markov Cachés et Modèle de Longueur pour la Reconnaissance de l'Écriture Arabe à Basse Résolution", MajecSTIC, Avignon, France, novembre 2009.
- [Soui06] Souici-Meslati L, "Reconnaissance des mots arabes manuscrits par integration neuro-symbolique", Thèse de Doctorat d'Etat, Labo. LRI, Département d'informatique, Université d'Annaba, Algérie, Février 2006.
- [Srig01] Sriganesh Madhvanath, Venu Krpasundar, and Venu Govindaraju. Syntactic methodology of pruning large lexicons in cursive script recognition. *Pattern Recognition*, Vol.34, N°.1, pp.37–46, 2001.
- [Srih06a] Ball G., Srihari S. N., Srinivasan H, "Segmentation-free and segmentation dependent approaches to Arabic word spotting", IWFHR'06, 10th International Workshop on Frontiers in Handwriting Recognition, pp.53-58, La Baule, France, October 2006.
- [Srih06b] Srihari S. N., Srinivasan H, Hang C., Shetty S., "Spotting words in Latin, Devanagari and Arabic scripts", *Vivek: Indian Journal of Artificial Intelligence*, Vol. 16, N°. 3, pp. 2-9, 2006.
- [Srih97] S.N. Srihari & E.J. Keubert. Integration of handwritten address interpretation technology into the united states postal service remote computer reader system. *ICDAR*, pp.892–896, 1997.
- [Srih99] Srihari S., Yang W., Govindaraju V., "Information theoretic analysis of postal address fields for automatic address interpretation", *Proceedings of ICDAR'99, 5th International Conference on Document Analysis and Recognition*, pp.309-312, Bangalore, India, September 1999.
- [Su07] Su, T.H., Zhang, T.W., Huang, H.J., Zhou, Y.: HMM-based recognizer with segmentation-free strategy for unconstrained Chinese handwriting text. In: *Proceedings of the International Conference on Document Analysis and Recognition*, Vol. 1, pp. 133–137. Curitiba, Brazil 2007.
- [Sun97] Changming Sun and Deyi Si "Skew and Slant Correction for Document Images Using Gradient Direction", *IEEE International Conference on Document Analysis and Recognition*, vol.1, pp.142-146, 1997.
- [Syia06] Syiam M., Nazmy T. M., Fahmy A. E., Fathi H., Ali K., "Histogram clustering and hybrid classifier for handwritten arabic characters recognition", *Proceedings of the 24th IASTED International Multi-Conference Signal Processing, Pattern Recognition, and Applications*, pp. 44-49, Innsbruck, Austria, 15-17 February 2006.
- [Tapp84] Tappert, C.C., "Adaptive on-line handwriting recognition", *Proceedings of ICPR'84, 7th International Conference on Pattern Recognition*, pp. 1004-1007, Montreal, Canada, July/August 1984.
- [Tapp90] C.C. Tappert, C.Y. Suen, and T. Wakara, The state of the art in on-line handwriting recognition, *IEEE Trans. on Pattern Analysis and Machine Recognition*, Vol.12, N°.8, pp.787–808, 1990.

- [Tay01] Tay Y. H., Marzouki K., Lallican P.M., Knerr S., Viard-Gaudin C., "An Analytical Handwritten Word Recognition System with Word-level Discriminant Training", Sixth International Conference on Document Analysis and Recognition (ICDAR'01), pp.726-731, 2001.
- [Tay01b] Y. Tay, P. Lallican, M. Khalid, C. Viard-Gaudin, and S. Knerr. An offline cursive handwritten word recognition system, 2001.
- [Toma98] C. Tomasi, R. Manduchi, Bilateral filtering for gray and color images, In International Conference on Computer Vision, pp. 839-846, Bombay, India, 1998.
- [Touj02] S.Touj, N.Ben Amara, H.Amiri : "Extraction de Caractéristiques Globales de l'écriture Arabe Manuscrite par Transformation de Hough", GEI'2002, Hammamet, Tunisie, Mars 2002.
- [Touj07] Touj, S.M., Ben Amara, N.E., Amiri, H.: A hybrid approach for off-line Arabic handwriting recognition based on a planar hidden Markov modeling. In: Proceedings of the International Conference on Document Analysis and Recognition, Vol. 2, pp. 964–968. Curitiba, Brazil 2007.
- [Trie95] Trier, O.D., Taxt, T.: Evaluation of binarization methods for document images. IEEE Trans. Pattern Anal. Mach. Intell. Vol.17, N°3, pp.312–315 1995.
- [Uchi01] Uchida, S., Taira, E. and Sakoe, H. "Nonuniform Slant Correction Using Dynamic Programming", IEEE International Conference on Document Analysis and Recognition, pp. 434-438, Septembre 2001.
- [Vapn95] Vapnik V., "The nature of statistical learning theory", Springer, 1995.
- [Vinc00] Vinciarelli A. and Luettin J.. "Off-line cursive script recognition based on continuous density hmm". In Seventh International Workshop on Frontiers in Handwriting Recognition (IWFHR'00), page 493-498, Amsterdam, September 2000.
- [Vinc01] A. Vinciarelli and J. Luettin. A new normalization technique for cursive handwritten words. Pattern Recognition Letters, Vol. 22, N°9, pp.1043–1050, 2001.
- [Vinc04] Vinciarelli, A., Bengio, S., Bunke, H.: Offline recognition of unconstrained handwritten texts using HMMs and statistical language models. IEEE Trans. Pattern Anal. Mach. Intell, Vol.26, N°6, pp.709–720, 2004.
- [Wall95] Å. Wallin and O. Kübler. Complete sets of Zernike moment invariants and the role of the pseudoinvariants. IEEE Trans. on Pattern Analysis and Machine Recognition, Vol.17, N°11, November 1995.
- [Wang97] Jiren Wang, Maylor K. H. Leung, and Siu Cheung Hui. Cursive word reference line detection. Pattern Recognition, Vol.30, N°3, pp.503– 511, 1997.
- [Werf99] E. Van Der Werf and R. Duin. Improved image features by training non-linear diablo networks. Proc. ASCI'99 5th Conf. on Advanced School for Computing and Imaging (ASCI'99), pp.229–234, June 1999.

- [Wien03] Wienecke, M., Fink, G.A., Sagerer, G.: Toward automatic video-based whiteboard reading. In: Proceedings of the International Conference on Document Analysis and Recognition, pp.87–91. IEEE, Edinburgh, Scotland 2003.
- [Wien05] Wienecke, M., Fink, G.A., Sagerer, G.: Toward automatic video-based whiteboard reading. *Int. J. Doc. Anal. Recogn*, Vol.7, N°.2–3, pp.188–200 2005.
- [Wood98] J.Wood. Invariant pattern recognition : A review. *Pattern Recognition*, Vol.29, N°.1, pp.1–17, 1996.
- [Xue02] Xu, Q., Kim, J.H., Lam, L., Suen, C.Y.: Recognition of handwritten month words on bank cheques. In: Proceedings of the International Workshop on Frontiers in Handwriting Recognition, pp.111–116, Niagara on the Lake, Canada 2002.
- [Xue06] Xue H, Govindaraju V., "Hidden Markov models combining discrete symbols and continuous attributes in Handwriting recognition", *IEEE Transactions on PAMI*, Vol. 28, N°.3, pp.458-462, March 2006.
- [Yani07] B. Yanikoglu and P. Sandon. Segmentation of off-line cursive handwriting using linear programming, 1998.
- [Yin08] F. Yin, C.Liu. Handwritten text line segmentation by clustering with distance metric learning. In Proc. 11th IFCHR, pp.229-234, 2008
- [Youn01] Young, S., Evermann, G., Kershaw, D., Moore, D., Odell, J., Ollason, D., Valtchev, V., Woodland, P., *The HTK Book*, Cambridge University Engineering Dept, 2001.
- [Youn02] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev and P. Woodland, *The HTK book (for HTK version 3.1)*, Microsoft Corporation and Cambridge University Engineering Department, 2002.
- [Youn06] Young & all. *The htk book (for htk version 3.4)*. Cambridge University Engineering Department, 2006.
- [Youn93] S. Young. *The htk hidden markov model toolkit : Design and philosophy*, 1993.
- [Zaho07] 1.Zahour, L.Likforman-sulem, W.boussellaa, and B.taconet. Text line segmentation of historical Arabic documents. In 9th Int. Conf. on Document Analysis and Recognition, pp.138-142, 2007.
- [Zerm07] Zermi N., Ramdani M., Bedda M., "Arabic handwriting word recognition based on hybride HMM/ANN approach", *International Journal of Soft Computing*, Vol. 2, N°. 1, pp. 5-10, 2007.
- [Zhan84] T.Y. Zhang et C.Y .Suen, "A fast parallel algorithm for thinning digital patterns", *Communications of the ACM*, 27(3): 236–240, mars 1984.
- [Zhon99] David X. Zhong and Hong Yan. Pattern skeletonization using run-length-wise processing for intersection distortion problem. *Pattern Recogn. Lett.*, Vol.20, N°.8, pp.833–846, 1999.

- 
- [Zimm02] Zimmermann, M., Bunke, H.: HiddenMarkov model length optimization for handwriting recognition systems. In: Proceedings of the International Workshop on Frontiers in Handwriting Recognition, pp. 369–374. Niagara on the Lake, Canada 2002.
- [Zoua05] Zouari H., Heutte L., Lecourtier Y., "Controlling the diversity in classifier ensembles through a measure of agreement", Pattern Recognition, Vol. 38, N°.1, pp. 2195-2199, November 2005.