N° d'ordre : 28/2010 – M/EL

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
**UNIVERSITE DES SCIENCES ET DE LA THECHNOLOGIE HOUARI BOUMEDIENE**

**FACULTE D'ELECTRONIQUE ET INFORMATIQUE**


MEMOIRE

Présenté pour l'obtention du diplôme de MAGISTER

**EN : ELECTRONIQUE**

Spécialité : Traitement du Signal et des Images

**Par : BRIK Youcef**

SUJET

# Reconnaissance des chiffres Manuscrits par les Modèles de Markov Cachés Continus

Soutenu publiquement le 15 /07 /2010 , devant le jury composé de :

| | | | |
|---|---|---|---|
| **Mme. A. KEMMOUCHE** | M. C. A | à l'USTHB | Président |
| **Mr. Y. CHIBANI** | Professeur | à l'USTHB | Directeur de mémoire |
| **Mme. A. SERIR** | M. C. A | à l'USTHB | Examinateur |
| **Mme. L. FALEK** | M. C. A | à l'USTHB | Examinateur |
| **Mlle. H. NEMMOUR** | M. C. B | à l'USTHB | Invité |

… à mes parents et ma famille.

# REMERCIEMENTS

Je remercie sincèrement mon directeur de mémoire **Youcef Chibani** professeur au département d'électronique à USTHB, sans son initiative ce projet n'aurait pas été possible. Je tiens à lui exprimer toute ma reconnaissance pour son dévouement, la confiance qu'il m'a accordée, sa rigueur et la qualité des commentaires et suggestions dont il m'a fait part. Mr. Chibani a été vraiment un très bon formateur, toujours à l'écoute de nouvelles idées, toujours plein d'enthousiasme, toujours prêt à me re-motiver.

Je remercie tous les membres de mon jury de mémoire qui ont pris de leurs temps pour lire et juger ce travail ainsi que pour leur déplacement le jour de la soutenance.

Je voudrais aussi remercier ici Hassiba Nemmour (maître assistante, USTHB) et Boudjellel A.Wahab (maître assistant, EMP) pour leur aide et leur gentillesse, ils m'ont présenté un excellant exemple de discipline, générosité et compétence scientifique.

Je tiens ensuite à remercier l'ensemble de mes collègues du LCPTS de la FEI pour leur sympathie et bonne humeur.

Merci à mes parents et à ma grande famille pour leur soutien et pour leur aide dans la préparation de ce mémoire et pour leurs encouragements.

Pour terminer, nous remercions toute personne ayant participé de prés ou de loin pour la réalisation de ce travail.

# SOMMAIRE

## Chapitre 3 :
## Description du Système de Reconnaissance des Chiffres Manuscrits



## Chapitre 4 : Résultats Expérimentaux

## LISTE DES FIGURES

### Chapitre 1



### Chapitre 2



### Chapitre 3



### Chapitre 4

## LISTE DES TABLEAUX

# Liste des Abréviations et sigles

| | |
|---|---|
| HMM : | Hidden Markov Model / modèle de Markov cachés. |
| DHMMs : | Discret Hidden Markov Models/ modèle de Markov cachés discret. |
| CHMMs : | continuous Hidden Markov Models/ modèle de Markov cachés continus. |
| GMMs : | Gaussian Mixture Model/ Modèle de mixture de gaussiennes. |
| PDA : | personal digital assistant / Assistant numérique personnel. |
| TDNN : | Time delay neural network / réseaux de neurones à délai. |
| NN : | Neural Network / Réseau de neurones. |
| Knn/Kppv: | k nearest neighbors / k plus proches voisins. |
| SVM : | Support Vector Machin / Séparateur à vaste marge. |
| fdp : | Fonction de densité de probabilité. |
| MLE : | Maximum Likelihood Estimation / estimation du maximum de vraisemblance. |
| EM : | Expectation-Maximization/ Espérance-Maximisation. |
| CC : | Caractéristiques classiques. |
| Cforme : | Caractéristiques liées à la forme. |
| Cfond : | Caractéristiques liées au fond. |
| NIST : | National Institute of Standards and Technology. |
| SD19 : | Special Database 19. |
| hsf : | Handwritten Sample Forms |
| TMBR : | Taux Moyen de Bon Reconaissance. |
| TRC : | Taux de Reconnaissance par classe. |
| L : | Largeur de la fenêtre glissante. |
| N : | Nombre d'états. |
| T : | Longueur d'observation (nombre de trames). |
| M : | Nombre de symboles observables. |
| R : | Nombre d'échantillons. |
| K : | Nombre de gaussiennes. |
| $i,j,k,n,r$ : | Indices. |
| X(t) : | Variable aléatoire. |
| P : | Probabilité. |
| $O$, $O_t$ : | Suite d'observation, observation à l'instant $t$ . |

$Q$, $q_t$ : Suite d'états, états à l'instant t.

S, $s_i$ : Ensemble d'états, $i^{ème}$ états.

V, $v_i$ : Ensemble de symboles d'observations, $i^{ème}$ symbole.

A, $a_{ij}$ : Matrice de transitions, la probabilité que le modèle évolue de l'état $s_i$ vers l'état $s_j$.

B, $b_j(k)$ : Matrice de probabilité d'observation, la probabilité d'observer $v_k$ dans l'état $s_j$.

$\Pi$, $\pi_i$ : Probabilité d'initiale, probabilité que l'état de départ de modèle soit $s_i$.

$\lambda$, $\hat{\lambda}$, $\dot{\lambda}$ : Modèle de HMM, modèle ré-estimé, modèle optimal.

$P(O/\lambda)$ : Probabilité de l'observation sachant le modèle $\lambda$.

$\alpha$, $\beta$ : Fonction forward, backward.

$\gamma$, $\xi$ : Formules de ré-estimation de Baum-welch.

$\hat{\pi}_i$, $\hat{a}_{ij}$, $\hat{b}_j$ : Paramètres ré-estimés d'un HMM.

$\delta_t(i)$ : Probabilité du meilleur chemin amenant à l'état $s_i$ à l'instant $t$.

$\psi_t$ : Tableau pour garder la trace de meilleur chemin.

$P^*$ : Probabilité du chemin obtenu.

$q_T^*$ : Dernier état du chemin.

$g$ : Fonction de densité de probabilité de la distribution gaussienne.

$c_{jk}$ : Poids de la k$^{ième}$ gaussienne de l'état $s_j$.

$\mu_{jk}$ : Moyenne de la kième gaussienne.

$\Sigma_{jk}$ : Matrice de variance-covariance de la k$^{ième}$ gaussienne.

$d$ : Dimension du vecteur $O_t$.

$Tr$ : Transposée d'une matrice.

$\theta$, $\overline{\theta}$ : Observation directionnelle, direction moyenne circulaire.

$F(\theta)$ : Distribution de pixels noirs selon la direction $\theta$.

$\overrightarrow{OP_i}$ : Vecteur unitaire.

$S_\theta$ : Variance d'une direction moyenne.

# INTRODUCTION GÉNÉRALE

Depuis son invention il y a plus de 5300 ans, l'écriture reste un moyen de communication privilégié entre les êtres humains. Bien que l'imprimerie créée il y a plus de 550 ans puis l'informatique aient permis son automatisation, l'écriture manuscrite est loin d'avoir disparu de notre société et les individus émettent et reçoivent une grande quantité de documents manuscrits. Le traitement de masse de ces documents apparaît alors incontournable. C'est ainsi que les recherches concernant la lecture automatique des documents manuscrits ont débuté il y a plus de 55 ans. Les premiers OCR (Optical Character Recognition) ont fait leur apparition dans les années 1950 et les premiers systèmes de lecture d'adresses postales utilisés pour le tri du courrier sont apparus en 1965. La lecture automatique de l'écriture manuscrite dans les formulaires a fait son apparition dans les années 1980 grâce au développement de moyens de calculs toujours plus puissants.

Malgré les efforts et progrès réalisés dans le domaine, on est encore loin du rêve d'un monde sans papier. Hormis pour des applications très précises telles que la lecture automatique de montants de chèques bancaires, d'adresses postales ou de formulaires, étude du patrimoine et sécurisation des documents, etc., la reconnaissance de l'écriture manuscrite reste un problème complexe et non résolu dans des cas de reconnaissance de documents manuscrits moins contraints.

La lecture automatique de montants numériques de chèques bancaires est une des nombreuses applications de la reconnaissance des caractères manuscrits. La tâche la plus basique dans ce type de système est la reconnaissance des chiffres isolés, cette tache demeure toujours un sujet de recherche très actif. L'effort d'analyse est concentré sur un seul élément à la fois (vue comme une forme globale). Ici, la difficulté du problème de la reconnaissance vient de la variabilité des formes de chiffres lorsqu'on se situe en contexte omni-scripteur, et également de contraintes principales :

*La variabilité de la taille des chiffres* peut se produire parmi les chiffres de même classe. Cela représente un problème difficile pour les méthodes de reconnaissance dans lesquelles il n'y a aucune opération de normalisation. Dans ce cas, une méthode de classification robuste à cette contrainte va s'intervenir pour éviter la déformation des formes au cour de la normalisation.

*L'écriture incomplète (encrage insuffisant)*, ce problème est fréquemment produit dans les chiffres de deux corps, tels que 4 et 5, et les chiffres qui possèdent d'écriture faible ou

incomplète, lors de la binarisation. Les chiffres ayant cette contrainte représentent un problème difficile pour la reconnaissance, dans lesquelles des parties des chiffres cassés doivent être regroupées avant le processus de la reconnaissance.

Les deux contraintes précédentes nous mènent à trouver un classifieur qui est utilisé pour le traitement de problèmes dans lesquels l'information est incertaine et incomplète. Pour résoudre ce problème, Divers travaux dont les modèles de Markov cachés, sont mis en œuvre afin d'aboutir à de meilleures performances et répondre à ces contraintes.

Pour la reconnaissance des chiffres manuscrits, les modèles de Markov cachés (Hidden Markov models : HMMs) ont été utilisés de nombreuses applications. Un HMM est un processus doublement stochastique dont une composante est une chaîne de Markov non observable. Ce processus peut être observé à travers un autre ensemble de processus qui produit une suite d'observations. Les HMMs sont riches en propriétés et fondés sur des bases mathématiques solides. Ces modèles possèdent des techniques d'apprentissage automatique et correctif.

Le HMM est l'une des méthodes de classification les plus populaires pour la reconnaissance d'une séquence de données, particulièrement pour la reconnaissance de la parole et les problèmes de l'écriture manuscrite. Les HMMs modélisent les données à partir de vecteurs des caractéristiques extraites dans les images des caractères.

La recherche sur des méthodes d'extraction de caractéristiques a connu une attention considérable puisqu'un ensemble de caractéristiques discriminantes est considéré comme le module le plus important dans un système performant. En général, les méthodes d'extraction de caractéristiques les plus utilisables pour la reconnaissance de chiffres manuscrits sont regroupées en deux catégories: statistique et structurale.

Les caractéristiques statistiques sont extraites des distributions statistiques des points, tels que zonages, les histogrammes de projection ou les histogrammes de direction. Les caractéristiques structurales sont fondées sur les propriétés topologiques et géométriques du caractère, comme les points de départ et de fin ou des intersections des segments et des boucles. Certaines approches tentent de combiner les deux catégories de caractéristiques pour extraire l'information pertinente.

Plus récemment, une nouvelle approche d'extraction de caractéristique a été développée fondée sur la forme et le fond des caractères. Dans ce travail, nous proposons d'évaluer les performances de deux catégories de caractéristiques, en utilisant les HMMs continus. L'évaluation est réalisée sur une fraction de la base de données standard NIST SD19.

Le travail présenté dans ce mémoire est organisé en quatre chapitres structurés comme suit :

Afin de présenter la méthodologie relative à la reconnaissance de l'écriture manuscrite, nous proposons dans le premier chapitre un panorama des méthodes existantes pour la reconnaissance des caractères manuscrits : mots, séquences numériques et chiffres isolés. Une revue de littérature sur des méthodes d'extraction des caractéristiques et des classificateurs est également présentée. Nous montrons que ces systèmes atteignent désormais des performances satisfaisantes, notamment grâce aux progrès réalisés ces dernières années dans le domaine de la classification statistique et la modélisation de l'écriture manuscrite.

Le deuxième chapitre présente les modèles de Markov cachés utilisés en reconnaissance des chiffres. Nous nous intéressons à la densités d'observations continues par les modèles de mélange de gaussiennes (Gaussian Mixture Model : GMM). Ce type de modélisation Markovienne conduit à résoudre trois types de problèmes, le problème d'évaluation des probabilités d'observation étant donné un modèle, le calcul du chemin optimal et finalement l'estimation du modèle. Le chapitre présente quelques systèmes de reconnaissance basés sur cette technologie. Enfin, une description de la méthode d'apprentissage et de la reconnaissance est présentée pour la reconnaissance des chiffres manuscrits.

Le troisième chapitre est consacré à la réalisation d'un système complet de reconnaissance des chiffres manuscrits dans lequel nous exposons également la démarche suivie. Nous nous concentrons plus particulièrement sur la recherche des méthodes d'extraction des caractéristiques afin d'extraire l'information la plus pertinente. Un classifieur de type HMM continu est utilisé pour modéliser les données.

Dans le dernier chapitre nous présenterons les résultats des différentes expériences réalisées pour le choix de la meilleure configuration pour atteindre les meilleures performances. La mise en œuvre repose sur deux étapes : la première consiste à évaluer plusieurs types de caractéristiques pour la reconnaissance du chiffres, puis une combinaison de ces caractéristiques sera réalisée. Une seconde étape consiste la combinaison des HMMs de colonnes avec HMMs de lignes. Les résultats montrent que le système permet d'obtenir des performances satisfaisantes.

# Chapitre 1:
# Généralité

## 1.1. INTRODUCTION

Le but de la reconnaissance de l'écriture est de transformer un texte écrit en une représentation compréhensible par une machine et facilement reproductible par un logiciel de traitement de texte. Cette tâche n'est pas triviale car les mots possèdent une infinité de représentations parce que chaque personne produit une écriture qui lui est propre, et parce qu'il existe de nombreuses polices de caractères pour l'imprimé avec de nombreux styles (gras, italique, souligné, ombré etc.) et des mises en page différentes et complexes. Suivant le type d'écriture qu'un système doit reconnaître (manuscrit, cursif ou imprimé), les opérations à effectuer et les résultats peuvent varier considérablement.

La reconnaissance de l'écriture manuscrite a connu ces dernières années de grands progrès, et les succès des travaux de recherches ont donné lieu à de nombreuses applications industrielles, notamment dans le domaine de la lecture automatique de formulaires (Ramdane et al, 03; Milewski et Govindaraju, 06), de chèques (Impedovo et al, 97) ou d'adresses postales (El-Yacoubi et al, 02), ainsi que les applications de reconnaissance de l'écriture dites en ligne (Connell et Jain, 02) à travers les PDA, tablet-PC ou stylo caméra. Notre projet s'intéresse principalement à la reconnaissance de chiffres manuscrits hors ligne.

Ce premier chapitre aborde tout d'abord la problématique de la reconnaissance de l'écriture manuscrite en explicitant les facteurs de difficulté pour son traitement, puis détaille les différentes étapes d'un système de reconnaissance de chiffres manuscrits. Les méthodes d'extraction de primitives et de classification sont en particulier étudiées.

## 1.2. RECONNAISSANCE DE L'ECRITURE MANUSCRITE

La lecture automatique des documents manuscrits est actuellement un sujet de recherche très actif et a pour objectif principal de convertir les images de documents manuscrits en fichiers informatiques en vue de l'archivage, la recherche, la modification, la réutilisation et la transmission de l'information que ces images contiennent. Les applications sont aussi nombreuses que variées : tri automatique de courrier, transcription de documents manuscrits anciens, lecture automatique et tri de formulaires, et encore le traitement automatique des chèques bancaires. On distingue communément deux secteurs d'application en reconnaissance de l'écriture manuscrite suivant le mode d'acquisition :

- la reconnaissance d'écriture dite ***en-ligne*** pour laquelle le texte est saisi avec un stylet sur une surface sensible, fournissant ainsi une information temporelle,

➢ la reconnaissance d'écriture dite ***hors-ligne*** pour laquelle aucune information temporelle n'est disponible : la page de texte est scannée afin d'obtenir une image numérique.
Dans cette étude, c'est la reconnaissance de l'écriture hors-ligne qui va nous intéresser.

### 1.2.1. Complexité de la reconnaissance d'écriture manuscrite

On distingue plusieurs degrés de difficulté du traitement de l'écriture manuscrite (figure1.1) suivant plusieurs critères (Poisson, 05) :

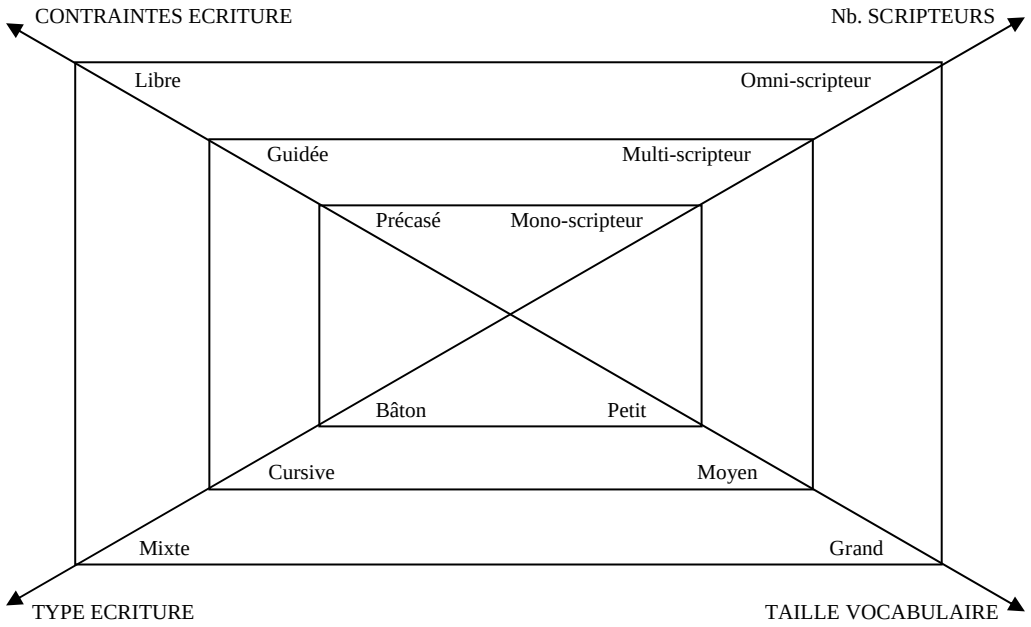



**Figure 1.1.** Difficultés de l'écriture manuscrite (Poisson, 05).

#### 1.2.1.1. Conditionnement de l'information

L'écriture reconnue peut être plus ou moins conditionnée par la présence de précasé (cas des formulaires, code postal d'une adresse) (cas 1 de figure I.2), des cadres (montants de chèques) ou lignes (lignes d'un bloc adresse, montant littéral d'un chèque), ou bien non conditionnée (documents libres).

#### 1.2.1.2. Style d'écriture

Chaque individu écrit d'une façon qui lui est propre, son écriture est différente de celle des autres même si l'alphabet de base est le même. Toute la difficulté de la reconnaissance de l'écriture manuscrite réside en majeure partie dans cette grande variété d'écritures, selon (Tappert et Suen, 90), la difficulté à reconnaître l'écriture augmente avec les styles d'écriture suivants (cas 2 à 5 de la figure 1.2):

- lettres séparées;
- lettres groupées liées, script (écriture bâton) ;
- purement cursive ou mixte.

De plus, d'une personne à l'autre, la forme donnée à chaque caractère diffère,

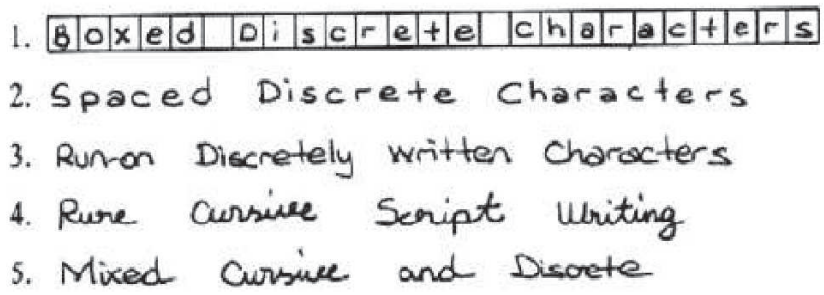



**Figure 1.2.** Style d'écriture selon (Tappert et Suen, 90).

### 1.2.1.3. Taille du vocabulaire

La taille du vocabulaire est également un facteur influant sur les performances d'un système de reconnaissance ainsi que sur la méthode adoptée. En effet, alors que pour des tâches nécessitant un petit vocabulaire (une trentaine de mots dans le cas de la reconnaissance de montants de chèques) on cherchera à reconnaître le mot comme une entité à part entière, pour des tâches de très grand vocabulaire on cherchera plutôt à reconnaître les lettres au sein du mot. Il est habituel de considérer que :

- un petit vocabulaire est constitué d'une dizaine de mots,
- un vocabulaire de taille moyenne comporte une centaine de mots,
- un grand vocabulaire rassemble des milliers de mots,
- enfin, un très grand vocabulaire renvoie à plus d'une dizaine de milliers de mots.

### 1.2.1.4. Nombre de scripteurs

La difficulté de reconnaissance augmente avec le nombre de scripteurs. En effet, plus il y a de scripteurs (droitier/gaucher, enfant/adulte, différentes professions), plus il y a de styles d'écritures différents, divisant l'échelle de difficultés en trois : mono, multi et omni-scripteurs. En multi-scripteur, le système doit s'adapter à l'écriture de plusieurs scripteurs, tandis qu'en omni-scripteur, le système doit être capable de généraliser son apprentissage à n'importe quel type d'écriture.

## 1.3. RECONNAISSANCE DE MOTS

Il existe deux stratégies possibles pour la reconnaissance des mots manuscrits, les approches globales qui considèrent le mot dans son ensemble sans chercher à identifier chacune des lettres qui le compose ; qu'on oppose aux approches analytiques qui cherchent à découper le mot en lettres afin de le reconnaître (Chatelain, 06).

### 1.3.1. Approche Globale

Dans les approches globales, des caractéristiques sont extraites sur le mot entier afin de calculer une distance à des modèles de mots (Koerich et al, 03). Ces approches présentent l'inconvénient de subir la variabilité des mots, plus importante encore que celle observée sur les lettres. Ainsi, elles requièrent des bases de mots conséquentes. Elles sont, de plus, peu discriminantes pour des mots différents dont la forme est proche, ce qui les limite à des applications à lexique réduit (cas des montants littérales de chèques (Knerr et al, 97)).

### 1.3.2. Approche Analytiques

Les approches analytiques visent à reconnaître les mots en identifiant les lettres qui le composent. Une étape de segmentation est donc nécessaire afin de déterminer les limites entre les lettres. Cette tâche est particulièrement délicate du fait de l'absence de segmentation idéale : les limites entre caractères sont parfois difficiles à déterminer même pour un être humain. Il existe deux types d'approches analytiques : une segmentation explicite ou implicite.

- ***Segmentation explicite :*** Les approches à segmentation explicite utilisent des algorithmes de segmentation généralement basés sur les contours, les histogrammes de projection ou les profils (Koch et al, 04; El-Yacoubi et al, 99; Kimura et al, 94) pour proposer des hypothèses de points de segmentation afin de séparer les caractères des uns des autres.

- ***Segmentation implicite :*** Les approches à segmentation implicite (Mohammed et Gader, 96; Senior et Robinson, 98 ; Morita et al, 03) considèrent tous les points du tracé comme des points de segmentation potentiels à l'aide d'une fenêtre glissante successivement décalée de un à quelques pixels. Des caractéristiques sont extraites de chaque fenêtre et sont soumises à un classificateur dynamique (Modèle de Markov Caché, réseaux de neurones à convolution, réseaux récurrents) afin de prendre une décision globale de segmentation et de reconnaissance sur l'ensemble du mot.

## 1.4. RECONNAISSANCE DE SEQUENCES NUMERIQUES

La reconnaissance de séquences numériques est certainement l'un des domaines les plus abouti en reconnaissance d'écriture manuscrite. Elle consiste à reconnaître tous les chiffres et éventuellement identifier les entités non numériques (séparateur, symbole, virgule, point, etc.) d'une séquence numérique déjà localisée. Les deux applications les plus connues sont la reconnaissance de montants numériques de chèques bancaires (Knerr et al, 97 ; Zhang et Suen, 02) et la reconnaissance de code postal dans les adresses (Liu et al, 02 ; Pfister et al, 00). Mais certains travaux ont également concerné la reconnaissance de dates sur les chèques (Morita et al, 02 ; Morita et al, 03 ; Xu et al, 03).

Toutes les approches de la littérature, pour faire la reconnaissance, procèdent donc à une localisation des chiffres par un processus de segmentation. Comme le cas des mots, la segmentation peut être implicite si tous les points du tracé sont susceptibles d'être choisis comme point de segmentation, ou explicite si un algorithme de segmentation sélectionne des points candidats à la segmentation (Chatelain, 06).

### 1.4.1. Segmentation Explicite

Cette approche consiste à trouver le meilleur chemin de séparation des chiffres des uns des autres, Les chemins de segmentation sont généralement obtenus par des points caractéristiques issus d'une analyse des contours de la forme, du squelette ou d'un amincissement du fond (Sadri et al, 04), d'une analyse en deux dimensions du tracé, ou d'une combinaison analyse des contours/squelette (Lei et al, 04 ; Liu et al, 04 ; Oliveira et al, 02 ; Oliveira , 03 ; Gattal, 09).

La figure 1.3 illustre l'architecture d'un système de reconnaissance de séquence numérique basé segmentation explicite.

### 1.4.2. Segmentation Implicite

La reconnaissance de chaîne numérique basée segmentation implicite consiste à concevoir un système capable d'intégrer, à la fois, la segmentation et la reconnaissance. La segmentation et la reconnaissance sont réalisées conjointement, d'où le nom parfois employé de (segmentation reconnaissance intégrée). Il s'agit de méthodes à fenêtres glissantes de taille fixe ou variable qui parcourent la séquence de chiffres (figure 1.4), chaque fenêtre est décrite par un vecteur caractéristique. L'analyse des fenêtres est effectuée soit par un classifieur classique, soit par des modèles dynamiques tels que les modèles de Markov cachés (HMM) ou les réseaux de neurones à convolution (TDNN), qui déterminent la classe d'appartenance de chaque fenêtre en fonction des fenêtres voisines (Cavalin et al, 06 ; Britto et al, 00 ; Britto et al, 03).

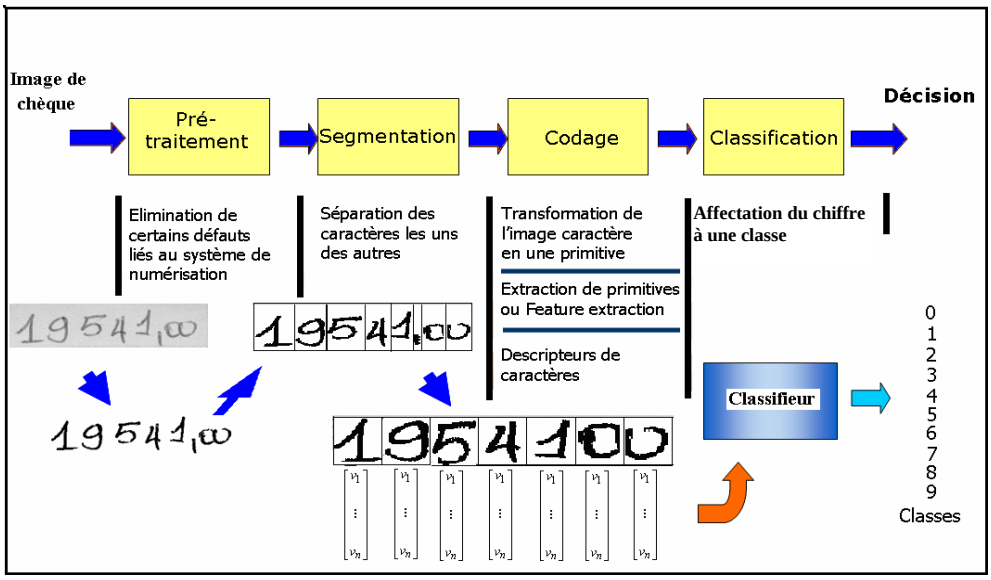



**Figure1.3.** Architecture d'un système de reconnaissance de chaîne numérique basé segmentation explicite (Gattal, 09).

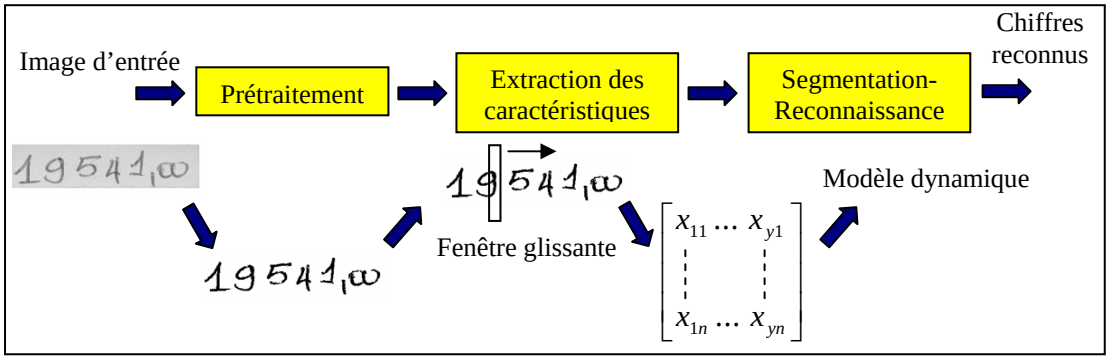



**Figure1.4.** Architecture d'un système de reconnaissance de chaîne numérique basé segmentation implicite.

## 1.5. RECONNAISSANCE DE CHIFFRES ISOLES

C'est la tâche la plus basique d'un système de reconnaissance de chaîne numérique. L'effort d'analyse est concentré sur un seul élément à la fois (vue comme une forme globale). Les méthodes de reconnaissance sont nombreuses, dépendant du choix du type des indices visuels (ou paramètres, ou encore primitives) extraits de la forme du chiffre. Ces paramètres peuvent être soit topologiques (éléments ou parties), soit géométriques ou métriques (de type distance, taille, courbure et angle), soit enfin statistiques relatives à des observations de points (de type présence, absence, agglomération et distribution).

La difficulté du problème de la reconnaissance vient de la variabilité des formes de chiffres lorsqu'on se situe en contexte omni-scripteur. On peut constater ces différences sur la figure 1.5

en considérant les trois classes de chiffres : remarquons les variations de taille, de structure, d'inclinaison et de trait au sein d'une même classe.

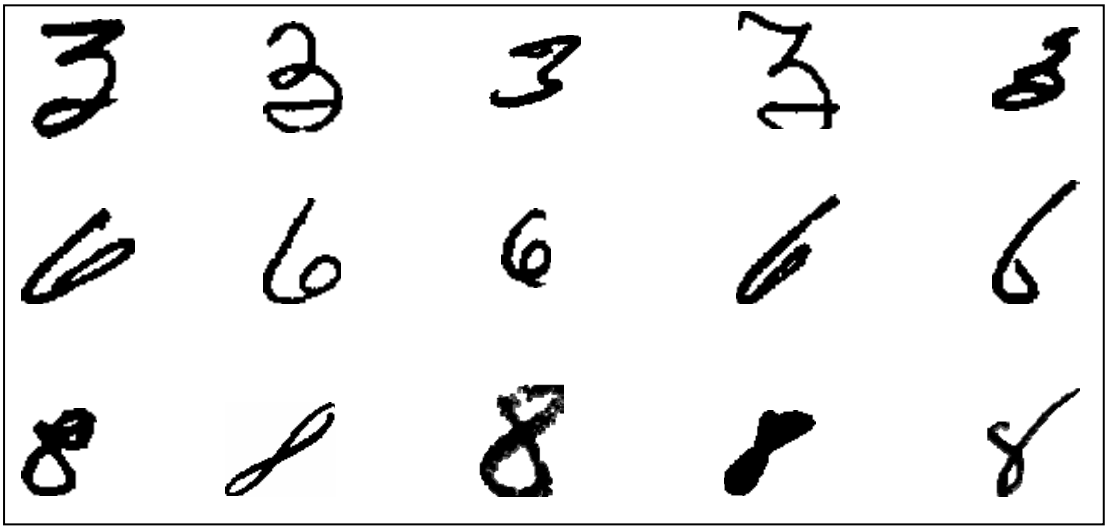


**Figure1.5.** Exemples de prototypes 3, 6 et 8 extraits à partir de la base NIST SD19.

Un système de reconnaissance du chiffre isolé est composé de trois étapes principales : le prétraitement, l'extraction des caractéristiques et la prise de discision (classification).

### 1.5.1. Prétraitement

Le prétraitement est une étape facultative, on trouve plus ou moins des prétraitements, parmi les quelles : la binarisation, le débruitage, lissage et la correction d'inclinaison. Toutes ces prétraitement ne sont pas systématiquement mises en ouvre.

### 1.5.2. Extraction des Caractéristiques

Cette étape est une des clefs d'un système efficace de reconnaissance de caractères. En effet, l'utilisation d'un classifieur très performant ne peut compenser une représentation mal adaptée ou peu discriminante.

On peut définir l'extraction de caractéristiques comme le 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 inter-classe* (Devijver et Kittler, 82), cette information pertinente prend souvent la forme d'un vecteur de valeurs numériques.

Les types de caractéristiques peuvent être classés en quatre catégories principales de primitives: structurelles, statistiques, globales (transformations globales) et topologiques ou métriques (Dargenton, 94 ; Al-Badr et Mahmoud, 95).

#### 1.5.2.1. *Primitives structurelles*

Les primitives structurelles (ou primitives locales) basées sur une représentation linéaire du caractère (décomposition du caractère en segments de droites et courbes, contours du caractère, squelette). Les primitives structurelles sont généralement extraites non pas de l'image brute, mais à partir d'une représentation de la forme par le squelette ou par le contour. Ainsi, parmi ces caractéristiques, il s'agit principalement des segments de droite, des arcs, boucles et concavités, des pentes, la hauteur et la largeur du caractère...etc.

#### 1.5.2.2. *Primitives statistiques*

Elles portent une information sur la distribution des pixels dans l'image du caractère ou chiffres, Les primitives statistiques décrivent une forme en termes d'un ensemble de mesures extraites à partir de cette forme. Les caractéristiques utilisées pour la reconnaissance des chiffres manuscrits sont :

- *Le zonage (zoning) :* consiste à partager l'image en région ou zones. L'image de départ est une image binaire. Pour chacune des régions résultantes, il s'agit de calculer la moyenne ou le pourcentage de points noirs ;
- *L'histogramme* représentant le nombre de pixels sur chaque ligne ou colonne de l'image, en est un exemple classique et simple à calculer histogramme directionnel, histogramme des transitions …).

Il est possible également d'utiliser aussi une autre primitive statistique basée sur un moyennage des pixels situés à l'intérieur d'un masque rectangulaire. La construction d'une matrice de masque recouvrant la totalité de la forme permet une représentation statistique à partir d'un nombre très réduit de valeurs correspondant à chaque masque

#### 1.5.2.3. *Primitives globales*

Les primitives globales sont naturellement basées sur une transformation globale de l'image. La caractéristique d'une primitive globale est de dépendre de la totalité des pixels d'une image, ces primitives sont donc dérivées de la distribution des pixels. Ils dirigent trois familles de caractéristiques telles que : les moments invariants, les projections et les profils.

Une des transformations les plus simples est celle qui représente le squelette ou le contour d'un caractère sous forme d'une chaîne de codes de directions (code de Freeman). Parmi les méthodes de transformation, nous pouvons citer les transformées de Hough, de Fourier et d'ondellettes.

#### 1.5.2.4. *Primitives topologiques ou métriques*

La topologie est l'étude des propriétés de l'espace (et des ensembles) du seul point qualitatif. Il s'agit d'effectuer une mesure sur l'échantillon au moyen d'une métrique. Parmi les mesures, nous pouvons citer :

- compter le nombre de trous,
- évaluer les concavités,
- mesurer des pentes, des courbures et évaluer des orientations principales,
- mesurer la longueur et l'épaisseur des traits,
- détecter les croisements et les jonctions des traits,
- mesurer les surfaces, les périmètres,
- déterminer le rectangle délimitant l'échantillon, ou le polygone convexe,
- évaluer le rapport d'élongation (ou allongement) longueur/largeur, ...,
- rendre compte de la disposition relative de ces primitives.

Toutes ces mesures peuvent être intégré dans un seul vecteur de caractéristiques pour la reconnaissance du chiffre manuscrit.

Une combinaison de différentes familles de caractéristiques est souvent mise en œuvre afin d'obtenir plusieurs représentations d'un même forme et d'améliorer la discrimination (Xue et Govindaraju, 06). Lorsque le nombre de caractéristiques devient trop élevé, des méthodes de sélection de caractéristiques peuvent être mises en ouvre (Oliveira et al, 06).

### 1.5.3. Classification

Le rôle du classifieur est de se prononcer sur l'appartenance d'une forme à chacune des classes de caractère à partir du vecteur de caractéristiques par un critère de décision.

Le terme classification regroupe en fait trois approches qui impliquent des stratégies différentes :

- *classification supervisée* : cette technique est basée sur l'étiquetage des observations en affectant chaque observation à une classe.
- *classification semi supervisée* : cette technique est basée sur l'étiquetage d'une partie des observations.
- *classification non supervisée* : aucune des observations n'est étiquetée.

Il est sûr que la complexité de la tâche est croissante avec le nombre de données non étiquetées. Dans le cas de la reconnaissance des chiffres manuscrits, la classification supervisée a été adoptée car elle simplifie considérablement le problème. Elle se déroule en deux étapes :

- *étape d'apprentissage:* elle consiste à apprendre au classifieur des formes connues pour caractériser les classes.
- *étape de Reconnaissance et décision :* elle cherche parmi les modèles de référence en présence, ceux qui lui sont les plus proches. Le problème revient à affecter une forme inconnue à l'une des classes obtenues pendant l'apprentissage (Al-Badr et Mahmoud, 95). Le résultat de la décision est un « avis » sur l'appartenance ou non de la forme aux modèles de l'apprentissage.

La reconnaissance peut conduire à :

- Un succès si la réponse est unique et correcte (un seul modèle répond à la description de la forme du caractère).
- Une confusion si la réponse est multiple (plusieurs modèles correspondent à la description).
- Un rejet de la forme si aucun modèle ne correspond à sa description.

Dans les deux premiers cas, la décision peut être accompagnée d'une mesure de vraisemblance, appelée aussi score ou taux de reconnaissance.

Il existe de nombreux classifieurs possédant des caractéristiques de performances et de vitesse différentes. Selon (Jain et al, 00), il existe quatre grandes familles de classifieurs : appariement de formes, les approches structurelles, les réseaux de neurones et la classification statistique.

#### 1.5.3.1. Mise en correspondance

La mise en correspondance ou appariement de forme (en Anglo-saxon : Template Matching) vise à comparer une forme à des représentants de chaque classe via une mesure de distance (corrélation ou similarité). Elle est peu adaptée à la reconnaissance de manuscrit car la très forte variabilité des caractères manuscrits implique un nombre très important de représentants pour chaque classe.

#### 1.5.3.2. Techniques structurelles

Elle est fondée sur la structure propre du caractère, privilégiant certains aspects dans le tracé même des lettres. La structure est exprimée en terme de composants primitifs correspondant à des formes élémentaires du tracé et à des événements produits lors du tracé comme un rebroussement, un changement d'orientation, un croisement, un accroissement ou un décroissement de la pente, etc. ces composants sont appelés *primitives*.

Dans cette technique, on distingue plusieurs méthodes telles que les structures de graphes, les structures syntaxiques, les méthodes de tests, la comparaison de chaînes,…etc.

#### 1.5.3.3. Approches connexionnistes

Le problème de reconnaissance de formes peut être résolu en trouvant une fonction, qui associe un ensemble de formes de départ à un ensemble de classes d'arrivée, donc le réseau connexionniste jouera le rôle d'une fonction de transfert de telle sorte que si on représente une valeur en entrée et on applique cette fonction elle nous fournit la valeur de sortie, alors on espère que cette dernière corresponde à la sortie d'appartenance de la valeur d'entrée.

Les réseaux connexionnistes modélisent les surfaces de séparation entre les classes, ce qui va diminuer l'erreur de classification, pour cela il prend en compte la structure interclasse des données. Les techniques connexionnistes sont très efficaces pour des tâches très difficiles de reconnaissance de formes. Cette approche est très coûteuse en temps pendant l'apprentissage et difficile de donner exactement l'architecture du réseau.

Les réseaux de neurones (Neural Network : NN) ont connu un grand succès à partir des années 90, notamment grâce à la mise au point d'un algorithme d'apprentissage efficace et facile à mettre en ouvre : la rétro-propagation du gradient (Zhang, 00). Il existe de nombreux type de réseaux de neurones mais les deux types les plus utilisés en reconnaissance de caractères sont les perceptrons multicouches (Tay et al, 03) et les réseaux à fonctions de base radiales (Augustin, 01 ; Gilloux et al, 95).

#### 1.5.3.4. Approches statistiques

La reconnaissance est fondée sur l'étude statistique des mesures que l'on effectue sur les formes à reconnaître. Elle a besoin d'un nombre élevé d'exemples afin de réaliser un apprentissage correct des lois de probabilité des différentes classes. L'étude de leur répartition dans un espace métrique et la caractérisation statistique des classes permet de prendre une décision du type : plus forte probabilité d'appartenance à une classe.

La phase d'apprentissage supervisé consiste à déterminer les règles de décision à partir des exemples de la base d'apprentissage. Pour un ensemble d'apprentissage donné, on peut construire les frontières de décision de deux manières différentes. La première solution consiste à générer les frontières implicitement à partir des distributions de probabilité de chaque classe (***approches modélisantes*** : mixture de gaussiennes, Modèle de Markov caché, K plus proches voisins). Le deuxième type d'approche consiste à estimer explicitement les frontières de décision entre les classes (***approches discriminantes***).

**1.5.3.4.1. Approches discriminantes**: elles visent à construire directement des frontières de décision par la minimisation d'un critère d'erreur entre les sorties réelles et escomptées du classifieur. Le critère d'erreur choisi est souvent le taux de classification ou l'erreur quadratique. Il existe plusieurs classifieurs discriminants, et le plus récemment, les machines à vecteurs de support (Vapnik, 98). Les machines à vecteurs de support (ils sont aussi appelées : Classifieurs à Marge Optimale ou encore Séparateur à Vaste Marge) ont connu de nombreuses applications en reconnaissance de caractères ces dernières années (Nemmour et Chibani, 06; Ayat, 05). Considérées comme les classifieurs possédant les meilleures capacités de généralisation.

**1.5.3.4.2. Approches modélisantes** : elles construisent les frontières de décision à partir des distributions de probabilités de chaque classe. Ces approches bénéficient des méthodes d'apprentissage automatique qui s'appuient sur des bases théoriques connues telles que :

**1.5.3.4.2.1.** ***Méthodes non paramétriques*** *:* elles sont mises en ouvre dans le cas où l'on ne dispose pas de connaissances *a priori* sur la distribution de probabilité des classes. L'approche non paramétrique la plus connue est le k plus proche voisin (K-PPV ou K-NN).

**Méthode par k plus proches voisins** est une méthode simple qui ne nécessite pas d'apprentissage. La modélisation utilisée pour la classification est la base d'apprentissage qui constitue l'ensemble des connaissances du système de reconnaissances. Le principe est d'attribuer à la donnée en entrée du reconnaisseur la classe la plus représentée dans son plus proche voisinage. Il faut donc déterminer le nombre *k* d'individus considérés et la métrique utilisée qui peut être une distance élastique pour comparer les individus et sélectionner les plus proches voisins. L'efficacité de cette méthode dépend de la qualité de sa base d'apprentissage et de la capacité de stockage de celle-ci. Elle est donc limitée à une reconnaissance globale avec un vocabulaire restreint pour ne pas alourdir les temps de traitement et les coûts de stockage. Elle est donc particulièrement adaptée à une reconnaissance mono-scripteur, l'utilisateur devra alors, avant d'utiliser le système, procéder à la saisie de la base d'apprentissage (Poisson, 05).

**1.5.3.4.2.2.** ***Méthodes paramétriques*** : dans les méthodes paramétriques, on considère que la forme des distributions de probabilité est connue, des gaussiennes dans le cas classique. Lors de l'apprentissage, on cherche donc à estimer les paramètres inconnus des gaussiennes pour chaque classe : moyennes et variances, ou éventuellement matrices de covariances pour chaque classe. Une fois ces paramètres estimés, la décision se fait naturellement par la règle de Bayes. L'approcha paramétrique la plus connue est les Modèles de Markov Cachés.

**Modèles de Markov Cachés :** Un modèle de Markov représente le caractère par un double processus stochastique: la séquence de primitives (symboles) appelée observations $O_1.....O_T$, et la séquence d'états $S_1.....S_N$. Dans la modélisation Markovienne, une suite d'observations est modélisée par une suite d'états avec des probabilités d'émission. La suite d'états est conditionnée par des probabilités de transitions. A chaque état est associé une probabilité d'émission de symboles observables $O_1.....O_T$. Ces états ne sont pas directement observables, c'est pourquoi, ils sont dits cachés, d'où le terme de Modèles de Markov Cachés. La comparaison consiste à chercher dans ce graphe d'état, le chemin de probabilité maximale correspondant à une suite d'éléments observés dans la chaîne d'entrée (Rabiner, 89). Cette approche sera développée plus en détail par la suite (chapitre 2).

## 1.6. CONCLUSION

Nous avons dressé dans cette étude bibliographique un panorama des systèmes de reconnaissance d'entités manuscrites isolées. Nous avons pu constater que grâce aux progrès dans le domaine de la classification statistique et la modélisation de séquences, les systèmes actuels offrent désormais des performances intéressantes pour la reconnaissance de caractères isolés.

Dans cette étude, ce sont les chiffres isolés qui vont nous intéresser. Et pour éviter l'étape de la normalisation des chiffres, notre système va se baser sur la technique de la fenêtre glissante. Cette approche utilise essentiellement les modèles de Markov cachés adéquats par leurs structures à modéliser des données séquentielles et leur capacité à modéliser justement les transitions et les variabilités de longueur des formes.

Nous reprendrons, plus loin, cette méthode et nous allons l'illustrer de manière plus fine sur la reconnaissance de chiffres isolés.

# Chapitre 2:
# Modèles de Markov Cachés

## 2.1. INTRODUCTION

Les modèles de Markov cachés (Hidden Markov Models ou HMMs) ont été introduits par Baum et ses collaborateurs dans les années 1960-70 (Rabiner, 89). Ces modèles sont fortement apparentés aux processus aléatoires. L'abréviation anglaise **HMMs** sera couramment utilisée dans la suite pour parler des modèles de Markov cachés. Les HMMs connaissent actuellement un net regain d'intérêt tant dans ses aspects théoriques qu'appliqués. Actuellement, ils sont largement appliqués dans les domaines de la reconnaissance de la parole, la reconnaissance de formes, le traitement d'images et des signaux, la robotique, ... Ces applications nécessitent une convergence d'approches du type compréhension, intelligence artificielle, programmation dynamique et celles du type traitement du signal, méthodes statistiques.

Les HMMs connaissent actuellement un essor important en reconnaissance des formes grâce à leur capacité d'intégration du contexte et d'absorption du bruit. Dans ces modèles, les formes sont décrites par une séquence de primitives qui seront observées dans les états du modèle. La probabilité d'émission de la forme par le modèle est calculée en maximisant, sur l'ensemble des chemins d'états, la probabilité d'observation des segments pondérée par les probabilités de transitions entre états. Ce calcul se fait généralement par maximum de vraisemblance. Le calcul de la vraisemblance de la forme par rapport au modèle est fondé sur la règle de Bayes qui inclut la probabilité à priori du modèle.

Ce chapitre sera basé sur les articles de Rabiner et ses collègues (Rabiner, 89, Juang et Rabiner, 91).

## 2.2. DEFINITIONS

### 2.2.1. Modèle Stochastique

Un modèle stochastique est défini comme un processus aléatoire qui peut changer d'état $s_i$, $i = 1,...,N$ au hasard aux instants $t$=1,2,...,$T$. A chaque état, il possède une variable aléatoire $X(t) = s_i, t = 1,...,T$ qui prend ses valeurs dans l'ensemble des observations que l'on fait dans le temps du phénomène analysé. Une évolution du système est donc une suite de transitions d'états à partir d'un état de départ $X(1) = s_i$ :

$$s_1 \rightarrow s_2 \rightarrow s_3 \ldots \rightarrow s_T .$$

La loi d'évolution du système consiste à avoir les chaînes de transitions $s_1, s_2, ..., s_m \quad \forall m \leq T$ avec la probabilité $P(s_1, s_2, ..., s_T)$. Cette probabilité est définie de proche en proche comme suit :

$$\begin{aligned} P(s_1, s_2, ..., s_T) &= P(s_1, s_2, ..., s_{T-1}) P(s_T \mid s_1, s_2, ..., s_{T-1}) \\ &= P(s_1, s_2, ..., s_{T-2}) P(s_{T-1} \mid s_1, s_2, ..., s_{T-2}) P(s_T \mid s_1, s_2, ..., s_{T-1}) \\ &= P(s_1).P(s_2 \mid s_1) P(s_3 \mid s_2, s_1) ... P(s_T \mid s_1, s_2, ..., s_{T-1}). \end{aligned} \tag{2.1}$$

Ainsi, pour calculer la probabilité de toute la chaîne de transition, il suffit de se donner la probabilité initiale $P(s_1)$ et les probabilités des états conditionnés par les évolutions antérieures. La situation générale est celle où la loi de probabilité des états à un certain instant dépend de la totalité de l'histoire du système. On dira que celui-ci garde la mémoire de son passé.

On dit qu'un processus aléatoire est un processus de Markov du premier ordre s'il vérifie la propriété de Markov :

$$P(X(t) = s_i \mid X(t-1) = s_j, X(t-2) = s_k \in ) = P(X(t) = s_i \mid X(t-1) = s_j) \tag{2.2}$$

Cette relation signifie qu'un système qui évolue en suivant la propriété de Markov est dépourvu de mémoire : toutes les informations utiles à la prévision de l'évolution future d'un tel système contenues dans son évolution passée sont condensées dans la connaissance de l'état du système à l'instant actuel. Une fois qu'on sait que le système est dans un certain état à un certain instant, peu importe la manière dont cet état a été atteint.

On remplace par la suite $X(t)$ par $q_t$.

### 2.2.2. Modèle de Markov Caché

Un HMM est un automate stochastique capable, après apprentissage, d'estimer la probabilité qu'une séquence d'observations ait été générée par ce modèle. Idéalement, il faudrait pouvoir associer à chaque classe un modèle constitué d'un nombre fini d'état prédéterminé.

Un HMM représente une forme par deux suites de variables aléatoires : l'une observable et l'autre cachée. La suite observable $O = O_1, O_2, ..., O_T$, où $O_1, ..., O_T$ sont aussi fonction du temps et se réalise parmi un ensemble de *M* symboles observables. La suite cachée correspond à la suite

d'états $q_1, q_2, ..., q_T$ notée $Q(1:T)$, où les $q_i$ prennent leur valeur parmi l'ensemble des *N* états du modèle $\{s_1, s_2, ..., s_N\}$. C'est pourquoi ces modèles sont dits 'cachés'. La séquence d'états n'est pas directement observable. Formellement, un modèle de Markov caché peut être défini par l'ensemble des paramètres suivants:

- un ensemble d'états représentés par $S = \{s_1, s_2, \ldots, s_N\}$ avec $N$ est le nombre d'état, $q_t$=$s_i$ est l'état du système à l'instant $t$ tel que $1 \le i \le N$ ;
- un ensemble de symboles d'observation représentés par $V = \{v_1, v_2, ..., v_M\}$ avec $M$ le nombre de symboles, $O_t$ est l'observation à l'instant $t$ tel que $t \in [1, ..., T]$, $T$ désigne le nombre d'observation (trame);
- La matrice de transition de probabilité, pour la chaîne de Markov : $A_{[N \times N]} = (a_{ij})$ $a_{ij}$ représente la probabilité que le modèle évolue de l'état $s_i$ vers l'état $s_j$. Soit dans le cas général :

$$a_{ij} = P(q_{t+1} = s_j \mid q_t = s_i) \quad avec \begin{cases} 1 \le i, j \le N \quad et \quad 1 \le t \le T \\ 0 \le a_{ij} \le 1 \quad et \quad \sum_{j=1}^{N} a_{ij} = 1 \end{cases} \tag{2.3}$$

- La matrice d'émission de probabilité d'observations des *M* symboles pour chaque état du modèle : $B_{[N \times M]} = (b_j(k))$ où $b_j(k)$ représente la probabilité d'observer $v_k$ à l'état $s_j$, tel que :

$$b_j(k) = P(O_t = v_k \mid q_t = s_j) \quad avec \begin{cases} 1 \le j \le N \quad et \quad 1 \le k \le M \\ \sum_{k=1}^{M} b_j(k) = 1 \end{cases} \tag{2.4}$$

- Un ensemble de densités de probabilités initiales $\Pi = \{\pi_i\}$ ; $i = 1, 2, \ldots, N$ où $\pi_i$ représente la probabilité initiale du modèle à l'état $s_i$, soit :

$$\pi_i = P(q_1 = s_i) \quad avec \begin{cases} 1 \le i \le N \\ 0 \le \pi_i \le 1 \quad avec \quad \sum_{i=1}^{N} \pi_i = 1 \end{cases} \tag{2.5}$$

## 2.3. TROIS PROBLEMES FONDAMENTAUX D'UN HMM :

L'article de Rabiner (Rabiner, 89) identifie trois problèmes pour rendre ce formalisme des HMMs utile :

- Problème 1 : **Evaluation du modèle**

  Etant donné une suite d'observations $O = O_1O_2...O_T$, et un modèle $\lambda = (A,B,\Pi)$, comment peut-on calculer efficacement $P(O/\lambda)$, la probabilité de la suite d'observation, sachant le modèle?

- Problème 2 : **Apprentissage**

  Comment peut-on ajuster les paramètres du modèle $\lambda = (A,B,\Pi)$ pour maximiser $P(O/\lambda)$ ?

- Problème 3 : **Estimation de la séquence d'états optimale**

  Etant donné une suite d'observations (visibles) $O = O_1O_2...O_T$, et un modèle $\lambda = (A,B,\Pi)$, comment peut-on trouver le chemin le plus vraisemblable, c'est-à-dire la séquence d'états (non visibles) la plus probable pour produire cette séquence d'observables ?

### 2.3.1. Evaluation du modèle :

Il existe plusieurs méthodes qui calculent la probabilité d'une suite d'observation $O$ étant donné le modèle $\lambda$ : $P(O|\ \lambda)$. Parmi ces méthodes, on trouve:

❖ **Evaluation directe :**

$P(O|\ \lambda)$ est égale à la somme de tous les chemins d'états possibles $Q = \{q_1, q_2, ..., q_T\}$ des probabilités conjointes de $O$ et de $Q$ :

$$P(O|\ \lambda) = \sum_Q P(O,Q|\ \lambda) = \sum_Q P(O|\ Q,\lambda).P(Q|\ \lambda) \tag{2.6}$$

Or $$P(Q|\ \lambda) = \pi_{q_1}.a_{q_1q_2}.a_{q_2q_3}...a_{q_{T-1}q_T} \tag{2.7}$$

Pour calculer $P(O|\ Q,\lambda)$, on suppose l'indépendance statistique des observations, on obtient :

$$P(O|\ Q,\lambda) = b_{q_1}(O_1).\ b_{q_2}(O_2)...\ b_{q_T}(O_T) \tag{2.8}$$

On déduit donc la relation :

$$P(O|\ \lambda) = \sum_Q \{\pi_{q_1}\ b_{q_1}(O_1)\}\{a_{q_1q_2}\ \pi_{q_2} b_{q_2}(O_2)\}...\{a_{q_{T-1}q_T}\ \pi_{q_T}\ b_{q_T}(O_T)\} \tag{2.9}$$

Il est à noter que cette formule nécessite ($N^T$-1) additions et (($2T$-1)$N^T$) multiplications ($N^T$ étant le nombre de chemins possibles de longueur $T$). L'algorithme qui calcule cette probabilité de façon directe a une complexité de ($2T.N^T$) (ordre exponentiel) ce qui est considérablement lourd (coûteux en temps de calcul). Il existe heureusement deux algorithmes rapides (de type programmation dynamique). Le premier est une procédure d'estimation directe (algorithme forward). Le deuxième est la procédure d'estimation rétrograde (algorithme de backward). Ces deux algorithmes permettent un calcul moins complexe.

### ❖ Evaluation en utilisant les fonctions Forward Backward

Dans cette approche, on considère que l'observation se fait en deux étapes :

1- Emission du début de l'observation $O_1, \ldots, O_t$ en aboutissant à l'état $q_i$ à l'instant $t$.

2- Emission de la fin de l'observation $O_{t+1}, \ldots, O_T$ sachant qu'on part de l'état $q_i$ à l'instant $t$.

Dans ce cas, la probabilité de l'observation générer par le modèle est :

$$P(O|\ \lambda) = \sum_i \alpha(t, q_i).\beta(t, q_i) \tag{2.10}$$

Où :

- $\alpha(t,q_i)$ est la probabilité d'émettre le début de l'observation $O(1:t)$ pour aboutir à l'état $q_i$ à l'instant $t$. Le calcul de $\alpha$ se fait avec $t$ croissant (d'où le terme Forward).
- $\beta(t,q_i)$ est la probabilité d'émettre la fin de l'observation $O(t+1:T)$ sachant qu'on part de $q_i$ à l'instant $t$. Le calcul de $\beta$ se fait avec $t$ décroissant (d'où le terme Backward).

On notera par la suite $\alpha(t,q_i)$ par $\alpha_t(i)$ et $\beta(t,q_i)$ par $\beta_t(i)$.

On a alors :

$$\alpha_t(i) = P(O(1:t), q_t = s_i|\ \lambda). \tag{2.11}$$

$$\beta_t(i) = P(O(t+1:T), q_t = s_i|\ \lambda). \tag{2.12}$$

Les $\alpha_t(i)$ seront calculés de manière récursive comme suit :

- Initialisation : (début de l'algorithme)

$$\alpha_0(i) = \pi_i \quad et \quad \alpha_1(i) = \alpha_0(i).b_i(O_1) \qquad 1 \le i \le N .$$

- Induction : (calcul intermédiaire)

$$\alpha_t(j) = b_j(O_t).\sum_{i=1}^{N}\{\alpha_{t-1}(i).a_{ij}\} \qquad 1 < t \le T \quad et \quad 1 \le j \le N .$$

- Terminaison : (fin de l'algorithme)

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

Les $\beta_t(i)$ seront calculés de manière récursive comme suit :

- Initialisation : (début de l'algorithme)

$$\beta_T(i) = 1 \qquad 1 \le i \le N .$$

- Induction : (calcul intermédiaire)

$$\beta_t(i) = \sum_{j=1}^{N}\{a_{ij}.b_j(O_{t+1}).\beta_{t+1}(j)\} \quad 1 \le i \le N \ et \ pour \ t = T-1, T-2, ..., 1 .$$

- Terminaison : (fin de l'algorithme)

$$\beta_0(i) = b_i(O_1).\beta_1(i) \qquad 1 \le i \le N .$$

En utilisant la technique Forward-Backward, la probabilité d'observation peut se calculer en utilisant l'une des trois équations :

$$P(O|\ \lambda) = \sum_{i=1}^{N}\alpha_t(i).\beta_t(i) \tag{2.13.a}$$

$$P(O|\ \lambda) = \sum_{i=1}^{N}\alpha_T(i) \tag{2.13.b}$$

$$P(O|\ \lambda) = \sum_{i=1}^{N}\pi_i.\beta_0(i) \tag{2.13.c}$$

Avec une complexité d'algorithme égale à ($N^2T$).

### 2.3.2. Apprentissage du modèle

Le but de l'apprentissage est de déterminer les paramètres du modèle qui maximisent la probabilité de la suite d'observations générée par le modèle $P(O|\ \lambda)$. La difficulté majeure est liée aux absences d'un critère d'optimisation global et d'une méthode directe.

L'idée de l'apprentissage est donc d'appliquer des techniques de ré-estimation qui affinent le modèle itérativement en suivant les étapes suivantes :

- Initialisation du modèle $\lambda_0 = (A_0, B_0, \Pi_0)$.
- Calculer $\lambda_{n+1}$ à partir de $\lambda_n$.
- Répéter ce processus jusqu'à un critère de fin en vérifiant que :

$$\prod_{r=1}^{R} P(O^r|\ \lambda_{n+1}) \geq \prod_{r=1}^{R} P(O^r|\ \lambda_n). \tag{2.14}$$

Cette méthode est appelée aussi l'estimation du maximum de vraisemblance (Maximum Likelihood Estimation : MLE).

Ceci montre que $\lambda_{n+1}$ doit améliorer la probabilité d'émission des observations, ce qui revient à définir une fonction $F$ telle que $\lambda_{n+1} = F(\lambda_n)$. Il existe plusieurs approches pour définir cette fonction. La plus simple consiste à faire des statistiques sur l'utilisation des transitions et des distributions. Ce qui revient à calculer des fréquences d'utilisation à partir de l'ensemble d'apprentissage. Si l'ensemble est important, les statistiques fournissent une bonne approximation des probabilités *à posteriori* utilisables alors comme paramètres du modèle pour l'itération suivante :

#### ❖ Les formules de ré-estimation

On définit :

$$\xi_t(i,j) = P(q_t = s_i, q_{t+1} = s_j|\ O, \lambda) = \frac{P(q_t = s_i, q_{t+1} = s_j, O|\ \lambda)}{P(O|\ \lambda)}. \tag{2.15}$$

En développant le numérateur et en introduisant $\alpha$, $\beta$, on aboutit à :

$$\xi_t(i,j) = \frac{\alpha_t(i).a_{ij}.b_j(O_{t+1}).\beta_{t+1}(j)}{P(O|\ \lambda)}. \tag{2.16}$$

On définit aussi la quantité

$$\gamma_t(i) = P(q_t = s_i \mid O, \lambda) = \sum_{j=1}^{N} P(q_t = s_i, q_{t+1} = s_j \mid O, \lambda) = \sum_{j=1}^{N} \xi_t(i,j). \quad (2.17)$$

Où $\sum_{t=1}^{T-1} \gamma_t(i)$ donne l'information sur le nombre de fois possibles d'être dans l'état $s_i$ et $\sum_{t=1}^{T-1} \xi_t(i,j)$ donne l'information sur le nombre de transitions possibles à partir de l'état $s_i$ à l'état $s_j$.

Des formules de ré-estimation convenables pour π, *A* et *B* peuvent être :

- $\hat{\pi}_i$ = Nombre de fois possibles d'être dans l'état $s_i$ à l'instant 1 ;
- $\hat{a}_{ij} = \dfrac{\textit{Nombre de transitions possibles de } s_i \textit{ à } s_j}{\textit{Nombre de transitions possibles à partir de } s_i}$ ;
- $\hat{b}_j(k) = \dfrac{\textit{Nombre de fois possibles d'être en } s_j \textit{ en observant } v_k}{\textit{Nombre de fois possibles d'être en } s_j}$.

Soient :

$$\hat{\pi}_i = \gamma_1(i), \quad \hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad et \quad \hat{b}_j(k) = \frac{\sum_{t=1, O_t = v_k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)} \quad (2.18)$$

Les contraintes à respecter par ces quantités sont :

$$\sum_{i=1}^{N} \hat{\pi}_i = 1 \quad et \quad \sum_{k=1}^{M} \hat{b}_j(k) = 1 \quad (2.19)$$

Ces formules sont déduites du théorème de *Baum-Welsh* dont nous donnons l'énoncé :

❖ **Algorithme de Baum-Welsh**

1. Fixer les valeurs initiales :
   $a_{ij}^0, b_j^0(k), \pi_i^0 \quad avec \ 1 \le i, j \le N, \ 1 \le k \le M$
2. Calculer à l'aide des fonctions Forward-Backward :
   $\xi_t(i,j)$ et $\gamma_t(i) \quad 1 \le i, j \le N, 1 \le t \le T\text{-}1.$
3. Nouvelles estimations, n=1, 2, 3, … (nombre d'itération)
   $\hat{a}_{ij}^n, \hat{b}_j^n(k), \hat{\pi}_i^n \quad avec \ 1 \le i, j \le N, \ 1 \le k \le M$
4. Recommencer en 2 jusqu'à un critère d'arrêt.

### 2.3.3. Estimation de la séquence d'états optimale

Cette tâche consiste à déterminer le "*meilleur*" chemin des états (suite d'états) qui maximise la probabilité d'une certaine observation $O$. Parmi les algorithmes qui effectuent cette tâche est celui de *Viterbi*.

**Algorithme de Viterbi :**

Pour trouver le "*meilleur*" chemin $Q=(q_1,q_2,...,q_T)$ pour une suite d'observations $O=(O_1,O_2,...,O_T)$, on définit $\delta_t(i)$ la probabilité du meilleur chemin amenant à l'état $s_i$ à l'instant $t$ en étant guidé par les $t$ premières observations :

$$\delta_t(i)=\max_{q_1,q_2,\delta,q_{t-1}} P(q_1,q_2,\delta\ ,q_{t-1},q_t=s_i,O_1,O_2,\delta\ ,O_t|\ \lambda). \tag{2.20}$$

Par induction, on calcule :

$$\delta_{t+1}(i)=b_j(O_{t+1})\left[\max_i \delta_t(i).a_{ij}\right] \tag{2.21}$$

Tout en gardant la trace, lors du calcul, de la suite d'états qui donne le meilleur chemin amenant à l'état $s_i$ à l'instant $t$ dans un tableau $\psi$. Formellement, l'algorithme se déroule de la façon suivante :

- Initialisation : (début de l'algorithme)

$$\left.\begin{aligned}\delta_1(i)&=\pi_i.b_i(O_1)\\ \psi_1(i)&=0\end{aligned}\right\}\quad 1\le i\le N$$

- Induction : (calcul intermédiaire)

$$\left.\begin{aligned}\delta_t(j)&=\left(\max_{1\le i\le N}\left[\delta_{t-1}(i).a_{ij}\right]\right)b_j(O_t)\\ \psi_t(j)&=\operatorname*{argmax}_{1\le i\le N}\left[\delta_{t-1}(i).a_{ij}\right]\end{aligned}\right\}\quad pour\begin{cases}1\le j\le N\\ t=2,3,...,T\end{cases}$$

- Terminaison : (fin de l'algorithme)

$P^*=\max_i[\delta_T(i)]$ $P^*$ étant la probabilité du chemin obtenu

$q_T^*=\operatorname*{argmax}_i[\delta_T(i)]$ $q_T^*$ étant le dernier état du chemin

- Chemin obtenu (en back-traking) :

$$q_t^*=\psi_{t+1}(q_{t+1}^*)\qquad 1\le t\le T-1$$

Il est à noter que la fonction "*argmax*" permet de mémoriser l'indice *i* avec lequel la valeur $\left[\delta_{t-1}(i).a_{ij}\right]$ est maximale. Donc, on utilise les données stockées dans les vecteurs $\psi_t(j)$ pour calculer le meilleur chemin. La complexité de cet algorithme est ( $N^2T$ ).

## 2.4. PRINCIPAUX TYPES DE HMM

Le type d'un HMM peut être spécifié selon :

- Contraintes liées à la matrice de transition *A*.
- Type de densité de probabilité d'observations $b_j(O_t)$**.**

### 2.4.1. Contraintes liée à la matrice de transition *A*

Deux types de modèles peuvent être obtenus selon les contraintes imposées sur les éléments de la matrice *A* :

- **Modèle ergodique (ergodic model) :** Il est sans contrainte, toutes les transitions d'un état vers un autre sont possibles, c'est-à-dire, que tous les états peuvent être atteints de n'importe où en un seul pas. On peut remarquer que la connaissance de π est primordiale pour le choix de l'état de départ (figure 2.1).

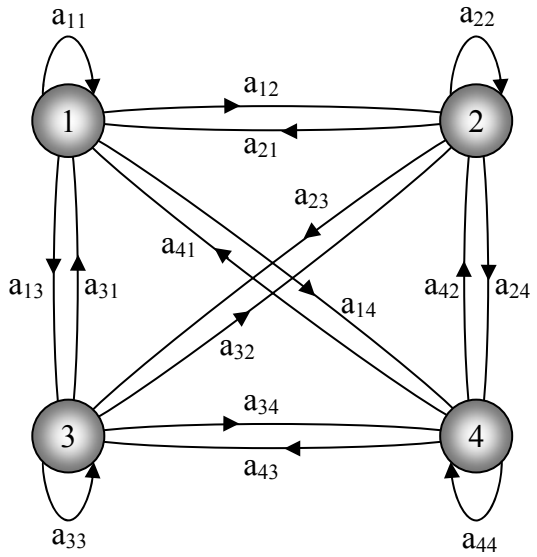



**Figure 2.1.** Modèle ergodique avec 4 états

- **Modèle gauche-droite (left-right model) :** Ces modèles peuvent être utilisés pour modéliser des processus possédant des propriétés variantes dans le temps, telle que les signaux de la parole et d'écriture manuscrite (Augustin, 01). Ils traduisent la causalité du processus de production de ces signaux. Leurs probabilités de transition vérifient: $i > j \Rightarrow a_{ij} = 0$ (figure

2.2). On peut remarquer que, dans la littérature, le modèle gauche-droite couvre souvent la plupart des applications et de ce fait, il est largement suffisant (Poisson, 05).

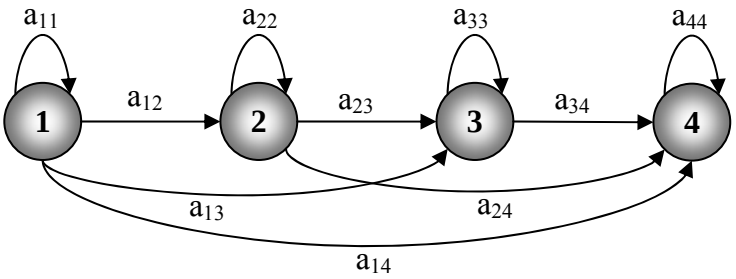



**Figure 2.2.** Modèle gauche-droite avec 4 états

### 2.4.2. Type de densité de probabilité d'observations $b_j(O_t)$

Selon le type de densité de probabilité d'observations générer par les symboles, discrète ou continue, nous pouvons construire deux types de modèles HMM : soit un HMM discret soit un HMM continu.

- **HMM discret "Discret Hidden Markov Models (DHMM)":** Les observations en général sont continues puisqu'elles proviennent de phénomènes physiques continus. Dans le cas d'un HMM discret, les observations continues sont quantifiées à l'aide d'un dictionnaire (codebook).

- **HMM continu "Continuous Hidden Markov Models (CHMM)":** Bien qu'il soit possible de quantifier les observations continues, il y a une sérieuse dégradation d'information associée à cette quantification. Il sera, alors avantageux de choisir une fonction de densité de probabilité (fdp) d'observations continues, conditionnée par les états du processus. Cependant, pour estimer, d'une façon consistante, les paramètres de cette fonction, certaines restrictions doivent être précisées sur la forme de son modèle.

Il existe plusieurs formes pour lesquelles des procédures de ré-estimation ont été formulées, à titre d'exemple : Gaussienne Scalaire, Gaussienne Multivariable, Gaussiennes Autorégressives, Mélange de Gaussiennes.

En général, on prend comme fdp un modèle de mixture de gaussiennes ayant la forme suivante (pour une observation $O_t$) :

$$b_j(O_t) = \sum_{k=1}^{K} c_{jk}\, g(O_t, \mu_{jk}, \Sigma_{jk}). \tag{2.22}$$

Avec :

$K$ : est le nombre de gaussiennes.

$c_{jk}$ : Poids de la $k^{ème}$ gaussienne de l'état $s_j$.

$\mu_{jk}$ : Moyenne de la gaussienne.

$\Sigma_{jk}$ : Matrice de variance-covariance.

$g(O_t, \mu_{jk}, \Sigma_{jk})$ : Valeur de la gaussienne au point $O_t$.

Une gaussienne de moyenne $\mu$ et de matrice variance-covariance $\Sigma$ au point $O_t$ est définie par:

$$g(O_t, \mu, \Sigma) = \frac{1}{\left(\sqrt{2\pi}\right)^d \left(\sqrt{|\det(\Sigma)|}\right)} \exp\left(-\frac{1}{2}\left\{(O_t - \mu)^{Tr} \Sigma^{-1} (O_t - \mu)\right\}\right) \quad (2.23)$$

Avec $d$ désigne la dimension du vecteur $O_t$ et Tr est la transposée.

Pour l'ajustement des paramètres de *B*, il est nécessaire de redéfinir $\gamma_t(i)$ pour l'adapter aux valeurs continues :

$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}\right]\left[\frac{c_{jk}.g(O_t, \mu_{jk}, \Sigma_{jk})}{\sum_{m=1}^{M} c_{jm}.g(O_t, \mu_{jm}, \Sigma_{jm})}\right] \quad (2.24)$$

La ré-estimation des paramètres des fonctions de densité peut être adaptée en employant les équations suivantes :

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

$$\hat{\mu}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k).O_t}{\sum_{t=1}^{T} \gamma_t(j,k)} \quad (2.25)$$

$$\hat{\Sigma}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k).(O_t - \hat{\mu}_{jk})(O_t - \hat{\mu}_{jk})^{Tr}}{\sum_{t=1}^{T} \gamma_t(j,k)}$$

## 2.6. CONCLUSION

Dans ce chapitre, nous avons décrit une méthode probabiliste de classification avec maximum de vraisemblance. On a tenté d'unifier les mécanismes de la reconnaissance probabiliste utilisés dans les modèles de Markov Cachés. Ces derniers ont été largement utilisés dans la reconnaissance automatique de l'écriture manuscrite. Cette méthode a donné de bons résultats vu la compatibilité des données avec la structure de la méthode.

La méthode adoptée dans ce travail s'intéresse à la modélisation markovienne des différentes classes de chiffres, et une compétition inter-modèle sera effectué pour sélectionner, parmi les dix modèles, celui qui maximise l'appartenance d'une forme de chiffre à la classe correspondante.

Les HMMs sont très souvent appliqués à la reconnaissance de l'écriture manuscrite (Graves et al, 09 ; Cavalin et al, 06 ; Britto et al, 03 ; Mohamed et al, 09), les données observées sont les séquences de graphèmes, ou d'observations de fenêtres ou autre segmentation implicite ou explicite de l'image. Les données cachées sont les états des HMMs.

Dans ce cas, les HMMs sont bien adaptés par leurs structures à modéliser des données séquentielles et leur capacité à modéliser justement les transitions et celle d'intégrer à la fois la segmentation et la reconnaissance

# Chapitre 3:

# Description du système de reconnaissance des chiffres manuscrits

## 3.1. INTRODUCTION

Ce chapitre est consacré à la description du système de reconnaissance des chiffres manuscrits. La première section présente l'architecture de système adopté. Une grande importance sera accordée à l'extraction de plusieurs types de caractéristiques. Enfin, la mise en œuvre d'un classificateur de type HMM continu sera détaillée.

## 3.2. ARCHITECTURE DU SYSTEME DE RECONNAISSANCE DE CHIFFRES ISOLES

Un système de reconnaissance d'un chiffre manuscrit est composé de trois modules (figure 3.2) :

- prétraitement
- extraction des caractéristiques et
- classification.

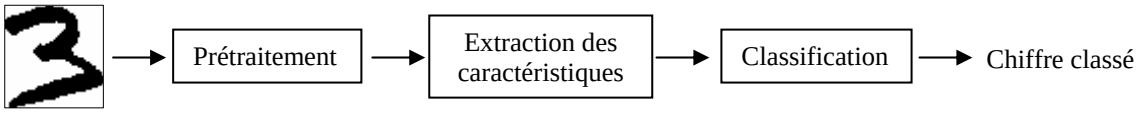



**Figure 3.1.** Architecture d'un système de reconnaissance de chiffre isolé.

Il s'agit ainsi de déterminer un modèle HMM adapté à chaque chiffre manuscrit. La présentation de chaque module est détaillée dans les sections suivantes.

### 3.2.1. Prétraitement

Le prétraitement consiste le plus souvent une binarisation. Plusieurs méthodes sont proposées dans la littérature (Trier et Jain, 95). Dans ce cas, un prétraitement supplémentaire est appliqué sur les chiffres. En effet, dans certains cas, la binarisation peut introduire des bruits dans l'image, qui se traduisent en particulier par la présence d'irrégularités le long des contours des chiffres. Ces bruits peuvent dégrader les performances de reconnaissance. Des techniques simples et rapides à mettre en oeuvre permettent de réduire l'influence de ces bruits (Cheriet et al, 07).

| | | |
|---|---|---|
| = | = | = |
| = | PC | = |
| x | x | x |

| | | |
|---|---|---|
| x | = | = |
| x | PC | = |
| x | = | = |

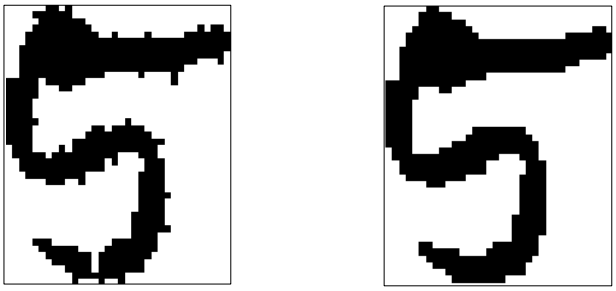
| | | |
|---|---|---|
| x | x | x |
| = | PC | = |
| = | = | = |

| | | |
|---|---|---|
| = | = | x |
| = | PC | x |
| = | = | x |

**Figure 3.2.** Masques de filtrage.

La méthode consiste à utiliser quatre masques simples pour lisser le contour des images. En notant PC le pixel central, le filtrage (ou lissage) consiste à remplacer le pixel courrant en testant

les huit voisinages. Ainsi quatre configurations sont possibles pour lisser le pixel central (figure 3.2). Les pixels notés 'x' ne sont pas pris en compte. Les pixels notés '=' sont tous égaux entre eux. Ces règles sont appliquées itérativement sur les pixels de forme et les pixels de fond. La figure 3.3 illustre un exemple de filtrage du chiffre 5.



a) avant b) après

**Figure 3.3.** Application des masques de filtrage sur le chiffre 5.

### 3.2.2. Extraction de Caractéristiques

L'extraction des caractéristiques de l'image du chiffre consiste à balayer une fenêtre glissante de la gauche vers la droite en adaptant la hauteur à celle du chiffre. Ces fenêtres sont de largeur fixe (L) et sont successivement décalées de L pixels (pas de recouvrement entre 2 trames successives) (figure 3.4). Dans chaque fenêtre, un ensemble de caractéristiques est extrait. Dans ce travail, trois catégories de caractéristiques sont considérées:

- caractéristiques classiques ou basées sur la structure du chiffre (CC) ;
- caractéristiques liées à la forme (Cforme) ;
- caractéristiques liées au fond ou à la concavité (Cfond).

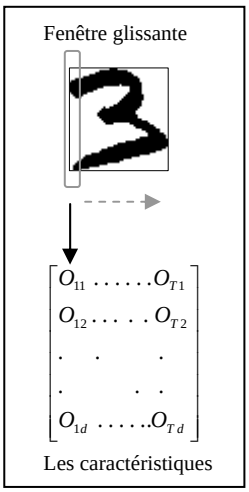



**Figure 3.4.** Extraction de caractéristiques d'une fenêtre glissante de L pixels de largeur.

### 3.2.2.1. Caractéristiques classiques

Les caractéristiques classiques utilisées dans ce travail sont au nombre de (14+L). Celles ci sont représentatives des densités et des distributions locales des pixels d'écriture. Les caractéristiques classiques (CC) sont:

- Densité : le nombre de pixels noirs dans chaque fenêtre divisé par sa taille;
- Centre de gravité : 2 coordonnées sont représentées le centre de gravité de chaque fenêtre. Elles sont normalisées par la taille de la fenêtre glissante ;
- Moment géométrique d'ordre 2: C'est une mesure statistique de la distribution des pixels autour du centre de gravité dans la fenêtre normalisée par la taille de la fenêtre;
- Histogrammes de projection : sont obtenus par projections horizontale et verticale des pixels noirs de l'image (pour l'illustration, nous utilisons un chiffre entier, figure (3.5)). Les caractéristiques utilisées dans la projection verticale sont directement les valeurs de l'histogramme normalisées selon la hauteur de la fenêtre (L caractéristiques), par contre, les caractéristiques de l'histogramme de projection horizontale sont sa moyenne et sa variance ;
- Profils (gauche, droit) : sont obtenus par l'intermédiaire de sondes appliquées sur le caractère. Pour le profil gauche d'un caractère, des sondes sont lancées depuis le bord gauche de l'image et s'arrêtent lorsqu'elles rencontrent le premier pixel noir. Les abscisses des sondes constituent le profil gauche du caractère, le même pour le profil droit mais depuis le bord droit (figure 3.6). Les caractéristiques utilisées pour chaque profil sont la moyenne et la variance;
- Nombre de transitions : selon les quatre directions principales, en comptant le nombre de transition entre les pixels (blanc/noir) ou (noir/blanc) normalisé par la taille de la fenêtre.

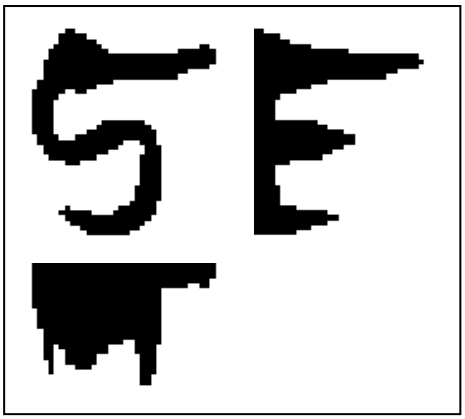


**Figure 3.5.** Histogrammes de projection horizontale et verticale.

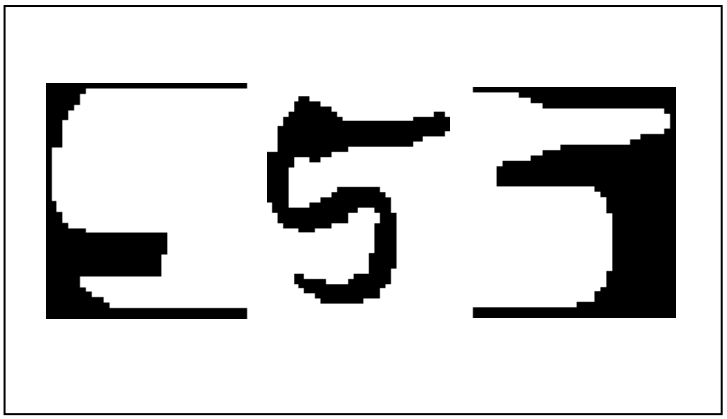


**Figure 3.6.** Profils : gauche et droit.

### 3.2.2.2. Caractéristiques liées à la forme

Le vecteur de caractéristiques basées sur la distribution des pixels noires de la forme (Cforme) se compose des caractéristiques locales calculées en tenant compte des pixels noirs des colonnes. Les caractéristiques locales sont basées sur les transitions entre les pixels noirs et blancs et vice versa. Pour chaque transition, la moyenne de la variance d'une direction sont obtenus à l'aide des estimateurs statistiques, ceux-ci sont plus appropriés aux observations directionnelles, puisqu'ils sont basés sur une échelle circulaire (Britto et al, 04 ; Chen et Wang, 00).

Par exemple, les angles de direction des observations sont compris entre 0° et 359°, soit $\theta_1, \theta_2, ..., \theta_D$ ensemble d'observations directionnelles associé à la distribution $F(\theta_i)$ et $D$ le nombre de directions possibles, tel que $1 \le i \le D$.

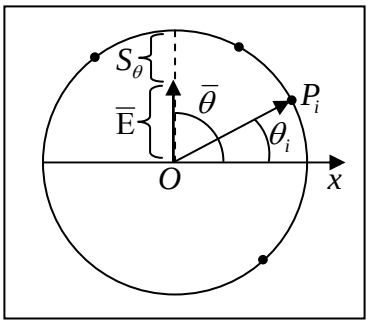



**Figure 3.7.** Cercle unitaire.

$\theta_i$ représente l'angle entre le vecteur unitaire $\overrightarrow{OP_i}$ et l'axe horizontal, tel que $P_i$ est le point d'intersection entre $\overrightarrow{OP_i}$ et le cercle unitaire. Les coordonnées cartésiennes de $P_i$ sont définies par :

$$(\cos(\theta_i), \sin(\theta_i)) \tag{3.1}$$

La direction moyenne circulaire $\overline{\theta}$ des $D$ observations directionnelles sur le cercle d'unité correspond à la direction du vecteur $\overline{\mathrm{E}}$ obtenu par la somme des vecteurs d'unité $\left(\overrightarrow{OP_1}, ..., \overrightarrow{OP_i}, ..., \overrightarrow{OP_D}\right)$. Ainsi, le centre de gravité $\left(\overline{C}, \overline{G}\right)$ de $D$ coordonnées est défini comme suit :

$$\begin{aligned} \overline{C} &= \frac{1}{\sum_{i=1}^{D} F(\theta_i)} \sum_{i=1}^{D} F(\theta_i).\cos(\theta_i) \\ \overline{G} &= \frac{1}{\sum_{i=1}^{D} F(\theta_i)} \sum_{i=1}^{D} F(\theta_i).\sin(\theta_i) \end{aligned} \tag{3.2}$$

Ces coordonnées sont utilisées pour estimer la moyenne $\overline{\mathrm{E}}$ :

$$\overline{\mathrm{E}} = \sqrt{\left(\overline{C}^{\,2} + \overline{G}^{\,2}\right)} \tag{3.3}$$

La direction moyenne circulaire peut être obtenue par la solution de l'une des équations :

$$\begin{aligned} \cos(\overline{\theta}) &= \frac{\overline{C}}{\overline{\mathrm{E}}} \\ \sin(\overline{\theta}) &= \frac{\overline{G}}{\overline{\mathrm{E}}} \end{aligned} \tag{3.4}$$

Finalement, la variance circulaire moyenne du $\overline{\theta}$ est calculée comme suit :

$$S_\theta = 1 - \overline{\mathrm{E}} \qquad 0 \le S_\theta \le 1 \tag{3.5}$$

Pour estimer $\overline{\theta}$ et $S_\theta$ pour chaque transition de l'image de chiffre, nous considérons {0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°} comme ensemble d'observations directionnelles, tandis que $F(\theta_i)$ est calculé en comptant le nombre de pixel noirs successifs à travers la direction $\theta_i$ dans une transition jusqu' à la rencontre d'un pixel blanc.

Sur la figure 3.8, les transitions dans une colonne du numéro 9 sont énumérées de 1 à 6. Les observations directionnelles possibles sont montrées pour les transitions 3 et 6.

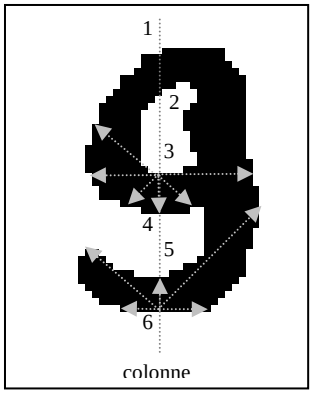



**Figure 3.8.** Transitions dans une colonne du chiffre 9 pour estimer la direction moyenne des transitions 3 et 6.

En plus de cette information directionnelle, nous rajoutons une autre caractéristique locale: la position relative de chaque transition par rapport au point supérieur du chiffre. Après la

vérification de plusieurs images de chiffre de la base NIST SD19, nous définissons un maximum de 8 transitions par colonne. Ainsi, le vecteur de caractéristique se compose à ce moment de 24 (3x8) caractéristiques.

### 3.2.2.3. Caractéristiques liées au fond

Les caractéristiques liées au fond (Cfond) permettant de mettre en évidence l'information de concavité. Elles sont employées pour ressortir les propriétés topologiques et géométriques des classes de caractère. Chaque caractéristique de concavité représente le nombre de pixels blancs appartenant à une configuration spécifique de concavité. L'étiquette pour chaque pixel blanc est choisie à base de la présence d'un ou plusieurs pixels noirs dans les quatre directions principales. Chaque direction est explorée jusqu' à la rencontre d'un pixel noir ou des limites imposées par les extrémités de chiffre (Cavalin et al, 06). Ainsi, neuf (9) configurations de concavité sont possibles. Par ailleurs, nous considérons cinq (5) configurations supplémentaires afin de détecter plus précisément la présence des boucles. La longueur de ce vecteur est alors 14. La figure (3.9) montre les 9 configurations de concavité ainsi que les 5 configurations (10, 11, 12, 13, 14) pour les boucles fausses. La figure 3.10 illustre les configurations possibles pour le chiffre 8.

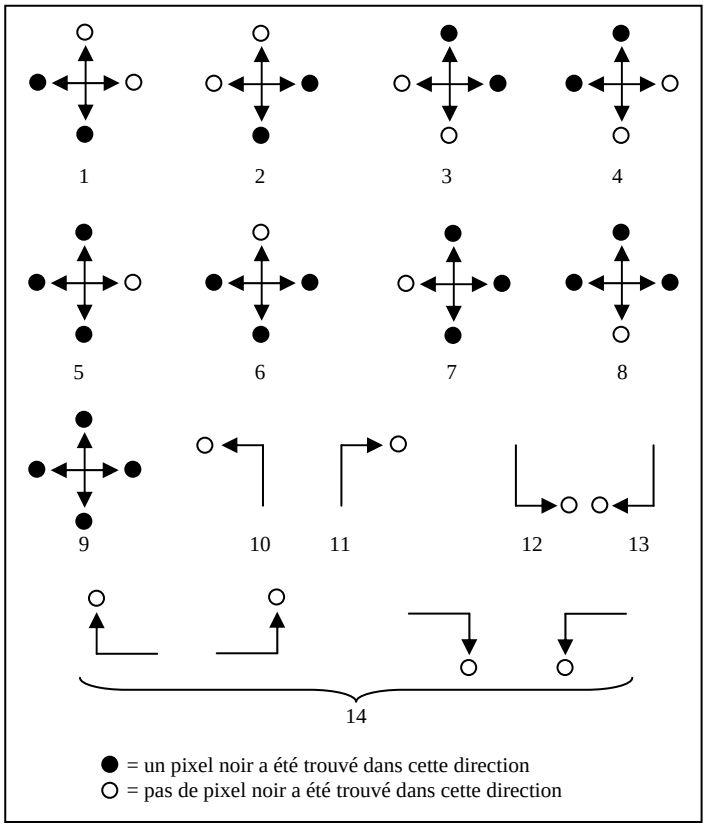



**Figure 3.9.** Différentes configurations de concavité.

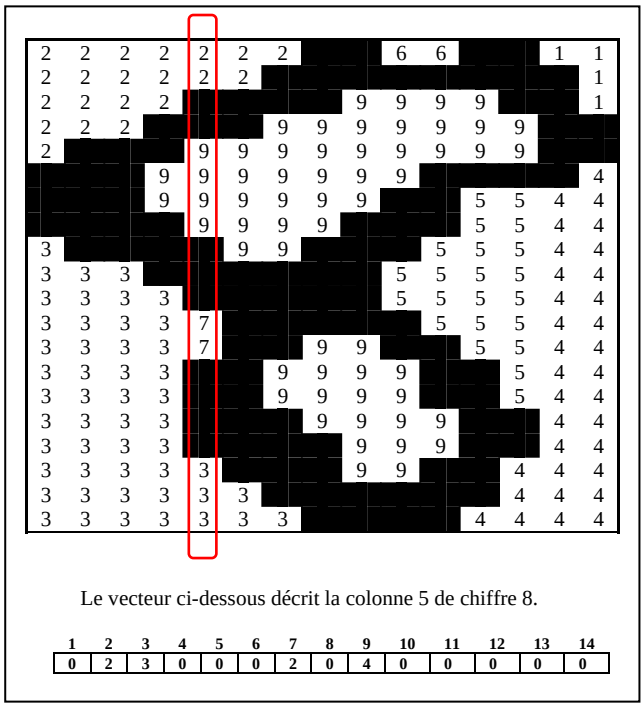



**Figure 3.10**. Etiquettes de concavité calculées pour le chiffre 8.

### 3.2.3. Classification

Le système de reconnaissance modélise les chiffres sous la forme de modèles de Markov cachés continus (CHMMs) du type gauche- droite. Les densités de probabilité des observations dans chaque état sont modélisées par une somme pondérée de gaussiennes multidimensionnelles (Gaussian Mixture Model, GMM). Au total, 10 modèles sont construits pour les chiffres manuscrits.

#### 3.2.3.1. Densités d'observations continues

Pour pouvoir utiliser des observations continues, on doit remplacer la distribution *B* de probabilités d'observations par GMM. La valeur de cette fonction pour une valeur d'observation donnée n'est alors plus une probabilité mais une vraisemblance (elle peut être supérieure à 1).

$$b_j(O_t) = \sum_{k=1}^{K} c_{jk}\, g(O_t, \mu_{jk}, \Sigma_{jk}) \tag{3.6}$$

Les $c_{jk}$ sont des coefficients de pondération satisfaisant la contrainte :

$$\sum_{k=1}^{K} c_{jk} = 1 \quad tel\ que \quad 1 \le j \le N$$
$$et \quad c_{jk} > 0 \quad tel\ que \quad 1 \le j \le N, 1 \le k \le K \tag{3.7}$$

Ainsi, la fonction de densité est bien normalisée :

$$\int_{-\infty}^{+\infty} b_j(x)dx = 1 \tag{3.8}$$

L'utilisation de plusieurs gaussiennes permet une meilleure modélisation de la probabilité $P(O_t / q_i)$. Le nombre de paramètres utilisés est alors beaucoup moins important car les matrices de variance-covariance des distributions sont diagonales.

### 3.2.3.2. Initialisation des modèles

L'algorithme d'apprentissage garantit l'obtention des valeurs optimales du modèle de telle sorte que $P(O|\lambda)$ soit un point critique. Expérimentalement, ce point critique correspond en fait à un maximum local, et ainsi différentes valeurs initiales des paramètres du modèle peuvent conduire à des valeurs différentes de $P(O|\lambda)$. Comme le montre la figure 3.11, $\lambda$ correspond au modèle de départ, $\hat{\lambda}$ correspond au modèle ré-estimé et $\lambda^*$ désigne le modèle optimal recherché. Il s'agit donc de trouver des valeurs des paramètres conduisant à une valeur de $P(O/\lambda)$ globalement maximum (et donc constituer le meilleur modèle possible).

Il est à noter que le choix d'un bon ensemble de paramètres initiaux est très important car celui ci permet une convergence rapide du modèle.

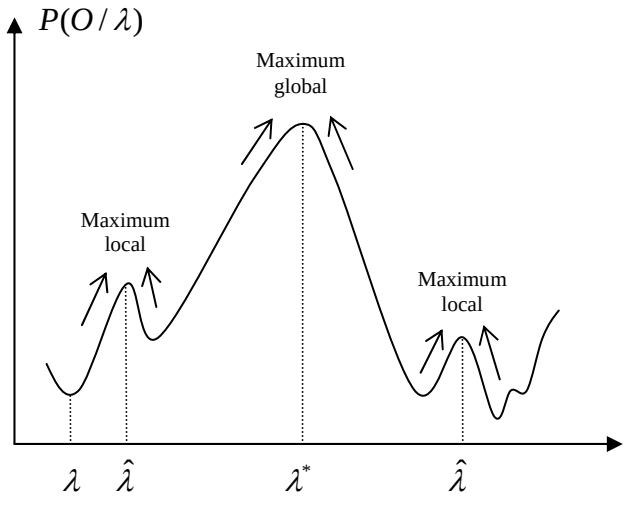



**Figure 3.11.** Points critiques.

Il existe plusieurs méthodes d'initialisation et le seul critère pour l'évaluation de ces méthodes est la qualité des résultats obtenus. Parmi ces méthodes, l'initialisation uniforme

(équiprobabilité) et l'initialisation aléatoire sont les plus utilisées. Cette dernière consiste à prendre aléatoirement des probabilités initiales et de transitions tout en respectant les deux conditions suivantes :

$$\begin{cases} \sum_{l=1}^{N} \pi_l = 1 \quad \forall l \le N \\ 0 \le a_{jl} \le 1 \;\; et \;\; \sum_{l=1}^{N} a_{jl} = 1 \quad \forall j,l \le N \end{cases} \tag{3.9}$$

En effet, Rabiner (Rabiner, 89) a constaté, par expérience, que des valeurs initiales aléatoires des matrices de transition et des matrices de probabilités initiales donnent de bons résultats.

*Remarques* :

- Le choix du modèle initial influe sur les résultats ; toutes les valeurs de *A* et *B* égales à 0 au départ restent égales à 0 à la fin de l'apprentissage.
- L'algorithme converge vers des valeurs de paramètres qui forment un point critique de $P(O|\ \lambda)$. Ce point critique peut correspondre à un maximum local ou à un point d'inflexion (maximum global), d'où le bon choix du modèle.
- Le nombre d'itérations est fixé empiriquement.

### 3.2.3.3. Apprentissage du CHMMs

L'apprentissage consiste à rechercher les paramètres des modèles maximisant la vraisemblance (MLE) de l'ensemble des données d'apprentissage. L'optimisation des paramètres GMMs est réalisée par l'algorithme de Baum-Welch par adaptation de l'algorithme EM (Expectation-Maximization). L'algorithme itératif EM est effectué de la façon suivante : les observations sont associées aux états les plus probables pendant la phase d'espérance (E), les paramètres des modèles sont re-estimés pendant la phase de maximisation (M). Ces deux étapes sont itérées jusqu'à la convergence.

Dans la pratique, des problèmes apparaissent souvent dans l'apprentissage de tels modèles de mélange (GMM), notamment pour des données en " grande " dimension. Les matrices de covariance obtenues ne sont pas toujours bien conditionnées et leur inversion pose souvent un problème. Une technique répandue consiste à régulariser les solutions pour pouvoir inverser la matrice variance-covariance.

Dans notre implémentation, à chaque étape de EM, après l'étape de ré-estimation, les matrices de covariances sont régularisées en ajoutant une faible valeur aux éléments de la diagonale (TriDo et Artières, 07):

$$\Sigma_{jk} = \Sigma_{jk} + \varepsilon.Id \tag{3.10}$$

Où $\varepsilon$ est un hyper-paramètre.

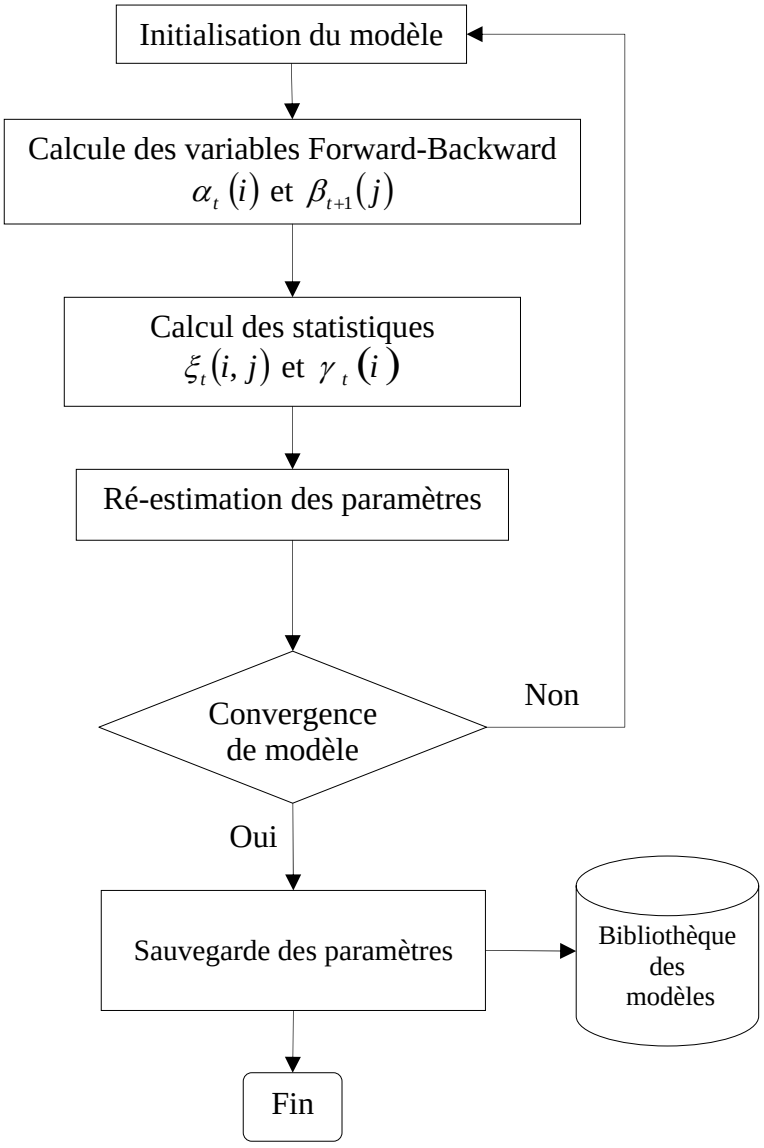



**Figure 3.12.** Algorithme d'apprentissage.

L'idée de l'apprentissage est donc d'appliquer des techniques de ré-estimation qui affinent le modèle itérativement en suivant l'algorithme ci-dessus (figure 3.12). Un schéma explicatif de l'apprentissage pour le chiffre 3 est illustré dans la figure 3.13.

On doit toujours vérifier que :

$$\prod_{r=1}^{R} P(O^r \mid \lambda_{n+1}) \geq \prod_{r=1}^{R} P(O^r \mid \lambda_n). \tag{3.11}$$

Où $R$ représente le nombre d'échantillons.

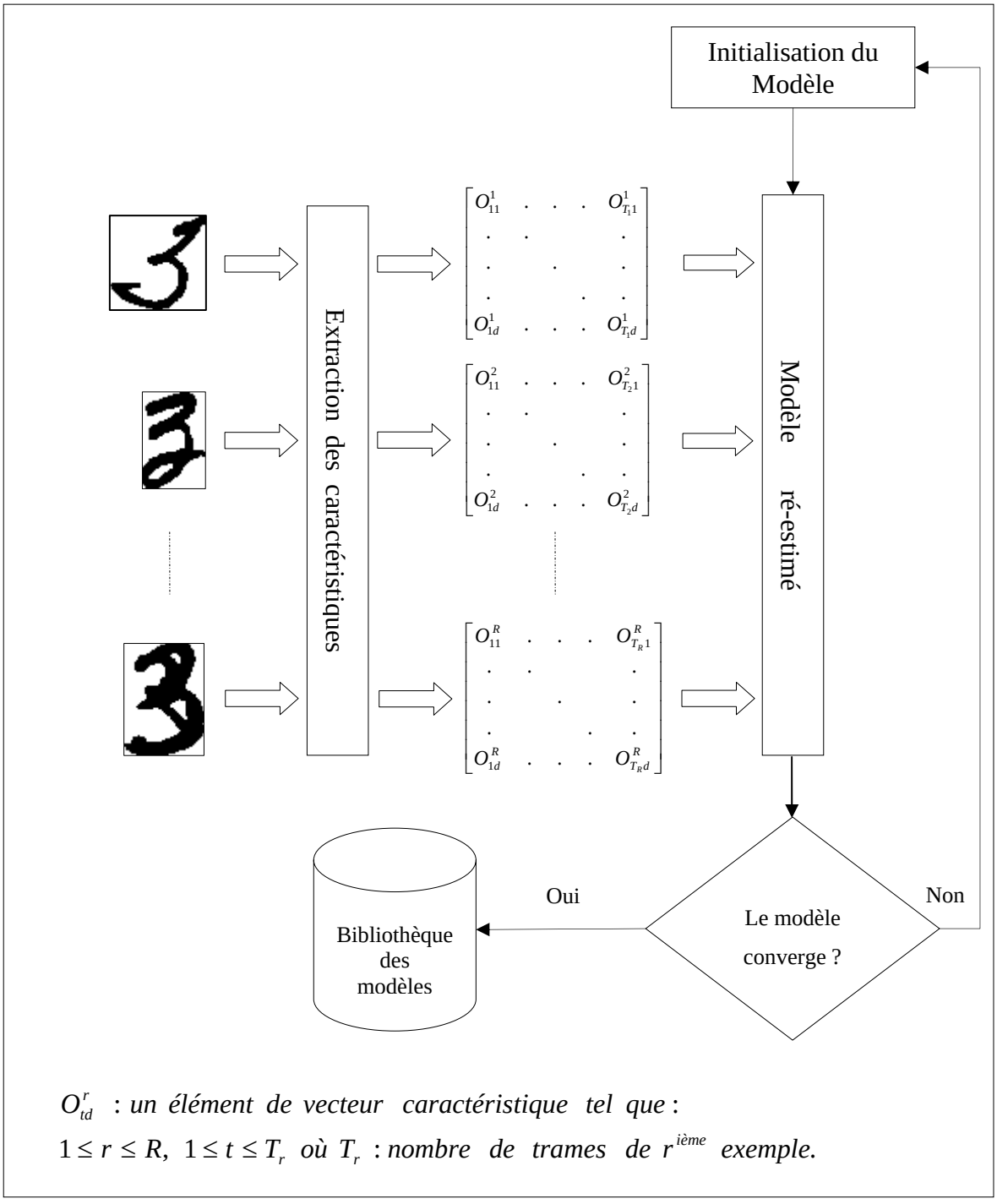



**Figure 3.13.** Schéma explicatif de l'apprentissage pour le chiffre 3.

La valeur de $P(O \mid \lambda)$ augmente à chaque itération et l'algorithme peut converger vers un optimum local. Malheureusement, il existe généralement un grand nombre d'optimums locaux et le paramétrage obtenu par ce processus dépend fortement du paramétrage initial choisi. Cela pose un problème relatif à l'absence de critère d'optimisation global et à l'absence d'une méthode directe.

L'optimum global est très difficilement atteint et il est nécessaire de définir un critère d'arrêt par exemple en stoppant la procédure lorsque l'augmentation de $P(O \mid \lambda)$ est inférieure à un certain seuil, ou lorsque le nombre d'itérations effectuées est jugé suffisamment important.

### 3.2.3.4. Reconnaissance par les CHMMs

La reconnaissance utilise l'algorithme de Viterbi. La séquence d'observations issue du chiffre à reconnaître est confrontée à la bibliothèque de modèles de chiffres. La séquence d'observations d'un caractère correspondant à la vraisemblance maximale $P(O \mid \lambda)$ identifie le chiffre reconnu. La figure 3.14 illustre un schéma explicatif de la reconnaissance pour le chiffre 3.

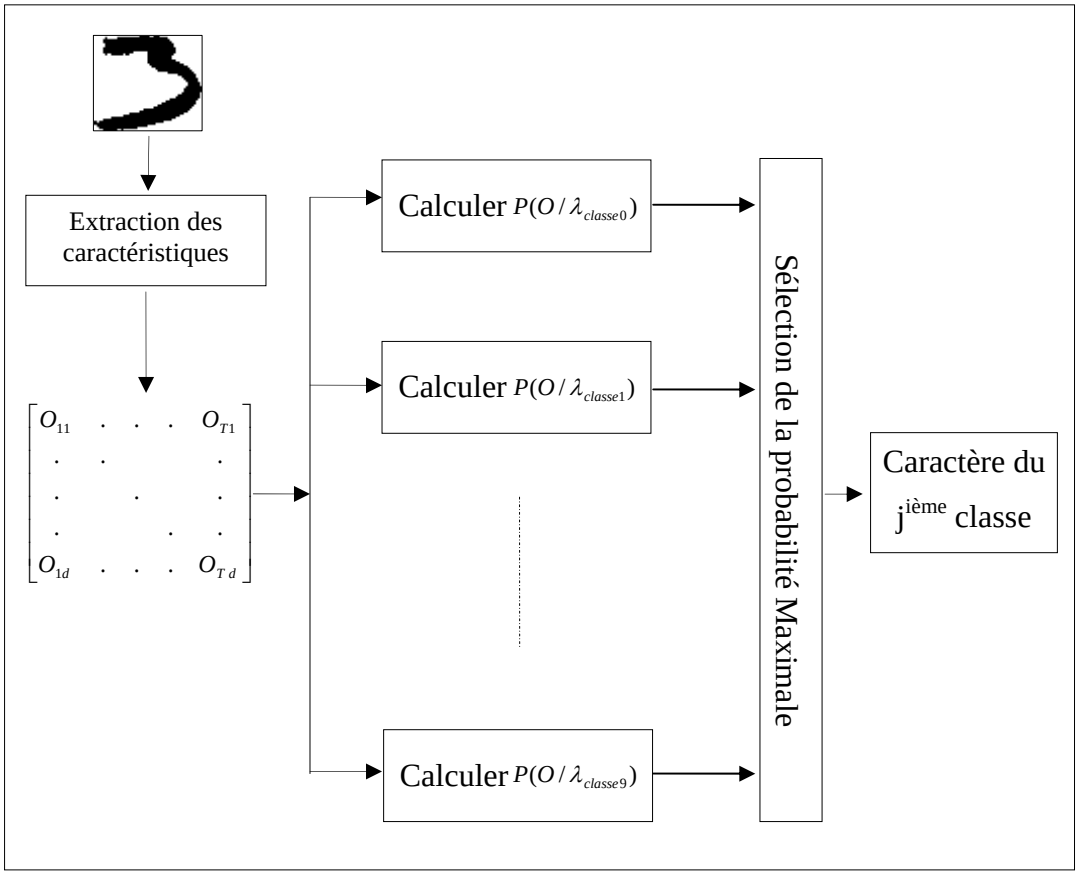



**Figure 3.14.** Schéma explicatif de la reconnaissance pour le chiffre 3.

## 3.3. CONCLUSION

Ce chapitre a été consacré à la description d'un système de reconnaissance des chiffres manuscrits en utilisant modèles de Markov cachés. Le système de trois modules : prétraitement, extraction de caractéristiques et génération du modèle de classification.

Dans notre travail, la recherche de méthodes d'extraction de caractéristiques a connu une attention particulière pour pouvoir générer un modèle HMM le plus représentatif. Ainsi, deux problèmes majeurs devraient être résolus : choix optimaux des paramètres du modèle et l'apprentissage du modèle.

Dans le chapitre suivant, nous proposons d'évaluer notre système sur une base de données standard.

# Chapitre 4:
# Résultats Expérimentaux

## 4.1. INTRODUCTION

Ce chapitre dédié à la présentation des résultats obtenus pour la validation du système. La description de la base de données utilisée est tout d'abord présentée. Puis plusieurs expériences sont menées pour l'étude de l'influence des paramètres initiaux (largeur de la fenêtre glissante, nombre d'états et nombre de gaussiennes pour chaque état) sur les performances du système. Les résultats obtenus sont présentés avec la combinaison des différentes catégories de caractéristiques. Ces résultats permettent de mesurer les performances obtenues lors de l'utilisation des caractéristiques statistiques, liées à la forme et au fond. Finalement, pour bien mettre en évidence la nature bidimensionnelle des données (image 2D), nous proposons d'évaluer les effets engendrés par la combinaison des CHMMs liées respectivement aux lignes et aux colonnes.

## 4.2. PROTOCOLE EXPERIMENTAL

### 4.2.1. Bases de données

Les expériences ont été conduites sur une base de données composée de chiffres manuscrits isolés stockés sous forme d'images binaires (figure 4.1). Cette base, appelée NIST SD19, est rendue disponible par l'organisme américain National Institute of Standards and Technology (NIST). Les chiffres manuscrits isolés présentent l'avantage d'être abondamment disponibles sous la forme de base de données étiquetées. Cette grande disponibilité permet d'ajuster le nombre d'exemples d'apprentissage selon les besoins tout en conservant de larges bases pour le test (Oliveira, 03b). Cette base est divisée en 3 divisions (hsf_0123, hsf_4 et hsf_7), la répartition de chaque division est illustrée dans le Tableau 4.1.

| Nom de la division | Nombre d'exemples |
|---|---|
| hsf_0123 | 195 000 |
| hsf_4 | 58 646 |
| hsf_7 | 60 089 |

**Tableau 4.1.** Répartition de la base NIST SD19 pour les chiffres isolés

Á partir de la division hsf_0123, une fraction de la base de données correspond à 15.568 exemples est considéré pour nos expériences dont 10.184 et 5.384 exemples sont utilisés respectivement pour l'apprentissage et le test.

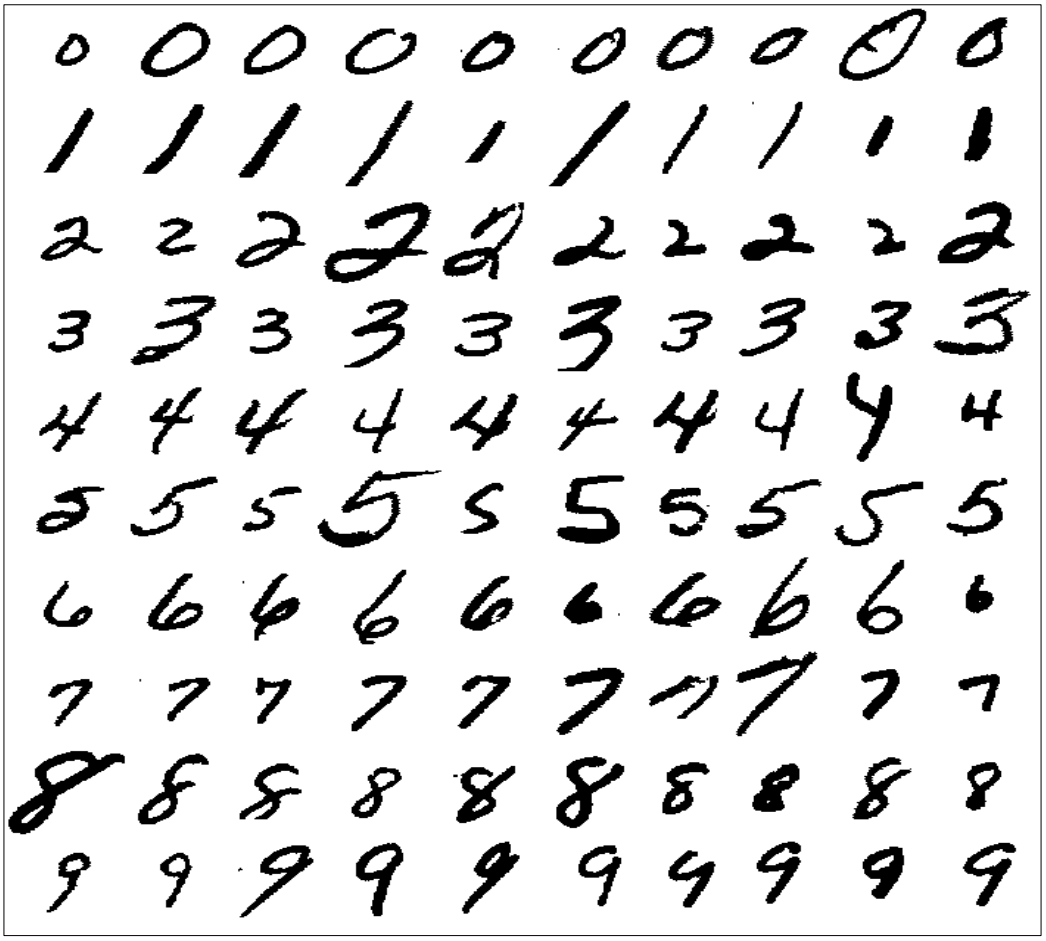


**Figure 4.1.** Exemples de la base de données NIST SD19.

| | Largeur des chiffres (en pixels) | | | Hauteur des chiffres (en pixels) | | |
|---|---|---|---|---|---|---|
| Classe | Minimale | Moyenne | Maximale | Minimale | Moyenne | Maximale |
| 0 | 19 | 36 | 67 | 19 | 39 | 77 |
| 1 | 6 | 21 | 64 | 24 | 44 | 80 |
| 2 | 24 | 43 | 89 | 21 | 43 | 80 |
| 3 | 19 | 38 | 75 | 25 | 46 | 75 |
| 4 | 16 | 42 | 88 | 27 | 52 | 82 |
| 5 | 21 | 50 | 90 | 25 | 46 | 73 |
| 6 | 19 | 35 | 68 | 27 | 48 | 81 |
| 7 | 17 | 36 | 71 | 26 | 47 | 77 |
| 8 | 20 | 39 | 82 | 30 | 50 | 80 |
| 9 | 17 | 35 | 69 | 29 | 53 | 80 |

**Tableau 4.2.** Variabilité de données de NIST SD19.

#### 4.2.1.2. Variabilité de données de NIST SD19

Les prototypes de la base NIST SD19 ne sont pas normalisés, le tableau 4.2 montre cette variabilité en largeur et hauteur des chiffres. Dans ce travail, nous nous intéressons à identifier les chiffres manuscrits sans faire la normalisation c-à-d que la forme réelle du chiffre est gardée quelle que soit sa taille. Á ce stade, nos efforts se sont focalisés sur la recherche des méthodes d'extraction des caractéristiques. D'une manière générale, la taille des vecteurs caractéristiques est de taille constante quelque soit les dimensions de l'image du chiffre.

La figure 4.2 illustre un exemple de chiffre de taille différente.

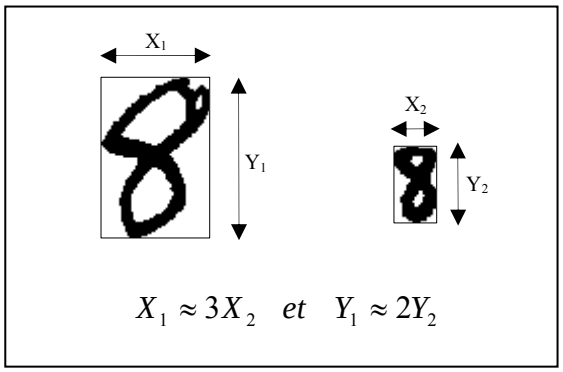



**Figure 4.2.** Variabilité de taille entre deux exemples de la même classe (chiffre 8).

### 4.2.2. Critère d'évaluation

La matrice de confusion (ou matrice d'erreur) est un excellent critère d'évaluation de la classification. Cette matrice permet d'estimer les taux de bonne classification (ou de bonne reconnaissance) pour chaque modèle et également le taux global du système. Nous avons choisi de mettre les classes de référence en lignes et les résultats de test en colonnes pour chaque classe (Nemmour, 03). Dans ce travail, nous utilisons deux taux : taux de reconnaissance par classe (équation 4.1) et le taux moyen de bonne reconnaissance (TMBR) qui est défini comme étant le nombre de chiffres reconnus sur le nombre totale du chiffres exprimé en % (équation 4.2).

$$TRC\ (\%) = \frac{nombre\ de\ chiffres\ reconnus\ d'une\ classe}{nombre\ de\ chiffres\ de\ la\ classe} \tag{4.1}$$

$$TMBR\ (\%) = \frac{nombre\ de\ chiffres\ reconnus}{nombre\ totale\ de\ chiffres} \tag{4.2}$$

## 4.3. ÉTUDE DE L'INFLUENCE DES PARAMETRES SUR LE SYSTEME

Cette section vise à évaluer les performances du système en tentant d'étudier l'influence des différentes paramètres notamment :

- largeur de la fenêtre (L) ;
- nombre d'états (N) ;
- nombre de mixture de gaussiennes (K).

Dans cette section, chaque classe de chiffre est représentée par un CHMM de type gauche-droite. Le modèle GMM est fixé à plusieurs gaussiennes pour chaque état. Les matrices Π et *A* sont initialisées de manière aléatoire. Ainsi, le nombre d'itérations est fixé à onze (11). Nous avons extrait les 17 caractéristiques classiques (CC) citées dans le paragraphe (3.2.2.1) du chapitre précédent pour évaluer les tests.

### 4.3.1. Influence de la largeur de la fenêtre

Cette étape consiste à trouver la largeur optimale (L, en pixels) de la fenêtre glissante donnant le meilleur Taux Moyen de Bonne Reconnaissance (TMBR). Nous tentons trois expériences pour trois valeurs de L (2, 3 et 4 pixels). Chaque modèle est défini par cinq (5) états et pour chaque état six (K=6) gaussiennes sont pris pour modéliser les vecteurs caractéristiques.

Les tableaux (4.3, 4.4 et 4.5) illustrent le taux de reconnaissance de chaque classe pour les trois valeurs de L sur la base de test. Nous remarquons que le TMBR croît de 41.49% à 74.57%, ce qui montre que la fenêtre de 4 pixels produit une amélioration considérable (de 33.08 %) de la performance en reconnaissance. Ceci démontre que l'information apportée avec une fenêtre de 4 pixels est plus pertinente que celle de 2 et 3 pixels.

Le tableau 4.3 montre une forte confusion entre les classes, notamment dans les colonnes des classes de test 0, 1, 7 et 9. Ceci est dû au fait que la fenêtre de 2 pixels présente des formes quasi-similaires. En augmentant la largeur de la fenêtre à 3 pixels (tableau 4.4), nous remarquons que la confusion entre les classes diminue, mais elle demeure dans les 3 classes de test (2, 5 et 8). Ce qui montre que la fenêtre de 2 ou 3 pixels est incapable de générer des trames représentatives, elle fournit des caractéristiques non discriminantes.

Classes de test

| Classes de référence | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | **59.77** | 8.25 | 8.55 | 0 | 0 | 0 | 14.60 | 4.13 | 0 | 4.70 |
| 1 | 6.65 | **60.44** | 2.10 | 9.28 | 0 | 2.90 | 2.20 | 10.63 | 5.80 | 0 |
| 2 | 8.45 | 34.20 | **43.73** | 6.75 | 0 | 0 | 0 | 4.32 | 0 | 2.55 |
| 3 | 4.17 | 82.08 | 06.65 | **29.25** | 0 | 2.77 | 0 | 2.42 | 0 | 2.66 |
| 4 | 10.11 | 18.20 | 4.03 | 2.00 | **33.24** | 2.12 | 0 | 10.20 | 0 | 18.10 |
| 5 | 8.50 | 28.66 | 2.23 | 11.30 | 0 | **36.41** | 0 | 10.20 | 0 | 2.70 |
| 6 | 4.05 | 30.42 | 0 | 0 | 0 | 0 | **37.10** | 10.13 | 0 | 18.30 |
| 7 | 12.78 | 36.91 | 0 | 0 | 0 | 0 | 0 | **45.77** | 2.20 | 2.34 |
| 8 | 2.45 | 40.32 | 4.10 | 14.75 | 0 | 0 | 0 | 2.23 | **27.85** | 8.30 |
| 9 | 0 | 20.02 | 2.30 | 2.21 | 22.00 | 0 | 0 | 12.12 | 0 | **41.35** |

TMBR = **41.49 %**

**Tableau 4.3.** Matrice de confusion calculée pour *L*=2.

Classes de test

| Classes de référence | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | **82.45** | 0 | 4.55 | 0 | 0 | 0 | 4.60 | 0 | 8.40 | 0 |
| 1 | 0 | **81.40** | 2.12 | 0 | 2.88 | 2.60 | 0 | 2.10 | 8.90 | 0 |
| 2 | 6.43 | 0 | **59.44** | 0 | 0 | 16.73 | 12.50 | 0 | 4.90 | 0 |
| 3 | 0 | 0 | 4.30 | **42.06** | 0 | 40.74 | 0 | 2.30 | 10.60 | 0 |
| 4 | 0 | 0 | 8.60 | 0 | **46.51** | 10.20 | 0 | 10.11 | 2.08 | 12.50 |
| 5 | 0 | 0 | 2.00 | 4.50 | 0 | **85.20** | 0 | 0.31 | 4.82 | 3.16 |
| 6 | 28.10 | 0 | 2.60 | 0 | 0 | 0 | **57.98** | 2.58 | 6.44 | 2.30 |
| 7 | 2.09 | 10.12 | 2.17 | 0 | 2.75 | 4.62 | 2.50 | **73.15** | 2.60 | 0 |
| 8 | 0 | 0 | 10.03 | 2.46 | 0 | 8.80 | 0 | 0 | **76.11** | 2.30 |
| 9 | 0 | 0 | 4.71 | 0 | 11.30 | 10.12 | 9.03 | 8.70 | 8.45 | **47.69** |

TMBR = **66.00 %**

**Tableau 4.4.** Matrice de confusion calculée pour *L*=3.

Classes de test

| Classes de référence | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | **83.10** | 0 | 3.90 | 0 | 0 | 3.80 | 6.20 | 0 | 2.00 | 1.00 |
| 1 | 0 | **91.20** | 0 | 0 | 2.80 | 4.10 | 0 | 1.90 | 0 | 0 |
| 2 | 02.0 | 0 | **79.70** | 0 | 0 | 9.80 | 4.20 | 0 | 4.30 | 0 |
| 3 | 0 | 0 | 10.15 | **57.84** | 0 | 25.10 | 0 | 0 | 4.36 | 2.55 |
| 4 | 0 | 2.33 | 12.2 | 0 | **62.47** | 10.70 | 0 | 6.60 | 2.90 | 2.80 |
| 5 | 0 | 0 | 6.45 | 2.75 | 0 | **85.92** | 0 | 0.25 | 4.63 | 0 |
| 6 | 10.33 | 0 | 2.15 | 0 | 0 | 0 | **74.33** | 2.00 | 6.87 | 4.32 |
| 7 | 10.15 | 0 | 6.03 | 0 | 3.32 | 6.20 | 6.20 | **71.27** | 2.73 | 4.10 |
| 8 | 0 | 0 | 12.10 | 5.33 | 0 | 2.27 | 0 | 0 | **80.30** | 0 |
| 9 | 4.15 | 0 | 0 | 0 | 2.44 | 8.06 | 14.80 | 4.07 | 6.90 | **59.58** |

TMBR = **74.57 %**

**Tableau 4.5.** Matrice de confusion calculée pour *L*=4.

### 4.3.2. Influence du nombre d'états

Pour trouver le nombre d'états optimal (N) adapté pour chaque modèle, Nous tentons plusieurs expériences en faisant varier la valeur de N pour mesurer les performances de chaque modèle. La largeur de la fenêtre est fixée à quatre (4) pour les caractéristiques des chiffres. Le nombre de gaussiennes (K) pour chaque état est fixé à six (6). Le tableau 4.6 illustre les taux obtenus pour chaque classe pour quatre configurations différentes.

| | Nombre d'états (N) | | | |
|---|---|---|---|---|
| Classes | 3 | 4 | 5 | 6 |
| 0 | 82.79 | 82.92 | **83.10** | 83.04 |
| 1 | **92.35** | 91.21 | 91.20 | 91.08 |
| 2 | 79.33 | 79.64 | 79.70 | **81.53** |
| 3 | 56.51 | 57.22 | **57.84** | 57.77 |
| 4 | 61.94 | **63.18** | 62.47 | 61.86 |
| 5 | 81.31 | 83.60 | 85.92 | **87.15** |
| 6 | 71.43 | 72.97 | **74.33** | 74.31 |
| 7 | 68.46 | 69.05 | **71.27** | 70.28 |
| 8 | 82.35 | 84.71 | 80.30 | **82.94** |
| 9 | 59.53 | **60.43** | 59.58 | 59.46 |
| TMBR | 73,60 | 74,49 | 74.57 | 74,94 |

**Tableau 4.6.** Evaluation de nombre d'états.

Les résultats rapportés dans le tableau 4.6 montre que le TMBR obtenu est maximal (74.94 %) pour N=6. L'examen des taux pour chaque classe montre le nombre d'états optimal dépend de la forme du chiffre. Par exemple, les chiffres {2, 5 et 8} nécessitent 6 états ; les chiffres {0, 3, 6 et 7} nécessitent 5 états tandis les chiffres {1} et {4 et 9} nécessitent respectivement 3 et 4 états. Nous reportons dans le tableau 4.7 les meilleurs taux obtenus pour chaque classe. Nous remarquons que le TMBR est notablement amélioré (de 74.94 % pour 6 états à 75.21 % pour N optimal pour chaque classe).

| Classe | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | TMBR (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| N optimal | 5 | 3 | 6 | 5 | 4 | 6 | 5 | 5 | 6 | 4 | |
| TRC (%) | 81.10 | 92.35 | 81.53 | 57.84 | 63.18 | 87.15 | 74.33 | 71.27 | 82.94 | 60.43 | 75.21 |

**Tableau 4.7.** TRC calculé pour N optimal pour chaque classe.

### 4.3.3. Influence de nombre de gaussiennes

L'objectif ici est de trouver le nombre optimal de mixture de gaussiennes (K) offrant le meilleur TMBR. La largeur de la fenêtre est fixée à quatre pixels et le nombre d'états de chaque modèle est lui-même N optimal. Les résultats obtenus sont illustrés dans le tableau 4.8 pour 6, 9, 12, et 15 gaussiennes pour chaque état.

| | K pour chaque état | | | |
|---|---|---|---|---|
| Classe | 6 | 9 | 12 | 15 |
| 0 | 83.10 | 84.29 | 84.74 | 84.75 |
| 1 | 92.35 | 92.74 | 93.18 | 93.18 |
| 2 | 82.53 | 84.27 | 86.63 | 86.67 |
| 3 | 57.84 | 59.36 | 61.21 | 61.23 |
| 4 | 63.18 | 66.07 | 67.36 | 67.36 |
| 5 | 87.15 | 88.41 | 88.79 | 88.82 |
| 6 | 74.33 | 75.18 | 76.20 | 76.18 |
| 7 | 71.27 | 74.25 | 74.93 | 74.89 |
| 8 | 86.94 | 88.03 | 88.56 | 88.64 |
| 9 | 60.43 | 63.77 | 64.08 | 64.11 |
| TMBR | 75.71 | 77.63 | 78.56 | 78.58 |

**Tableau 4.8.** Influence du nombre de gaussiennes sur le TMBR.

Nous pouvons constater que le TMBR augmente avec le nombre de gaussiennes, le TMBR le plus bas est 75.71 % avec 6 gaussiennes par état, l'augmentation de nombre de gaussiennes permet d'améliorer le TMBR pour atteindre 78.56 % avec 12 gaussiennes par état. Avec cette valeur, la mixture de gaussiennes peut bien modéliser les distributions réelles des données. Nous avons montré aussi que le nombre de gaussiennes dans un CHMM peut être diminué (de 15 à 12) tout en conservant un niveau de performance acceptable et en diminuant le temps de calcul, la différence entre le TMBR avec 12 gaussiennes et celui de 15 gaussiennes est d'ordre 0.02%, cette différence est pratiquement non significative.

### 4.3.3. Discussion

La difficulté majeure des HMMs est la recherche des paramètres optimaux (largeur de la fenêtre, nombre d'états et nombre de gaussiennes) adaptés pour chaque classe. En effet, il faut tenter plusieurs expériences pour trouver le meilleur modèle HMM. D'une manière générale, nous constatons que plus la largeur du chiffre est importante, plus le nombre de paramètres à régler est important. Dans le but de réduire le nombre de paramètres et d'améliorer également le TMBR, nous proposons de tester des nouvelles caractéristiques plus pertinentes liées à la forme et au fond.

## 4.4. ÉVALUATION DES CARACTÉRISTIQUES LIÉES Á LA FORME ET AU FOND

Dans cette section, nous tentons d'évaluer les performances des nouvelles méthodes d'extraction de caractéristiques liées à la forme du chiffre et l'information complémentaire représentée par le fond (§ 3.2.2.2 et § 3.2.2.3). Nous rappelons que l'extraction de ces caractéristiques se faisant par balayage d'une fenêtre de 1 pixel de largeur. La topologie des CHMMs est définie en tenant compte de la largeur de chaque chiffre. Une démarche expérimentale consiste à déterminer la largeur minimale du chiffre d'une classe (tableau 4.2) comme étant le nombre d'états de son modèle pour générer le modèle HMM (Cavalin, 06).

### 4.4.1. Caractéristiques liées à la forme

Dans cette expérience, les caractéristiques liées à la forme sont extraites à partir d'une fenêtre glissante de 1 pixel pour construire les GMMs. La topologie du HMM est décrite par un nombre d'états égal à la largeur minimal du chiffre.

Les résultats obtenus sur la base de test sont reportés dans le tableau 4.9 et montrent que les caractéristiques liées à la forme donnent une amélioration très importante comparativement aux caractéristiques classiques. En effet, nous remarquons que le TMBR s'améliore jusqu'à atteindre 84.90% (soit une amélioration de 6.34 %) avec une réduction importante des confusions. Ce qui montre que les caractéristiques liées à la forme offrent une bonne discrimination des chiffres manuscrits.

Classes de test (%)

| Classes de référence (%) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | **90.15** | 0 | 1 | 0.55 | 0.6 | 2.5 | 1.75 | 0 | 2.85 | 0.6 |
| 1 | 0 | **90.23** | 0.6 | 0 | 2.25 | 0 | 0.6 | 4.32 | 0 | 2 |
| 2 | 1.3 | 0.6 | **83.82** | 1.6 | 4.05 | 0.63 | 0.67 | 1.35 | 3.98 | 2 |
| 3 | 1.25 | 0 | 1.3 | **80.65** | 0.1 | 3.93 | 0 | 2.6 | 7.32 | 1.95 |
| 4 | 0 | 0.55 | 2.25 | 1.11 | **83.55** | 1.2 | 2 | 1.2 | 0.65 | 7.49 |
| 5 | 1.46 | 0 | 1.86 | 2.67 | 0.26 | **80.65** | 3.3 | 1.1 | 4.94 | 3.76 |
| 6 | 3.65 | 0.65 | 1.8 | 0.55 | 2 | 3.75 | **85.32** | 1.3 | 2.18 | 0.6 |
| 7 | 0 | 3.65 | 1.37 | 2 | 0.53 | 1.3 | 0.6 | **86.90** | 1.3 | 2.35 |
| 8 | 1.3 | 0 | 0.65 | 3.85 | 0.6 | 3.85 | 2.25 | 0.35 | **84.75** | 2.6 |
| 9 | 2.05 | 0 | 1.3 | 0.55 | 4.25 | 2 | 0.65 | 2.15 | 4 | **83.0** |

TMBR = **84.9 %**

**Tableau 4.9.** Matrice de confusion calculée sur les caractéristiques liées à la forme.

### 4.4.2. Caractéristiques liées au fond

De la même manière, la largeur de fenêtre est fixée par 1 pixel pour calculer les caractéristiques. La topologie de CHMM est également modifiée de sorte que le nombre d'état pour chaque modèle est égal à la valeur minimale de la largeur des chiffres de même classe. Les résultats obtenus sont reportés dans le tableau 4.10.

Classes de test (%)

| Classes de référence (%) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | **95.25** | 0 | 0.45 | 0 | 0 | 1.3 | 1.4 | 0 | 1.2 | 0.4 |
| 1 | 0 | **94.68** | 0.23 | 0 | 0 | 0 | 0.34 | 3.65 | 0 | 1.1 |
| 2 | 1.3 | 0.35 | **83.33** | 2.3 | 2.83 | 1.0 | 0.55 | 1.0 | 3.9 | 0.5 |
| 3 | 0.75 | 0 | 1.64 | **83.2** | 0.61 | 3.7 | 0.4 | 2.3 | 5.6 | 1.8 |
| 4 | 0 | 0.7 | 1.7 | 1.1 | **84.1** | 1.3 | 3.1 | 0.7 | 0.65 | 6.45 |
| 5 | 1.3 | 0 | 1.1 | 1.1 | 0.6 | **82.7** | 3.3 | 1.4 | 3.6 | 4.9 |
| 6 | 2.4 | 0.25 | 1.65 | 0.7 | 1.85 | 2.9 | **87.45** | 0.85 | 2.3 | 0.5 |
| 7 | 0 | 1.85 | 0.2 | 1.67 | 0.46 | 1.3 | 0.5 | **90.6** | 0.86 | 2.7 |
| 8 | 1.25 | 0 | 2.3 | 2.95 | 0.33 | 2.95 | 1.32 | 0.9 | **87.2** | 0.8 |
| 9 | 2.3 | 0.15 | 0.9 | 0.85 | 3.5 | 1.35 | 1.2 | 1.8 | 3.3 | **84.7** |

TMBR = **87.32 %**

**Tableau 4.10.** Matrice de confusion calculée sur les caractéristiques liées au fond.

En exploitant l'information du fond, nous pouvons constater que les différentes topologies des zones (demi-plan supérieur, demi-plan inférieur) permettent une discrimination des chiffres notamment entre les boucles et les fausses boucles. Ces configurations permettent de minimiser la confusion entre les chiffres corrélés, notamment entre {2, 3, 5, 8}, {4, 9} et {1, 7}. Les résultats obtenus sont 87,32%. Le plus fort taux de confusion (6,45%) vient des chiffres 4 et 9. Nous remarquons également que ce type de caractéristiques donne un résultat plus performant que les caractéristiques liées à la forme.

## 4.5. COMBINAISON DES CARACTERISTIQUES

Afin d'améliorer les performances du système de reconnaissance des chiffres manuscrits, nous proposons de combiner les différentes caractéristiques décrites précédemment. Comme les caractéristiques liées à la forme et au fond exigent une largeur de 1 pixel, nous devons retenir parmi les caractéristiques classiques que seulement pour lesquelles il est possible de les calculer. Pour cela, trois caractéristiques classiques sont retenues :

- densité ;
- centre de gravité ;
- nombre de transitions noir-blanc ou blanc-noir.

Ainsi, le vecteur caractéristique se compose :

- 3 composantes classiques (CC);
- 24 composantes liées à la forme (Cforme);
- 14 composantes liées au fond (Cfond).

Les différentes combinaisons testées comprennent :

- classiques et forme (un vecteur de 27 éléments) ;
- classiques et fond (un vecteur de 17 éléments) ;
- forme et fond (un vecteur de 38 éléments) ;
- toutes les caractéristiques (un vecteur de 41 éléments).

Le tableau 4.11 présente les résultats obtenus pour toutes les combinaisons testées.

| Classe | CC+Cforme | CC+Cfond | Cforme+Cfond | CC+ Cforme+Cfond |
|---|---|---|---|---|
| 0 | 90.35 | 95.77 | **96.44** | 96.44 |
| 1 | 91.18 | 95.35 | 96.13 | **96.18** |
| 2 | 83.94 | 83.54 | 84.87 | **84.96** |
| 3 | 81.36 | 83.60 | 86.15 | **86.32** |
| 4 | 83.87 | 84.18 | 84.63 | **84.64** |
| 5 | 81.15 | 82.72 | 84.26 | **84.29** |
| 6 | 85.75 | 87.73 | 88.04 | **88.27** |
| 7 | 86.98 | 90.83 | 92.33 | **92.46** |
| 8 | 84.82 | 87.25 | 88.16 | **88.25** |
| 9 | 83.16 | 84.91 | 85.65 | **85.66** |
| TMBR | 85.25 | 87.58 | 88.66 | **88.74** |

**Tableau 4.11.** TMBR calculé pour les différentes combinaisons.

### 4.5.1. Discussion

Les résultats portés dans le tableau 4.11 montrent que la combinaison de caractéristiques permet une amélioration importante du TMBR. Cependant, nous pouvons constater que les caractéristiques liées à la forme et au fond permettent une meilleure discrimination des chiffres comparativement aux caractéristiques classiques. En combinant les caractéristiques classiques à celles liées à la forme et au fond ne conduisent pas forcément à une amélioration importante du TMBR. Cela montre que l'ajout de caractéristiques classiques n'apport pas une amélioration importante du TMBR ce qui permet de conclure que ces caractéristiques ne sont pas complémentaires.

## 4.6. COMBINAISON DES CHMMs EN LIGNES ET EN COLONNES

La démarche adoptée dans la section précédente est fondée sur une fenêtre glissante verticalement de gauche vers la droite, colonne par colonne, ce qui permet d'extraire l'information des zones verticales. Dans cette section, nous évaluons un système de reconnaissance des chiffres avec les mêmes caractéristiques précédentes (41 éléments) basé sur une combinaison de deux classifieurs de type CHMMs. L'un a été évalué dans la section précédente (CHMMs en colonnes) et l'autre basé sur une fenêtre glissante horizontalement de haut vers le bas, qui découpe l'image de chiffres ligne par ligne, pour extraire l'information des zones horizontales (CHMMs en lignes). Le principe d'un CHMM en ligne est tout d'abord présenté dans la figure 4.3.

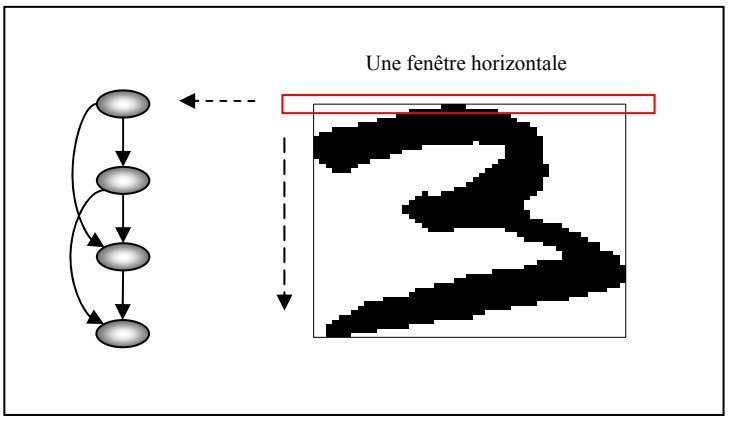



**Figure 4.3.** CHMM en lignes pour le chiffre 3.

Par conséquent, chaque classe de chiffre est représentée par deux CHMMs, un basé sur les colonnes et l'autre basé sur les lignes de l'image du chiffre. Les vecteurs caractéristiques extraits pour chaque colonne sont les mêmes dans la section précédente (41 éléments: 3 de CCs + 24 de Cforme + 14 de Cfond). Par contre, et après la vérification de plusieurs images de chiffre, nous définissons un maximum de 6 transitions par ligne. Ainsi, le vecteur caractéristique de la forme se compose, dans ce cas là, de 18 (3x6) caractéristiques. Ainsi, le vecteur caractéristique extrait pour chaque ligne de l'image de chiffre se compose de 35 éléments: (3 de CCs + 18 de Cforme + 14 de Cfond). De la même manière que les CHMMs en colonnes, nous définissons aussi le nombre d'états pour chaque CHMM en lignes par l'hauteur minimale de l'image de chiffres de chaque classe (voir tableau 4.2).

La phase d'apprentissage des modèles en colonnes et en lignes se fait de manière séparer sur la même base d'apprentissage (Figure 4.4)

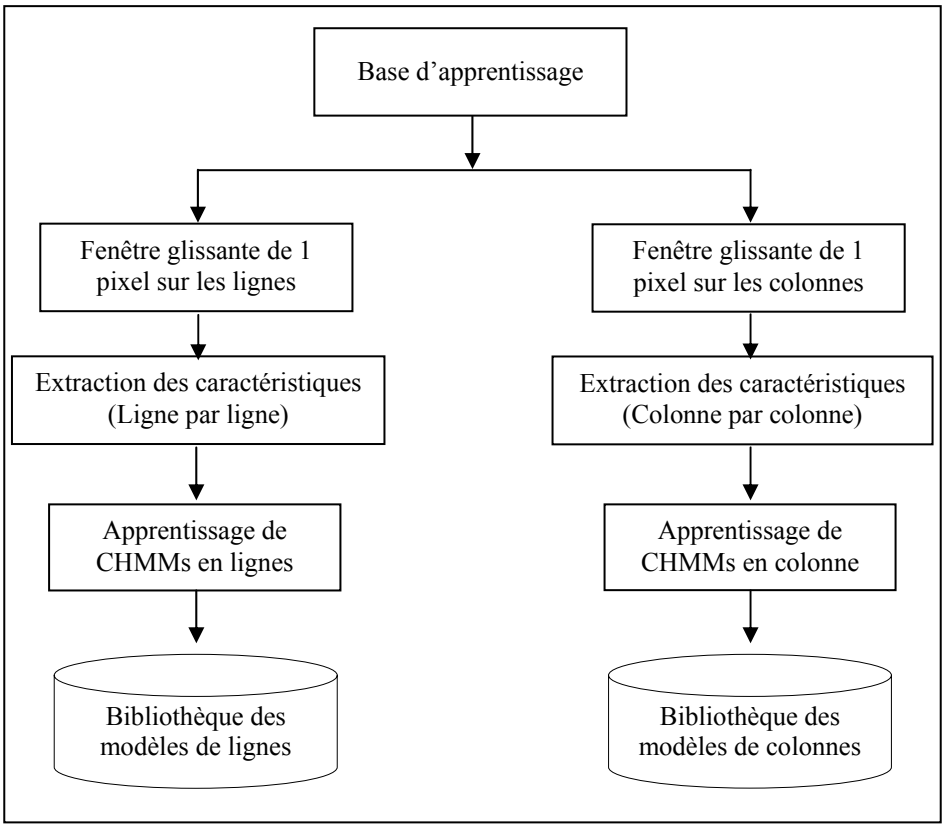



**Figure 4.4.** Apprentissage de CHMMs en lignes et en colonnes.

Dans la phase de reconnaissance, dix (10) CHMMs en lignes sont combinés avec les 10 CHMMs en colonne pour représenter plus précisément les 10 classes de chiffres.

La figure 4.5 illustre l'architecture globale de la reconnaissance des chiffres basée sur la combinaison des modèles en colonnes avec les modèles en lignes.

Dans cette combinaison, le modèle sélectionné est celui qui correspond la somme maximale de la probabilité d'appartenance d'une observation à une classe de lignes et la probabilité d'appartenance de même observation à une classe de colonne, selon la règle mathématique suivante :

$$\underset{1 \le i \le nbre\ de\ \text{mod}\grave{e}le}{\arg\max} \left[\log P(O^r / \lambda_i^{ligne}) + \log P(O^r / \lambda_i^{colonne})\right] \tag{4.3}$$

$$tel\ que\ \ r \in [1, R] \quad et \quad i = [1, 10]$$

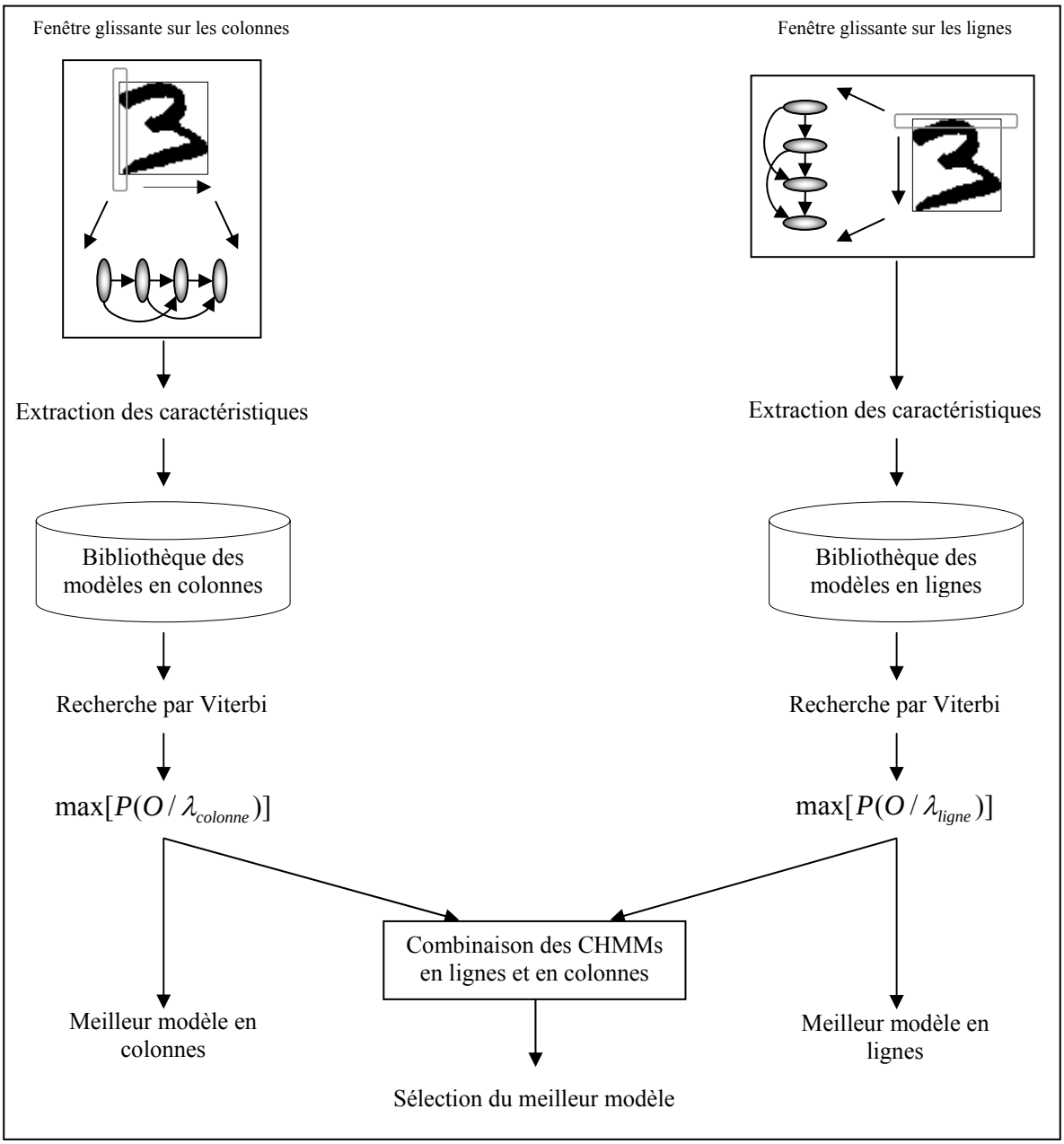



**Figure 4.5.** Reconnaissance par la combinaison des modèles en lignes et en colonnes.

Le tableau 4.12 montre les résultats obtenus lors la combinaison des modèles CHMMs en lignes et en colonnes.

| Type de modèle | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | TMBR |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CHMMs en colonnes | **96.44** | **96.18** | 84.96 | **86.32** | 84.64 | **84.29** | 88.27 | **92.46** | 88.25 | **85.66** | 88.74 |
| CHMMs en lignes | 96.25 | 95.46 | **85.20** | 85.77 | **84.68** | 84.14 | **88.52** | 91.13 | **88.28** | 84.89 | 88.44 |
| Combinaison de 2 modèles | 96.87 | 96.73 | 86.02 | 86.59 | 85.44 | 84.37 | 89.13 | 92.28 | 88.67 | 86.04 | 89.21 |

**Tableau 4.12.** TMBR calculé pour les différents modèles.

### 4.6.1. Discussion

Les résultats portés dans le tableau 4.12 montrent les taux de reconnaissance correspondants à chacun des trois tests. Nous avons remarqué que le système basé sur les CHMMs en lignes est moins performant pour la reconnaissance de chiffres comparativement aux CHMMs en colonnes, ce qui montre que la fenêtre glissante verticalement de gauche vers la droite est globalement plus représentative que la fenêtre horizontale. Mais, dans certain cas le balayage horizontal donne des résultats meilleurs que le balayage vertical comme le cas de chiffres 2, 4, 6, et 8. Cela est lié à la topologie géométrique des chiffres précédents, par exemple, le chiffre 8 possède deux boucles verticalement superposées, et les chiffres 6 et 9 possèdent respectivement une boucle supérieure et une autre inférieure. Ce qui démontre que le balayage horizontal donne aussi des trames assez significatives et discriminantes. Nous constatons que le meilleur TMBR est lors de la combinaison des CHMMs en lignes et en colonnes (89,21%). Les balayages horizontal et vertical permettent de bien caractériser la nature bidimensionnelle d'image de chiffres. Ainsi, nous pouvons conclure que les CHMMs en lignes sont des modèles complémentaires pour les CHMMs en colonnes.

## 4.7. CONCLUSION

L'objectif du chapitre a porté sur l'évaluation des différentes configurations du CHMM et les différentes caractéristiques pour assurer le meilleur taux de reconnaissance sans revenir à la normalisation de la taille des chiffres. Le travail élaboré a été consacré à l'évaluation des méthodes d'extraction des caractéristiques classiques, liées à la forme et au fond. Les expériences réalisées confirment l'importance et l'utilité de notre démarche pour extraire l'information la plus pertinente du chiffre. Plusieurs tests ont été faits pour bien optimiser les paramètres initiaux des CHMMs, tels que le nombre d'états, nombre de gaussiennes et la largeur de la fenêtre glissante. Les résultats montrent que ces tests permettent d'améliorer considérablement le TMBR. La combinaison de 3 types de caractéristiques implémentées donne une amélioration considérable dans les performances du système. Cette amélioration est due à l'information de la forme qui permet d'avoir une idée sur la distribution de pixels noirs et également le nombre et la position des transitions. Par ailleurs, l'information de concavité permet d'avoir une meilleure discrimination du chiffre caractérisé par des configurations topologiques (les boucles, les demi boucles et les fausses boucles), notamment entre les chiffres : {2,3,5,8}, {1,7}, {4,9} et {0,6}.

Une autre démarche a été proposée fondée sur la combinaison des CHMMs en lignes et en colonnes pour caractériser complètement le chiffre à deux dimensions. Le TMBR obtenu dépasse 89 % grâce à une règle de combinaison des modèles basée sur la sélection de la probabilité maximale de la somme de vraisemblances des modèles en ligne et en colonne.

# CONCLUSION GÉNÉRALE

L’objectif de ce travail est la mise en œuvre et l’évaluation de plusieurs méthodes d’extraction de caractéristiques pour la reconnaissance des chiffres manuscrits non normalisés en utilisant les modèles de Markov cachés.

Le travail présenté dans ce mémoire décrit les différentes étapes nécessaires à la construction d’un système de reconnaissance de chiffres manuscrits isolés. Pour chacune de ces étapes à savoir : les pré-traitements, l’extraction des caractéristiques et la classification.

Dans le domaine de la reconnaissance de chiffres manuscrits, les caractéristiques peuvent être décrites comme un moyen permettant de distinguer un caractère d’une classe d’un autre caractère d’une autre classe. Dès lors, il est nécessaire de définir des caractéristiques significatives lors du développement d’un système de reconnaissance. Les caractéristiques sont généralement définies par expérience ou par intuition. Plusieurs primitives peuvent être extraites. Nous avons tenté de proposer plusieurs méthodes d’extraction des caractéristiques pertinentes. De nombreuses études ont été proposées dans la littérature. Notre orientation s’est focalisée sur les caractéristiques classiques, liées à la forme et au fond. L’évaluation de ces caractéristiques a été faite par  une modélisation stochastique de type modèles de Markov cachés (HMMs).

Au cours de ce mémoire, nous avons rappelé les différentes techniques de reconnaissance de l'écriture manuscrite. Nous avons pu constater que la reconnaissance d'écriture a connu ces dernières années des progrès très importants, permettant désormais de faire face à la variabilité de l'écriture, et ainsi de reconnaître de manière satisfaisante des entités manuscrites isolées : mots et chiffres isolés. Dans ce travail, nous nous sommes focalisé sur la reconnaissance des chiffres manuscrits isolés par les HMMs continus

Ainsi, après avoir présenté les différents modules constituant notre système de reconnaissance, nous avons détaillé la phase d’extraction des caractéristiques ainsi que les différents avantages qui lui sont associés. Nous avons proposé d’utiliser des méthodes d’extraction de caractéristiques classiques, basé sur la forme et sur le fond. Ces caractéristiques présentant l’avantage d’être robustes au défaut d’encrage et à la variabilité d’image de chiffre d’entrée, et permettant d’obtenir des informations sur les transitions, la concavité et la distribution des pixels noirs sur toute la forme. Pour la modélisation Markovienne, l’idée générale est de créer un modèle de Markov reliant des états cachés à des caractéristiques extraites des images pour chaque classe de chiffres. Ces modèles étant utilisés lors de la phase de

reconnaissance pour déterminer la classe des différentes images de la base de test à partir de leurs caractéristiques. L'approche proposée consiste à découper les images par une fenêtre glissante de manière à obtenir des trames de même largeur pour extraire les caractéristiques.

Nous avons enfin présenté les résultats expérimentaux de ces approches. Nous avons exposé premièrement les problèmes liés à l'initialisation des paramètres de CHMMs. Ces problèmes concernent principalement le choix des paramètres qui doivent être très représentatifs de l'ensemble des exemples d'apprentissage. Ce choix est important car il conditionne toute la méthodologie mise en œuvre pour la reconnaissance. Nous avons fait plusieurs tests afin de trouver les paramètres optimaux du système, parmi lesquels, nous avons testé l'influence de la largeur de la fenêtre glissante, nous avons constaté que les fenêtres de 2 et 3 pixels peuvent contenir des trames assez complexes contenant différentes formes corrélées.

Un test a été fait sur l'influence de nombre d'états sur le TMBR, nous avons montré qu'il existe une relation proportionnelle entre le nombre d'états et la taille d'image des chiffres. Le troisième paramètre qui a été évalué est le nombre de gaussiennes, nous sommes arrivés à diminuer le nombre de gaussiennes tout en conservant un taux de reconnaissance acceptable (de 18 à 12 gaussiennes).

Les expériences élaborées sur les caractéristiques liées à la forme et au fond montrent que le fond et la forme du chiffres ne sont pas des informations redondantes, mais bien complémentaires puisque leur efficacité diffère selon la topologie de l'image des caractères. Ainsi, les informations du fond permettent de prendre une meilleure décision dans le cas des chiffres qui contiennent une boucle ou demi boucle. Leur combinaison donne une amélioration considérable de 84.9% à 88.66 et cette amélioration exprime leur complémentarité. Nous avons également évalué la combinaison de 3 types de caractéristiques afin de tester le pouvoir discriminant du système, dont le TMBR a augmenté à 88.74%.

Nous avons montré que le modèle de lignes d'un chiffre donne des informations complémentaires sur les zones horizontales de la forme, cela permet de diminuer la confusion entre les chiffres qui peuvent présenter une similarité dans les zones verticales, comme {3,8,2,5} et {6,4}. Nous avons amélioré le TMBR jusqu'à atteindre 89.21 % lors de la combinaison des modèles en colonnes et en lignes.

Au cours de ce travail, nous avons pu augmenter le taux de reconnaissance de 84.90 jusqu'à 89.21 %, ce qui traduit des performances satisfaisantes et ces expériences ont prouvé que les HMMs continus peuvent fournir un taux de reconnaissance satisfaisant.

Avant d'aborder les perspectives nouvelles, ce système est considéré comme une étape préliminaire pour réaliser un système de reconnaissance des chaînes de caractères (chiffres et mots cursifs). Dans cette approche, la phase la plus basique est la modélisation des caractères isolés par les HMMs sans faire la normalisation. Au-delà un certain nombre d'amélioration, quelques perspectives peuvent être envisagées :

- Remplacer la mixture de gaussiennes par un réseau de neurones où les machines à vecteurs de support (SVMs) pour meilleure modélisation des chiffres ;
- Etendre le système pour la reconnaissance d'une chaîne de chiffres ;
- Tester d'autres descripteurs multidirectionnels pour une meilleure caractérisation des chiffres.

***"Everything should be made as simple as possible, but not simpler"***

Albert Einstein

# Références Bibliographiques


**Al-Badr, B. et Mahmoud, S.A., 1995,** "Survey and bibliography of Arabic optical text recognition", *Signal processing*, vol. 41, Issue 1, pp. 49-77.

**Augustin, E., 2001,** "Reconnaissance de mots manuscrits par systèmes hybrides Réseaux de Neurones et Modèles de Markov Cachés", *Thèse de Doctorat*, Paris V, 188 pages.

**Ayat, N.E., Cheriet, M. et Suen, C.Y., 2005,** "Automatic model selection for the optimization of SVM kernels", *Pattern Recognition*, vol. 38, no. 10, pp. 1733-1745.

**Britto, A., Sabourin, R., Lethelier, E., Bortolozzi, F. et Suen, C., 2000**, "Improvement handwritten numeral string recognition by slant normalization and contextual information", *International Workshop on Frontiers in Handwriting Recognition*, pp 323-332.

**Britto, A., Sabourin, R., Bortolozzi, F. et Suen, C., 2003,** "Recognition of handwritten numeral strings using a two-stage HMM Based method", *International Journal on Document Analysis and Recognition*, vol. 5, pp. 102-117.

**Britto, A., Sabourin, R., Bortolozzi, F. et Suen, C., 2004,** "Foreground and Background Information in an HMM-based Method for Recognition of Isolated Characters and Numeral Strings", *International Workshop on Frontiers in Handwriting Recognition*. pp. 371-376.

**Cavalin, P. R., Britto, A., Bortolozzi, F., Sabourin, R. et Oliveira, L., 2006,** "An implicit segmentation-based method for recognition of handwritten strings of characters", *ACM symposium on Applied computing*, pp. 836-840.

**Chatelain, C., 2006,** "Extraction de séquences numériques dans des documents manuscrits quelconques", Thèse de Doctorat, Université de Rouen, 196 pages.

**Chen, Y. et Wang, J.F., 2000,** "Segmentation of Single- or Multiple-Touching Handwritten Numeral String Using Background and Foreground Analysis". *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 22, no. 11, pp. 1304-1317.

**Cheriet, M., Kharma, N., Liu, C. et Suen, C., 2007,** "Character Recognition Systems: A Guide for Students and Practitioners", *Wiley-Interscience*, 361 pages.

**Connell, S.D. et Jain, A.K., 2002,** "Writer Adaptation for Online Handwriting Recognition", *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 24, no. 3, pp. 329-346.

**Dargenton, P., 1994,** "Contribution à la segmentation et la reconnaissance de l'écriture manuscrite". *Thèse de Doctorat*. Institut National des Sciences Appliquées de Lyon, 174 pages.

**Devijver, P.A. et Kittler, J., 1982,** "Pattern recognition, a statistical approach", *Prentice Hall*, London, 480 pages.

**El-Yacoubi, A., Gilloux, M., Sabourin, R. et Suen, C. Y., 1999,** "An HMM Based Approach for Off-line Unconstrained Handwritten Word Modelling and Recognition", *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 21, no. 8, pp. 752-760.

**Gattal, A. J., 2009,** "Segmentation automatique pour la reconnaissance numérique des chèques bancaires Algériens". *Mémoire de magister*, C. U. de Khenchla, Algeria, 89 pages.

**Gilloux, M., Lemari, B. et Leroux, M., 1995,** "Hybrid radial basis functions Metwork/Midden Markov Model handwritten word recognition system", *International Conference on Document Analysis and Recognition*, vol. 1, pp. 394-397.

**Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H. et Schmidhuber, J., 2009,** "A Novel Connectionist System for Unconstrained Handwriting Recognition". *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 31, no. 5, pp. 855-868.

**Impedovo, S., Wang, P. S. P. et Bunke, H., 1997,** "Automatic bankcheck processing", *Machine Perception and Artificial Intelligence*, vol. 28, World scientific publishing company.

**Jain, A.K., Duin, R.P.W. et Mao, J., 2000,** "Statistical pattern recognition: A review", *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 22, no. 1, pp. 4-37.

**Juang, B. H. et Rabiner, L. R., 1991,** "Hidden Markov Models for Speech Recognition", *Technometrics*, vol. 33, no. 3, pp. 251-272.

**Kimura, F., Tsuruoka, S., Miyake, Y. et Shridhar, M., 1994,** "A Lexicon Directed Algorithm for Recognition of Unconstrained Handwritten Words", *IEICE Trans. on Information & Systems*, vol. E77-D, no. 7, pp. 785-793.

**Koch, G., Heutte, L. et Paquet, T., 2004,** "Combination of Contextual Information for Handwritten Word Recognition", *International Workshop on Frontiers in Handwriting Recognition*, pp. 468-473.

**Koerich, A.L., Sabourin, R. et Suen, C.Y., 2003,** "Large vocabulary off-line handwriting recognition: A survey", *Pattern Analysis and Applications*, vol. 6, pp. 97-121.

**Knerr, S., Asimov, V., Baret, O., Gorsky, N. et Simon, J.C., 1997,** "The A2IA inter-cheque system: Courtesy amount and legal amount recognition for French checks", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 11, no. 4, pp. 505-547.

**Lei, Y., Liu, C. S., Ding, X.Q. et Fu, Q., 2004,** "A Recognition Based System for Segmentation of Touching Handwritten Numeral Strings", *International Workshop on Frontiers in Handwriting Recognition*, pp. 294-299.

**Liu, C.L., Nakashima, K., Sako, H. et Fujisawa, H., 2002,** "Handwritten Digit Recognition Using State-of-the-Art Techniques", *International Workshop on Frontiers in Handwriting Recognition*, pp. 320-325.

**Liu, C. L., Sako, H. et Fujisawa, H., 2004,** "Effects of Classifier Structures and Training Regimes on Integrated Segmentation and Recognition of Handwritten Numeral Strings", *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 26, no. 11, pp. 1395-1407.

**Mohammed, M. et Gader, P., 1996,** "Handwritten Word Recognition Using Segmentation-Free Hidden Markov Modeling and Segmentation-Based Dynamic Programming Techniques", *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 18, no. 5, pp. 548-554.

**Mohamad, R.A., Likforman-Sulem, L. et Mokbel, C., 2009,** "Combining Slanted-Frame Classifiers for Improved HMM-Based Arabic Handwriting Recognition", *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 31, no. 7, pp. 1165-1177.

**Morita, M., Sabourin, R., Bortolozzi, F. et Suen, C. Y., 2002,** "Automatic recognition of handwritten numerical strings: A recognition and verification strategy", *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol.24, no. 11, pp. 1438–1454.

**Morita, M., Sabourin, R., Bortolozzi, F. et Suen, C. Y., 2006,** "Segmentation and Recognition of Handwritten Dates: An HMM-MLP hybrid approach", *International Journal on Document Analysis and Recognition*, vol. 6, no. 4, pp. 248-262.

**Milewski, R. et Govindaraju, V., 2006,** "Extraction of Handwritten Text from Carbon Copy Medical Form Images", *Document Analysis Systems*, vol. 3872, pp. 106-116.

**Nemmour, H**., **2003**, "Méthodes de détection de changement dans les images optiques de télédétection". *Mémoire de Magister : Université des Sciences et de la Technologie Houari Boumediene USTHB*, 125 pages.

**Nemmour, H. et Chibani, Y., 2006,** "Multi-Class SVMs Based on Fuzzy Integral Mixture for Handwritten Digit Recognition". *Proceedings of the Geometric Modeling and Imaging*, pp. 145-149

**Oliveira, L.E., Sabourin, R., Bortolozzi, F. et Suen, C. Y., 2002,** "Automatic Recognition of Handwritten Numeral Strings: A Recognition and Verification Strategy", *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 24, no. 11, pp. 1438-1454.

**Oliveira, L.S., 2003,** "Automatic recognition of handwritten numerical strings", *Thèse de Doctorat*. Ecole de technologie superieure, University of Quebec. Canada, 184 pages.

**Oliveira, L.E., Morita, M. et Sabourin, R., 2006,** "Feature Selection for Ensembles Applied to Handwriting Recognition", *International Journal on Document Analysis and Recognition*, vol. 8, no. 4, pp. 262-279.

**Pfister, M., Behnke, S. et Rojas, R., 2000,** "Recognition of Handwritten ZIP Codes in a Real-World Non-Standard- Letter Sorting System", *Applied Intelligence*, vol. 12, no. 1-2, pp. 95-115.

**Poisson, E., 2005,** "Architecture et Apprentissage d'un Système Hybride Neuro-Markovien pour la Reconnaissance de l'Ecriture Manuscrite En-Ligne", *Thèse de Doctorat*, Université de Nantes, 213 pages.

**Rabiner, L. R., 1989,** "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286.

**Ramdane, S., Taconet, B. et Zahour, A., 2003,** "Classification of forms with handwritten fields by planar hidden Markov models", *Pattern Recognition*, vol. 36, no. 4, pp. 1045-1060.

**Sadri, J., Suen, C. Y. et Bui, T. D., 2004,** "Automatic Segmentation of Unconstrained Handwritten Numeral Strings", *International Workshop on Frontiers in Handwriting Recognition*, pp. 317-322.

**Senior, A.W. et Robinson, A. J., 1998,** "An Off-Line Cursive Handwriting Recognition System", *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 20, no. 3, pp. 309-321.

**Tappert, C. C., Suen, C. Y. et Wakahara, T., 1990,** "The state of the art in on-line handwriting recognition", *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 12, Issus 8, pp. 787-808.

**Tay, Y. H., Lallican, P. M., Khalid, M., Viard-Gaudin, C. et Knerr, S., 2003,** "Offline handwritten word recognition using a hybrid Neural network and hidden Markov model", *IEEE International Symposium on · Computational Intelligence in Robotics and Automation*, vol. 3, pp. 1190-1195.

**Trier, O. et Jain, A. K., 1995,** "Goal-Directed Evaluation of Binarization Methods". *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 17, no. 12, pp. 1191-1201.

**TriDO, T. M. et Artières, T., 2007,** "Apprentissage de Mélanges de Gaussiens par Maximisation de la marge avec SMO", *Conférence de francophone sur l'Apprentissage automatique*.

**Vapnik, V., 1998,** "Statistical Learning Theory", *Wiley-Interscience Publication*, 740 pages.

**Xu, Q., Lam, L. et Suen, C. Y., 2003,** "Automatic segmentation and recognition system for handwritten dates on Canadian bank cheques", *International Conference on Document Analysis and Recognition*, pp. 704-708.

**Xue, H. et Govindaraju, V., 2006,** "Hidden Markov Models Combining Discrete Symbols and Continuous Attributes in Handwriting Recognition", *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 28, no. 3, pp. 458-462.

**Zhang, G.P., 2000,** "Neural Networks for Classification: A Survey", *IEEE Tranactions on Systems, Man, and Cybernetics - Part C*, vol. 30, no. 4, pp. 641-662.

**Zhang, L.Q. et Suen, C. Y., 2002,** "Recognition of courtesy amounts on bank checks based on a segmentation approach", *Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition*, pp.298-302.