

Résumé

Dans cette thèse on présente une nouvelle technique OCR de reconnaissance des caractères arabe manuscrits par utilisation des réseaux de neurones. En exploitant des caractéristiques couramment utilisées, et en procédant à une nouvelle manière de programmation de ces réseaux notre système présente une bien meilleure aptitude à reconnaître ces caractères.

Un taux de reconnaissance plus élevé peut être obtenu par l'augmentation du nombre de caractéristiques globales des caractères. Pour cela une nouvelle technique est mise en œuvre. Cette technique considère les caractéristiques de chaque région de l'image du caractère plutôt que les caractéristiques de toute l'image. Un découpage est alors opéré sur l'image selon des directions bien spécifiques. Pour chacune des régions, résultat du découpage, le vecteur des caractéristiques est calculé. Un vecteur total rassemblant l'ensemble des caractéristiques ainsi obtenu remplacera alors l'image dans l'espace des caractéristiques. Il s'agit en fait de pratiquer un zoom dans le corps de l'image du caractère. Les premiers résultats, obtenus sur un échantillon restreint, sont plutôt encourageant : 15~20% d'amélioration en

pourcentage de reconnaissance. Reste à élargir la base d'apprentissage pour conclure quant à la validité du processus.

TABLE DES MATIERES

INTRODUCTION GENERALE.....	1
CHAPITRE 1 : LES DIFFERENTS ASPECTS DE LA RECONNAISSANCE OPTIQUE DE L'ECRITURE	9
1-1 Modes de saisie de l'écriture.....	9
1-1-1 Le mode statique.....	9
1-1-2 Le mode continu.....	10
1-2 Les différents aspects de la reconnaissance optique de l'écriture	10
1-2-1 La reconnaissance en-ligne (on-line) ou dynamique.....	11
1-2-2 La reconnaissance hors-ligne (off-line) ou statique.....	12
1-2-3 Reconnaissance de l'imprimé ou du manuscrit.....	12
1-2-4 Reconnaissance monospace, multicosse, omnicosse	12
1-2-5 Reconnaissance de caractères ou analyse de documents	13
1-3 Techniques de prétraitement.....	13
1-3-1 Lissage.....	14
1-3-1-1 Le redressement de la ligne de base.....	14
1-3-1-2 Le redressement des écritures penchées.....	14
1-3-2 La squelettisation.....	15
1-3-3 La normalisation	15
1-4 Techniques de reconnaissance.....	15
1-4-1 Reconnaissance de caractères isolés.....	16
1-4-2 Reconnaissance de mots.....	16
CHAPITRE 2 : PRETRAITEMENT POUR LA RECONNAISSANCE... 18	

2-1 INTRODUCTION.....	18
2-2 Binarisation.....	18
2-2-1 Seuillage.....	18
2-2-2 Seuillage global.....	18
2-2-3 Seuillage adaptatif.....	19
2-3 Elimination du bruit (Filtrage)	20
2-3-1 Types de filtrages.....	20
2-3-1-1 Les filtres linéaires.....	20
2-3-1-2 Les filtres non linéaires.....	21
2-3-2 Les filtres utilisés.....	21
2-3-2-1 Filtre passe-bas	21
2-3-2-2 Filtre passe-haut.....	22
2-3-2-3 Filtre Moyeneur.....	23
2-3-2-4 Filtre Médian.....	24
2-4 Normalisation.....	25
2-4-1 Correction de l'inclinaison des lignes.....	25
2-4-2 Correction de l'inclinaison des lettres.....	26
2-4-3 Normalisation de l'Echelle (Ré échantillonnage)	27
2-5 Les techniques de changement de représentation.....	29
2-5-1 Extraction des composantes connexes.....	29
2-5-2 Extraction du contour.....	32
2-5-2-1 La vectorisation.....	32
2-5-2-2 La transformation globale du contour.....	33
2-5-2-3 L'approximation du contour : représentation en lignes raccordées (Fitting line segment)	34
2-6 Squelettisation.....	35
 CHAPITRE 3 : EXTRACTION DES PRIMITIVES.....	 40
3-1 Introduction.....	40

3-2 Moments invariants de Hu	40
3-2-1 Application de la méthode des moments pour la reconnaissance.....	43
3-2-2 Découpage de l'image du caractère	46
3-2-2-1 Découpage horizontal	47
3-2-2-2 Découpage vertical	49
3-2-2-3 Découpage horizontal et vertical	50
3-3 Les moments de Zernike	52
3-3-1 Introduction	52
3-3-2 Moments complexes de Zernike	53
3-3-3 Application de la méthode des moments de Zernike sur une image	55
3-4 Descripteurs de Fourier.....	59
3-4-1 Calcul des descripteurs de Fourier	61
3-4-2 Propriété des descripteurs de Fourier	63
3-4-2-1 La translation.....	63
3-4-2-2 La rotation.....	64
3-4-2-3 Changement d'échelle.....	64
3-4-2-4 Choix du point de départ z_0.....	66
3-5 Caractérisation du contour par le Code de Freeman.....	68
3-6 Caractérisation par squelettisation.....	68
3-7 La représentation du contour par une Chaîne de code (chain code)	70
3-8 La représentation en lignes raccordées (Fitting line segment)	71
3-9 les moments affins invariants (AMI)	72

CHAPITRE 4 : ETAT DE L'ART DE LA RECONNAISSANCE.....75

4-1	Caractéristiques de la langue arabe	75
4-2	Etat de l'art de la reconnaissance	80
4-2-1	Méthodes structurelles	80
4-2-2	Méthodes topologiques	81
4-2-3	Les primitives statistiques	81
4-2-3-1	Caractéristique spectral	84

CHAPITRE 5 : CONTRIBUTION A LA RECONNAISSANCE

DE L'ECRITURE ARABE MANUSCRITE

5-1	Reconnaissance des caractères arabes	
	manuscrits isolés	85
5-1-1	Introduction	85
5-1-2	Réseaux de neurones et classification	85
5-1-3	Application	86
5-1-3-1	Extraction des primitives (caractéristiques)	86
5-1-4	Préparation du réseau	86
5-1-4-1	Structure du réseau	86
5-1-4-2	Phase d'apprentissage	88
5-1-4-3	Activation des sorties	90
5-1-4-4	Fichier de reconnaissance	90
5-1-4-5	Phase de reconnaissance	91
5-1-4-6	Résultats et discussions	92
5-1-5	Méthode de découpage de l'image	93
5-2	Reconnaissance des caractères arabes manuscrits	
	à partir d'un texte	95
5-2-1	Introduction	95
5-2-2	Reconnaissance des caractères arabes manuscrits.....	96
5-2-3	Segmentation du texte en lignes	97
5-2-3-1	Récupération des lignes	97

5-2-3-2 Détection de la ligne de base	97
5-2-4 Segmentation des lignes en mots	98
5-2-5 Décomposition du mot en ses composantes	99
5-2-6 Processus de segmentation	104
CHAPITRE 6 : SIMULATION ET RESULTATS.....	109
6-1 Introduction	109
6-2 Architecture du réseau de neurone utilisé	110
6-2-1 La couche d'entrée	110
6-2-2 La couche cachée	110
6-2-3 La couche de sortie	111
6-3 Résultat et discussion	111
6-3-1 Taux de Reconnaissance des caractères arabes manuscrits isolés	111
6-3-1-1 Comparaison entre les deux méthodes.....	113
6-3-1-2 Comparaison entre les différents types de découpage....	114
6-3-1-3 Conclusion	118
6-3-2 Reconnaissance des mots arabes manuscrits	118
Conclusion et Perspectives.....	120
Annexe A : Interface utilisateur.....	122
Annexe B : Organigrammes	128
Annexe C :	135
Références Bibliographiques	136

INTRODUCTION GÉNÉRALE

Les techniques liées au traitement de l'information connaissent actuellement un développement très actif en liaison avec l'informatique et présentent un potentiel de plus en plus important dans le domaine de l'interaction Homme-machine. L'Homme veut communiquer avec l'ordinateur de la façon la plus simple, la plus naturelle pour faciliter et accélérer l'interaction et l'échange d'informations. Il cherche à rendre ces machines accessibles par la voix, capables de lire, de voir, de se déplacer...et de traiter et d'analyser rapidement l'information reçue.

Ecrire pour communiquer a été de tous les temps une préoccupation première de l'Homme. L'écrit a été, et restera, l'un des grands fondements des civilisations et le mode par excellence de conservation et de transmission du savoir. Malgré les avancées d'autres moyens de communication tel que l'audio visuel, nombreuses sont les applications dont l'existence commence sur le papier, plus particulièrement dans la bureautique (saisie, indexation et archivage automatique de documents), en publication assistée par ordinateur (pour faciliter la composition à partir d'une sélection de plusieurs documents), dans la poste (lecture des adresses et tri automatique), dans les banques (traitement des chèques, des factures)... Cependant malgré les progrès technologiques, le clavier reste encore un moyen obligé de communication avec l'ordinateur.

Contrairement au latin, la reconnaissance de l'écriture arabe manuscrite ou imprimée reste encore aujourd'hui au niveau de la recherche et de l'expérimentation, le problème n'est pas encore résolu bien que l'on sache atteindre des taux assez élevés dans certaines applications pour lesquelles soit le vocabulaire est limité [Ema 90], soit la fonte est unique [Badr 95]. Les travaux sont généralement axés sur la méthodologie du développement plutôt que sur la réalisation d'un produit fini vendable. Une version commercialisable reste encore au stade du rêve, les efforts doivent se multiplier pour le réaliser. Il est vrai que l'écriture arabe présente certaines caractéristiques qui sont à l'origine de la complexité du traitement, particulièrement parce qu'elle est cursive aussi bien dans sa forme imprimée que manuscrite; de plus les caractères changent de forme selon la position dans le mot, et même dans certains cas, selon sa phonétique. D'autre part, le caractère arabe présente une forme cursive voyellée nécessitant, pour la

majorité des lettres, des matrices de dimensions plus grandes que celles communément utilisées pour le latin, ce qui laisse jusqu'à présent les formes informatisées des caractères arabes non encore normalisées.

La reconnaissance de l'écriture arabe date des années 80 [Ami 85] [Saad 85]. Depuis, les recherches se sont multipliées dans ce domaine. Certains chercheurs se sont intéressés à la reconnaissance en temps réel [Alimi 96] en utilisant des tablettes graphiques, ce qui simplifie en partie le problème en restituant le sens du tracé, d'autres se sont penchés sur l'imprimé et/ou le manuscrit en "Off-line" en utilisant un scanner ou une caméra pour la saisie des documents. Depuis, plusieurs outils de prétraitement ont été développés pour la squelettisation de l'image, de son lissage, pour la détermination du contour et l'extraction de primitives...et différentes techniques de reconnaissance ont été élaborées, parmi lesquelles figurent des méthodes statistiques, structurelles ou géométriques, avec ou sans segmentation en caractères [Dabi 90] [Romeo 95]. Généralement toutes ces méthodes et bien d'autres, tendent à extraire, chacune à sa façon, une catégorie de caractéristiques et d'évaluer par la suite la vraisemblance entre les primitives extraites et celles de formes prototypes déjà apprises par le système de reconnaissance.

Avec la reconnaissance de l'écriture se produit une autre révolution, intellectuelle plus que technologique. L'expression de "reconnaissance de l'écriture" désigne le fait que les ordinateurs sont devenus progressivement capables d'identifier des caractères écrits, manuscrits et imprimés, de les "reconnaître" et de les transformer immédiatement en telle ou telle représentation numérisée selon les codes utilisés. Les tout premiers systèmes de reconnaissance de caractères par des procédés de lecture magnétique ou optique ont été mis au point entre 1950 et 1960. C'étaient des systèmes de saisie indirecte, qui exigeaient que les caractères à identifier ont été soit imprimés ou dactylographiés auparavant à l'aide d'encre magnétisable soit convertis au préalable en codes à chiffres ou à barres. Ces systèmes de lecture optique sont devenus directs entre 1960 et 1970, mais nécessitaient toujours que l'on eût recours à des normes de transcriptions dactylographiques ou typographiques particulières, spécialement conçues à l'époque pour faciliter ces modes de saisie nouveaux. À partir de 1980, l'apparition de langages informatiques inédits de description de caractères permit d'assouplir

l'utilisation de ces systèmes d'identification. Dès 1985, la plupart des polices de caractères alphabétiques étaient susceptibles d'être ainsi décrites et identifiées et, dès 1987, les polices de caractères idéographiques japonaises ou chinoises l'étaient à leur tour. La reconnaissance de l'écriture manuscrite a accompli de même des progrès spectaculaires, en particulier depuis 1980, jusqu'à permettre sa saisie dans l'instant même de sa transcription par le biais de tablettes de téléécriture qui se confondent parfois avec l'écran même de visualisation. Ces systèmes ont commencé à se répandre depuis 1988. Ils permettent désormais d'effectuer à volonté la synthèse de l'écriture manuscrite de tout un chacun. L'apparition également, depuis 1988, des premiers systèmes de traitement de texte (et donc d'écriture) à entrée vocale a enfin achevé de bouleverser les frontières admises entre l'écrit et l'oral. Ces systèmes permettent de produire en effet directement des signes écrits, des mots et des phrases, à partir de leur énoncé oral. En ce domaine, les progrès ont été lents pourtant. Les premiers systèmes de reconnaissance des mots qui aient été réellement opérants remontent au début des années 1970. En 1982, ces systèmes étaient déjà devenus capables de reconnaître entre une centaine et un millier de mots prononcés en différentes langues, de l'anglais au japonais en passant par le russe, l'espagnol, l'allemand ou l'italien. Ils réussissaient à identifier entre 3 000 et 5 000 mots en 1985, plus de 20 000 mots en 1988. Ces chiffres et ces dates donnent une idée de la rapidité des progrès récents de ces systèmes. L'ambition proclamée de ces recherches est immense puisqu'il s'agit de rendre les ordinateurs capables de "comprendre" le langage humain, de le parler et de l'écrire du moins, et cela dès la décennie 1990-2000. En cette fin du XXe siècle, ce sont ainsi la place, le statut et le rôle que tous les systèmes d'écriture ont pu acquérir lentement au cours de ces cinq ou six derniers millénaires qui se trouvent sur le point d'être radicalement bouleversés. Cette conquête de l'écriture par l'informatique aura été extraordinairement rapide à l'échelle de l'histoire de l'humanité. De 1950 à 1985, les ordinateurs se seront en effet substitués, en moins de trente cinq ans, à pratiquement tous les procédés antérieurs connus de transcription, de transmission, de communication, de conservation ou de transformation d'un message écrit ou imprimé. De surcroît, en ramenant par des biais successifs tous les systèmes d'écriture concevables à un système unique de représentation à l'aide de deux chiffres, de deux

caractères numériques, le "zéro" et le "un" seulement, la numérisation de l'écriture a brusquement révélé l'existence d'une espèce d'unicité des modes de transcription des langages humains qu'on ne faisait que soupçonner auparavant. En multipliant également les techniques de restitution, de régénération et de télécommunication ou de télédiffusion, l'informatique en aura aussi accentué la multiplicité. En ce sens, l'informatisation de ces systèmes d'écriture recèle autant de promesses d'éclatement que de perspectives de fusion vers un système de transcription unique, universel. Dès l'instant où ils ont commencé à être informatisés et numérisés, ces systèmes d'écriture ont cessé d'être des réalités stables, permanentes et figées comme on le concevait jadis. Ils sont devenus des entités instables, éphémères et mouvantes, sujettes à d'infinies métamorphoses par le fait même de leur médiatisation et de leur numérisation. Les frontières avec ce qui était purement oral auparavant se déplacent également. C'est le statut même de ce qui était écrit, imprimé, lu, vu, entendu ou dit jusqu'à présent qui se déplace.

Grace au système de reconnaissance optique de caractères (OCR) un ordinateur peut, depuis plusieurs années, reconnaître un caractère comme image et de le transformée en code ASCII. Alors que cette reconnaissance est de plus en plus performante pour l'écriture imprimée, il n'en est pas de même pour l'écriture manuscrite. La reconnaissance de l'écriture manuscrite est actuellement un domaine très actif de recherche, où l'on distingue deux grands secteurs : la reconnaissance *en ligne* (on-line) et hors ligne (off-line). C'est une discipline de l'intelligence artificielle et ses applications sont loin d'être limitées. La reconnaissance automatique des codes postaux pour l'orientation et le trie du courrier, la reconnaissance automatique des montants des chèques bancaires, la validation des chèques par une reconnaissance automatique des signatures, et la reconnaissance des personnes eux mêmes sont des applications parmi tant d'autres. On assiste ainsi à la sortie sur le marché d'un ensemble varié de système de reconnaissance plus ou moins efficace. Ces systèmes traitent cependant les langues à caractères latins. La langue arabe reconnaît dans toutes ces nouvelles techniques un développement assez timide pour ne pas dire rare.

Ces techniques de reconnaissance traitent l'écriture comme un objet graphique. Une chaîne composée d' :

- Un élément d'acquisition de l'image du document comprenant le texte (papier...) : il peut s'agir d'une caméra d'acquisition ou d'un scanner ;
- Un ordinateur (PC) pour la récupération du fichier graphique correspondant.

Est nécessaire pour la récupération du fichier graphique correspondant au texte à traiter. A ce document graphique on applique un ensemble d'algorithmes de traitement pour lui extraire le texte qu'il contient.

La figure ci-dessous résume tout le chemin d'un procédé OCR (Optical character recognition).

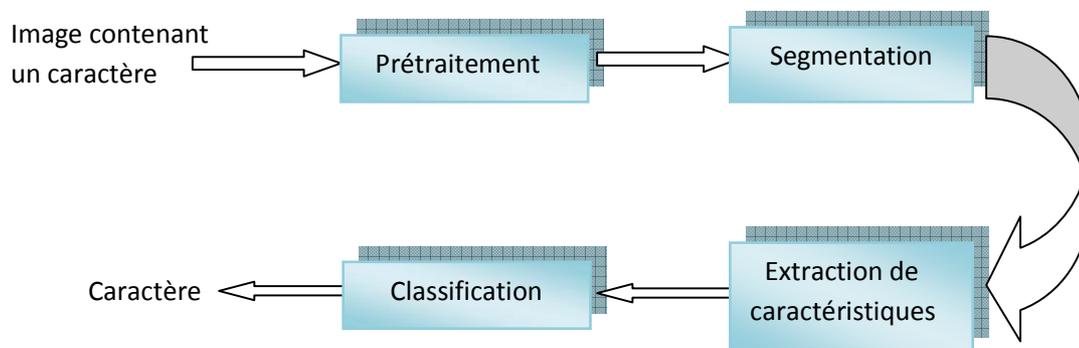


Figure 1.1 Le modèle générique d'un SRE

A la tête de ce chemin on trouve ***l'image du caractère***. C'est une image acquise par un scanner ou obtenue par un quelconque moyen de numérisation.

Vient ensuite le prétraitement. Il s'agit de l'ensemble des techniques qui sont appliquées à cette image pour la préparer à la suite du traitement. Il peut s'agir d'un filtrage, d'une correction d'inclinaison (redressement ou tout autre opération).

La suite est une opération de décomposition d'un texte en ses constituants. C'est le plus bas niveau qui permet par la suite une reconnaissance de tout type de mot en se basant sur la reconnaissance des caractères.

L'extraction des primitives qui suit, ramène le caractère dans un autre repère. C'est dans ce repère que s'opère la classification et c'est le choix de ce repère même qui décide la réussite ou non de l'opération de séparation entre les différents caractères.

La classification n'est autre que la concrétisation de l'utilisation des caractéristiques déjà calculé. A ce stade le caractère est placé dans une position parmi toutes celles affectées à tous les caractères.

Toutes ces opérations sont extrêmement liées les unes avec les autres. La réussite de toute la chaîne est conditionnée par une réussite de chacune de ses parties. Si par exemple l'opération de filtrage est mal menée, on ne peut espérer trouver un classement correct en fin de chaîne. Si le filtrage est bien traité et que le calcul des caractéristiques n'est pas correct on ne peut non plus espérer un classement juste. Il va sans dire donc que pour réussir toute la chaîne une bonne maîtrise de chacun de ses éléments est impérative.

L'idée de ce travail de thèse est le développement d'un système de reconnaissance de l'écriture arabe manuscrite basé sur les derniers développements scientifiques en matière d'algorithmique. Les algorithmes de reconnaissances qui ont connus un développement spectaculaire ces dernières années vont être ainsi appliqués à l'écriture arabe avec toutes ses particularités.

Les développements logiciels qui accompagneront ce travail vont être faits en langage C et en MATLAB. A part l'étape d'acquisition, le logiciel développé devrait permettre la reconnaissance d'un texte arabe manuscrit à partir de son document présenté comme fichier graphique en entrée. Dans cette thèse, le détail de chacune de ces opérations sera donné.

Cette thèse est organisée, en cinq chapitres, comme suit :

Le premier chapitre traite « **Les différents aspects de la reconnaissance optique de l'écriture** ». On y trouve : La reconnaissance en-ligne, La reconnaissance hors-ligne, Reconnaissance de l'imprimé ou du manuscrit, Reconnaissance monofonte, multifonte, omnifonte, Reconnaissance de caractères ou analyse de documents, Modes de saisie de l'écriture, Techniques de prétraitement, Techniques de reconnaissance et Techniques d'analyse. Ces différents paragraphes sont clairement exposés et

constituent une bonne base pour un départ en toute connaissance de cause dans ce domaine de recherche.

Dans le deuxième chapitre « **Prétraitement pour la reconnaissance** », se sont les différentes étapes qui prépare une image à la phase de reconnaissance qui sont exposés. Il s'agit de : La Binarisation, La Normalisation, L'Extraction des composantes connexes, L'Extraction du contour et L'Extraction du squelette. L'explication détaillée de ces différentes techniques permet de sortir avec des idées claires sur pourquoi et comment opérer telle ou telle opération.

L'extraction des primitives c'est-à-dire le vecteur de caractéristiques qui permet la distinction entre différentes formes a fait l'objet du chapitre trois « **Extraction des primitives** ». Ce n'est pas l'ensemble de ces méthodes d'extraction qui est exposé. On s'est limité au détail de certaines d'entre eux il s'agit de : La méthode des moments invariants, la méthode des moments de Zernike, La méthode des descripteurs de Fourier et la méthode des moments affine. Le détail de toute la formulation de ces méthodes est donné dans ce chapitre. Il signale pour chaque méthode la longueur du vecteur des caractéristiques et son importance dans les phases d'apprentissage et de reconnaissance.

Le quatrième chapitre intitulé « **Etat de l'Art** », fait, en premier lieu, une description détaillée de l'écriture arabe. Il s'agit de donner les différentes possibilités d'écriture d'un caractère arabe en fonction de sa position dans le texte. Par la suite, et dans une deuxième partie, un tour d'horizon des différentes techniques utilisées dans le domaine de l'OCR est présenté. Il s'agit là des méthodes structurelles, statistiques et géométriques.

Dans le cinquième chapitre « **Contribution à la reconnaissance de l'écriture arabe manuscrite** », c'est l'apport personnel dans ce domaine qui est détaillé. Cette contribution est portée sur deux axes. Le premier concerne la reconnaissance des caractères arabes manuscrits isolés. Une nouvelle approche par utilisation des réseaux de neurones a permis de donner l'image des différents sous forme de banques spécialisées. Ce sont ces banques, construites pendant la phase d'apprentissage, qui

sont utilisées pendant la phase de reconnaissance. Le deuxième axe traite la reconnaissance du texte arabe manuscrit. Toute la chaîne, partant du texte manuscrit jusqu'à obtention des caractères isolés, est clairement détaillée. La aussi c'est une nouvelle approche qu'on propose. Il s'agit en première phase de récupérer les lignes du texte. Ensuite ces lignes sont séparées en mots et enfin ces mots sont séparés en caractères. C'est dans cette séparation de mots en caractères que les choses sont prises d'un autre angle tout nouveau : La méthode de récupération des pseudo-mots même s'il s'agit de pseudo-mots chevauchés.

Dans le dernier chapitre intitulé « **Simulations et résultats** », On présente les résultats de reconnaissance obtenu par application des différentes techniques. On attire l'attention, tout particulièrement, sur l'intérêt des deux nouvelles méthodes : La première concerne la structuration du réseau alors que la deuxième concerne le découpage d'image.

CHAPITRE 1

LES DIFFERENTS ASPECTS DE LA RECONNAISSANCE OPTIQUE DE L'ECRITURE

1-1 Modes de saisie de l'écritur

Avant de parler reconnaissance proprement dite, il question d'abord de saisir l'objet à reconnaître en l'occurrence l'écriture. La première étape d'un système OCR consiste alors à saisir l'objet écriture. Cette saisie est opérer par une digitalisation de cette écriture et à la présenter au système sous une forme lisible avec un minimum de dégradation possible. Cette tâche est relativement difficile à cause de la diversité des formats et de la qualité de l'écriture et du papier. Le capteur utilisé peut ne pas être sensible aux variations des tons et à l'épaisseur des traits, ce qui occasionne des dégradations très perceptibles du dessin des caractères. Deux modes de saisie sont utilisés : le mode statique pour les caractères déjà écrits et le mode dynamique pour les caractères manuscrits à reconnaître en cours d'écriture.

1-1-1 Le mode statique

Il utilise essentiellement des scanners. Le scanner balaye le texte ligne par ligne en une série plus ou moins importante de points. La résolution d'un scanner, exprimée en nombre de points par pouce (ppp ou dpi en anglais), désigne sa capacité à digitaliser les traits fins. En OCR, les valeurs les plus courantes vont de 200 à 600 dpi; une résolution plus grande n'apporte rien à la précision de la digitalisation, bien au contraire elle augmente le nombre de points à traiter et génère même du bruit (grains du papier). Plusieurs types de scanners existent sur le marché, fournissant le choix des images binaires, des images de niveaux de gris et des images en couleur.

1-1-2 Le mode continu

Il utilise une caméra vidéo. Ce mode est très utilisable dans des applications en temps réel (tri automatique du courrier ou reconnaissance de plaques minéralogiques).

1-2 Les différents aspects de la reconnaissance optique de l'écriture

1-2 -1 Introduction

Ces dernières années, des progrès considérables ont été réalisés dans la mise en œuvre des systèmes pour la reconnaissance de l'écriture manuscrite, et cela grâce, d'une part aux nombreux travaux effectués dans ce domaine, et, d'autre part, à la diffusion massive à bas prix des micro-ordinateurs et des systèmes d'acquisition (scanner, tablette à digitaliser...etc.). En outre, la disponibilité de bases de données internationales standards relatives à l'écriture manuscrite permettait à certains chercheurs de ce domaine de rapporter de façon crédible les performances de leurs approches, avec la possibilité de les comparer objectivement. La submersion du marché d'électronique par des PDAs (Personnel Digital Assistants) et des téléphones mobiles intelligents (Smart Phones) avec des capacités de reconnaissance de l'écriture manuscrite latine, est, sans doute, le résultat le plus palpable de ces progrès.

Les travaux de recherches sur l'écriture arabe, bien qu'ils soient moins nombreux en comparaison avec d'autres types d'écriture (ex. le Latin et le Japonais), connaissent ces dernières années un regain d'intérêt [AlBadr 9] [Ami98] [Ben 03]. Alors que certains travaux de recherche ont été consacrés aux caractères isolés [Amin 80] [Mah 94] [Mah 96], d'autres se sont orientés vers les textes, en s'investissant dans la segmentation des mots en caractères et aux aspects liés à leur variation de formes en fonction de leur position dans le mot [Mah 96] [Mah 87]. D'autres travaux se sont consacrés à la reconnaissance des mots manuscrits isolés [Alm 87] [Ame 94] [Beno 98]. Avant de voir les différents aspects de la reconnaissance de l'écriture notons dore et déjà que cette reconnaissance est partagée en deux classes selon le mode d'acquisition. Il s'agit de : La reconnaissance en-ligne et la reconnaissance hors-ligne. Ce sont deux modes

différents d'OCR, ayant chacun ses outils propres d'acquisition et ses algorithmes correspondants de reconnaissance.

1-2 -2 La reconnaissance en-ligne (on-line) ou dynamique

Dans le cadre de la reconnaissance en-ligne, l'écriture est traduite sous forme d'un ensemble de coordonnées ordonnées dans le temps [LEC 93] [ALB 95]. Ce sont les échantillons. Il est ainsi possible de suivre le tracé, de connaître les posés et levés de stylo et éventuellement l'inclinaison et la vitesse. Il faut évidemment un matériel spécifique pour saisir un tel échantillon, c'est le cas notamment des stylos numériques ou des stylets sur agendas électroniques ou sur les Tablets PC.

La reconnaissance en-ligne est généralement beaucoup plus efficace que la reconnaissance hors-ligne car les échantillons sont beaucoup plus informatifs. En revanche, elle nécessite un matériel beaucoup plus coûteux et impose de fortes contraintes au scripteur puisque la capture de l'encre doit se faire au moment de la saisie (capture synchrone) et non a posteriori (capture asynchrone).

Les techniques usitées peuvent avoir un champ applicatif plus vaste permettant la reconnaissance de toute forme abstraite simple (cf. Reconnaissance de formes, Intelligence artificielle faible). Les systèmes actuels procèdent majoritairement par une comparaison de l'échantillon à reconnaître avec ceux contenus dans une base de données. Cette base de données peut être créée de toutes pièces ou être l'objet d'une phase d'apprentissage.

Les techniques de comparaison reposent généralement sur des méthodes statistiques simples pour gagner en vitesse de traitement. La conséquence est que le nombre de formes reconnaissables doit être limité, sans quoi les résultats risquent d'être souvent erronés. En effet, toute la difficulté de la reconnaissance est d'évaluer la similarité entre une forme étudiée et chaque forme de la base de données (il est presque impossible qu'il y ait une correspondance exacte). Il suffit alors de choisir la forme la plus similaire. La reconnaissance idéale doit avoir la même évaluation de similarité que le cerveau, ce dont on se rapproche avec les réseaux de neurones. Mais les méthodes plus rapides (moins complexes) évalueront une similarité entachée d'erreur. Lorsqu'il y a peu de formes dans la base de données, bien séparées, la forme

la plus similaire restera la même, et donc le résultat final sera juste. En augmentant la taille de la base des données, on "rapproche" nécessairement les formes modèles entre elles, et l'erreur sur la similarité peut plus facilement faire pencher la balance vers une mauvaise forme.

1-2 -3 La reconnaissance hors-ligne (off-line) ou statique

La reconnaissance hors-ligne travaille sur une image numérique. C'est le cas notamment de la Reconnaissance Optique de l'Écriture. Dans ce contexte il est impossible de savoir comment ont été tracés les différents motifs [TSAN 00]. Il est seulement possible d'extraire des formes à partir de l'image, en s'appuyant sur les technologies de reconnaissance de forme.

C'est évidemment le type de reconnaissance privilégié pour les traitements asynchrones tels que la lecture de chèque bancaire ou le tri postal.

1-2-4 Reconnaissance de l'imprimé ou du manuscrit

L'approche n'est pas la même selon qu'il s'agisse de reconnaître un imprimé ou un manuscrit. Dans le cas de l'imprimé, les caractères sont bien alignés et souvent bien séparés verticalement, ce qui simplifie la phase de lecture, bien que certaines fontes présentent parfois des accollements qu'il faut défaire [ANI 92]. De plus, le graphisme des caractères est conforme à un style calligraphique (fonte) qui constitue un modèle pour l'identification. Dans le cas du manuscrit, les caractères sont souvent ligaturés et leur graphisme est inégalement proportionné. Cela nécessite l'emploi de techniques de délimitation très spécifiques et souvent des connaissances contextuelles pour guider la lecture.

1-2-5 Reconnaissance monofonte, multifonte, omnifonte

Dans le cas de l'imprimé, la reconnaissance peut être monofonte, multifonte ou omnifonte.

Un système est dit *monofonte* s'il ne peut reconnaître qu'une seule fonte à la fois c'est à dire qu'il ne connaît de graphisme que d'une fonte unique. C'est le cas le plus simple de reconnaissance de caractères imprimés [ANI 92].

Un système est dit *multifonte* s'il est capable de reconnaître divers types de fontes parmi un ensemble de fontes préalablement apprises [BEN 99].

Et un système *omnifonte* est capable de reconnaître toute fonte, généralement sans apprentissage préalable. Cependant ceci est quasiment impossible car il existe des milliers de fontes dont certaines illisible par l'homme (sauf bien sure pour celui qui l'a conçue) et avec un logiciel de création de fonte n'importe qui peut concevoir des fontes à sa guise [ANI 92] a présenté une autre définition pour ce terme c'est l'expression « *polyfonte* » et a qualifié un système polyfonte de système capable de reconnaître un très grand nombre de fontes.

1-2-6 Reconnaissance de caractères ou analyse de documents

Dans le premier cas, la structure du texte est limitée à quelques lignes ou mots. La lecture consiste en un simple repérage des mots dans les lignes, puis à un découpage de chaque mot en caractères. Dans le second cas, il s'agit de données bien structurées dont la lecture nécessite la connaissance de la typographie et la mise en page du document (structure physique et logique du contenu). Certaines fantaisies de présentation, comme dans certains magazines ou journaux, obligent parfois à reconnaître d'abord quelques caractères pour s'assurer du sens de la lecture. La tâche de lecture n'est plus ici un simple prétraitement mais une démarche experte d'analyse d'image: localisation de régions, séparation des régions du texte des régions graphiques et photographiques, étiquetage sémantique des zones textuelles à partir de modèles, détermination de l'ordre de lecture et de la structure du document [TRE97].

1-3 Techniques de prétraitement

Afin d'atténuer les variations dues au scripteur et augmenter les chances d'une bonne reconnaissance, certains prétraitements sont nécessaires. Parmi ces prétraitements, on trouve :

1-3-1 Lissage

L'image du caractère peut être entachée de bruit dû aux artefacts de l'acquisition et souvent à la qualité du document, conduisant soit à des absences de points (trous) soit à des empâtements ou des excroissances et donc à une surcharge de points. Les techniques de lissage permettent de résoudre ces problèmes (voir Figure 1-(b)).

1-3-1-1 Le redressement de la ligne de base

L'idée est de rendre horizontaux les mots à l'aide d'une transformation géométrique de type rotation isométrique des points de l'image (voir Figure 1-(e)).

1-3-1-2 Le redressement des écritures penchées

Cette technique facilite la segmentation préalable des mots en caractères. L'idée est de trouver l'angle moyen d'inclinaison puis de faire une transformation géométrique de type cisaillement de l'image (voir Figure 1-(f)).

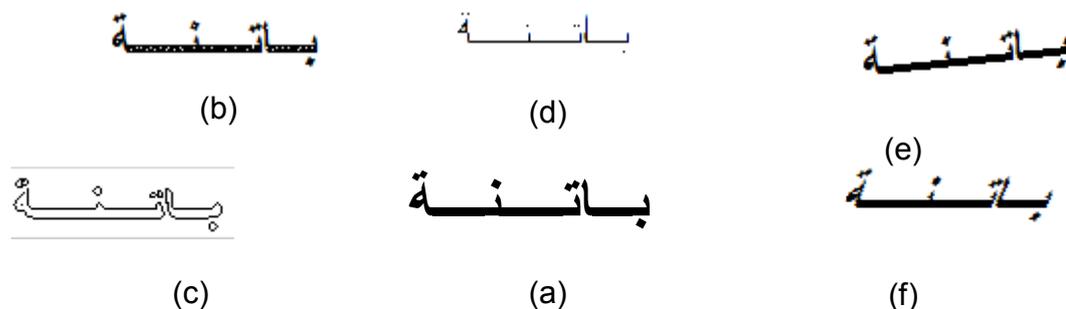


Figure 1-1 Quelques opérations de prétraitement

- (b) → (a) il s'agit d'un filtrage (lissage)
- (a) → (c) il s'agit d'une détection de contour
- (f) → (a) il s'agit d'une correction du slop (inclinaison du mot)
- (e) → (a) il s'agit d'une correction du slant (inclinaison par rapport à la Ligne de base)
- (a) → (d) il s'agit d'une squelettisation

1-3-2 La squelettisation

Sert à obtenir une épaisseur égale à 1 du trait d'écriture et de se ramener ainsi à une écriture linéaire. Le squelette doit préserver la forme, connexité, topologie et extrémités du tracé, et ne doit pas introduire d'éléments parasites (voir Figure 1-d).

1-3-3 La normalisation

Pour se donner toutes les chances d'une réussite pendant la phase de reconnaissance, les images «écriture» doivent être ramenées à une forme standard. Cette standardisation peut concerner la taille de l'image comme elle peut concerner la correction de l'inclinaison des mots ou des lettres. Les détails de ces normalisations seront présentés au chapitre II.

1-4 Types de reconnaissance en écriture

Un objet écriture peut être tout simplement un caractère isolé ou un mot. Tout autre type peut être ramené à l'un de ces deux. Par exemple une ligne texte, est un ensemble de mots lesquels ils sont un ensemble de caractères. Toute l'opération, de reconnaissance en écriture, consiste donc d'abord à récupérer soit le mot soit le caractère à partir de l'image écriture. Cette récupération passe inévitablement par un procédé de découpage de l'objet écriture en mot ou en caractère. Il s'agit de la segmentation.

1-4 Segmentation

Une image texte est un ensemble de lignes. La segmentation commence d'abord par la séparation de ces lignes les uns par rapport aux autres. On peut s'aider pour réaliser cette opération des histogrammes horizontaux [Al-Badr 95].

Puis ces lignes sont séparées en mots. Pour cette opération, on peut se servir des histogrammes verticaux [Al-Badr 95]. La phase de segmentation est terminée à ce stade s'il s'agit d'un traitement des mots entiers. Si c'est plutôt le traitement des caractères

qui est envisagé une étape supplémentaire est encore à envisager : Le découpage des mots en lettres.

1-4-1 Reconnaissance de caractères isolés

C'est la tâche la plus basique d'un système de reconnaissance de l'écriture. L'effort d'analyse est concentré sur un seul élément à la fois du vocabulaire (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. 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).

Ensuite, suivant le type d'indices visuels, plusieurs méthodes sont proposées. Il est souvent habituel de traiter le cadre géométrique par les méthodes syntaxiques ou stochastiques [Bel 97], décrivant la forme à la manière d'une phrase d'un langage. Ces méthodes sont capables de décrire la forme par décomposition en parties et intègrent facilement le contexte linguistique dans la décision. Les indices numériques ou statistiques sont plutôt utilisés par des méthodes globales de corrélation ou de type neuronal.

1-4-2 Reconnaissance de mots

Deux approches s'opposent en reconnaissance des mots : globale et analytique. L'approche globale a une vision générale du mot; elle se base sur une description unique de l'image du mot, vue comme une entité indivisible. Disposant de beaucoup d'informations, elle absorbe plus facilement les variations au niveau de l'écriture. Cependant, cet aspect généraliste la limite à des vocabulaires distincts et réduits. En effet, la discrimination de mots proches est très difficile, et l'apprentissage des modèles nécessite une grande quantité d'échantillons qui est souvent difficile à réunir.

Cette approche est souvent appliquée pour réduire la liste de mots candidats dans le contexte d'une reconnaissance à grands vocabulaires. Il est nécessaire d'utiliser

dans ce cas des primitives très robustes (coarse features), comme dans les travaux de [Gov 94], pour ne pas manquer le mot réel parmi les mots candidats. Le mot reconnu est ensuite trouvé à l'aide de primitives de plus en plus précises (ou d'un classifieur de plus en plus fin). Cette combinaison de classifieurs est appelée combinaison sérielle par [Mad 95], par opposition à la combinaison parallèle où les sorties des classifieurs sont considérées en même temps. Pour les vocabulaires réduits et distincts (exemple: reconnaissance de montants littéraux de chèques bancaires), cette approche reste parfaitement envisageable comme cela a été fait par [Sim 92], [Gil 92], [Kne 97], [Gui 97] et [Sao 97] .

CHAPITRE 2

PRETRAITEMENT POUR LA RECONNAISSANCE

2-1 Introduction

A l'entrée on dispose d'une image numérique brute. On peut soit garder cette image telle qu'elle pour en extraire les primitives de reconnaissance, ou alors à partir de cette image brute extraire une autre forme qui sera utilisée dans le processus de reconnaissance. C'est de cette deuxième alternative qu'il s'agit dans ce chapitre. Il est question donc de voir les différentes formes aux quelles on peut ramener une image pour en extraire les primitives de reconnaissance. Tout d'abord l'objet image est ramené à une forme simple qui a deux niveaux (binaire) : soit le noir soit le blanc. Cette opération est appelée binarisation.

2-2 Binarisation

Pour une évaluation de différentes méthodes de binarisation, on pourra consulter les travaux de [Oiv 95] et [Oiv 96].

On distingue en général deux approches : les approches à seuillage global, et les approches à seuillage adaptatif.

2-2-1 Seuillage global

Le seuillage global consiste à prendre un seuil ajustable, mais identique pour toute l'image. Chaque pixel de l'image est comparé à ce seuil et prend la valeur blanc ou noir selon qu'il est supérieur ou inférieur. Cette classification ne dépend alors que du niveau de gris du pixel considéré. Cette méthode convient pour les documents simples et de bonne qualité. Néanmoins, elle n'est plus applicable lorsque la qualité d'impression du texte n'est pas constante dans toute la page, des caractères peuvent être partiellement perdus. Des problèmes surviendront également si le fond est bruité ou non homogène, dans ce cas des taches parasites peuvent apparaître. Les histogrammes de la répartition des niveaux de gris des pixels de ces images contiennent deux pics nets :

l'un pour la forme, l'autre pour le fond. Un seuil global peut aisément être sélectionné entre les deux pour classer les pixels selon qu'ils appartiennent aux formes ou au fond.

Pour l'évaluation du seuil optimal, on se référera aux travaux de [Ots 79].

2-2-2 Seuillage adaptatif

Dans les documents pour lesquels l'intensité du fond et l'intensité de la forme peuvent varier au sein du document, un seuillage global est inadapté. Il devient nécessaire de choisir le seuil de binarisation de manière locale. On calcule un seuil de binarisation pour chaque pixel de l'image, en fonction de son voisinage.

Par exemple, dans [Ber 86], propose le calcul suivant pour déterminer le seuil de binarisation de chaque pixel :

$$Fmin(x, y) = \min_{(xi,yi) \in v(x,y)} f(xi, yi) \quad 2.1$$

$$Fmax(x, y) = \max_{(xi,yi) \in v(x,y)} f(xi, yi) \quad 2.2$$

$v(x,y)$ est le voisinage du pixel (x, y)

$f(x, y)$ est la valeur de l'intensité du pixel (x, y) dans l'image

$Fmin(x, y)$ est la valeur minimale de l'intensité dans le voisinage du pixel considéré

$Fmax(x, y)$ est la valeur maximale de l'intensité dans le voisinage du pixel considéré

La moyenne des valeurs $Fmin(x, y)$ et $Fmax(x, y)$ est utilisée pour déterminer le seuil de binarisation du pixel (x, y) :

$$g(x, y) = \frac{Fmax(x, y) + Fmin(x, y)}{2} \quad 2.3$$

$$b(x, y) = \begin{cases} 1 & \text{si } f(x, y) < g(x, y) \\ 0 & \text{sinon} \end{cases} \quad 2.4$$

La binarisation des documents bruités reste un domaine de recherche actif. Pour un survey des différentes techniques, on pourra consulter [Sez 04]. Citons aussi les travaux récents de [Gat 06], appliqués à la binarisation de documents dégradés (taches, ombres). Cette méthode inclut une phase de filtrage, une première estimation grossière

du découpage fond/forme, la phase de calcul des seuils locaux proprement dite, et enfin une phase de post-traitements spécifiques. Et citons également les travaux de 2007 de [Che 07], qui combinent deux méthodes anciennes [Way 85] et [Swa 86]. Le seuillage de [Way 85] conserve bien les caractères, mais a tendance à laisser beaucoup de bruits de fond dans l'image. Le seuillage de [Swa 86] filtre une grande partie des bruits de fond, mais a parfois tendance à également supprimer des parties de caractères. En combinant les deux méthodes, le système offre des résultats intéressants, pour une large gamme de situations dans lesquelles les images sont bruitées.

2-3 Elimination de bruit (filtrage)

La binarisation étant faite, on dispose maintenant d'un objet image à deux niveaux. A cette objet sont superposés des pixels «étrangés» qu'il aurait été mieux de s'en débarrassé. Ces pixels sont appelés «bruit» et des techniques spéciales sont utilisées pour nettoyer l'image de ces pixels. Il s'agit des techniques de filtrages qu'on essayera de voir dans cette partie.

2-3-1 Types de filtrages

2-3-1-1 Les filtres linéaires

Un filtre est linéaire s'il peut s'écrire de la manière suivante :

Filtre (c1.image1+c2.image2) = c1Filtre (image1) +c2Filtre (image2)

Filtrer signifie convoler une image $I(x,y)$ avec une fonction $f(x,y)$ appelée réponse

Impulsionnelle (ou opérateur de convolution) du filtre. Dans le cas continu, l'image filtrée est donnée par :

$$If(x,y) = (f * I)(x,y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x',y')I(x-x',y-y')dx'dy' \quad 2.5$$

Dans le cas discret et pour prendre l'exemple simplifié d'une image carrée, les domaines de I et de f sont bornés. Le domaine de I est $[-N/2, +N/2]^2$ si N désigne la taille

de l'image et le domaine de f est $[-K/2, +K/2]^2$ avec nécessairement $K \times N$. La convolution s'écrit alors :

$$If(x, y) = (f * I)(x, y)$$

$$= \sum_{i'=-k/2}^{i'+k/2} \sum_{j'=-k/2}^{j'+k/2} f(i-i', j-j')I(i', j') \quad 2.6$$

On notera que le filtrage linéaire consiste simplement à remplacer chaque niveau de gris par une combinaison linéaire des niveaux de gris des points voisins.

Les filtres linéaires les plus connus sont les filtres passe-haut, passe-bas, Sobel, Prewitt et Laplacien.

2-3-1-2 Les filtres non linéaires

Comme pour la convolution, ces filtres travaillent avec le voisinage du pixel courant. On prend alors le minimal, le maximal ou le médian du pixel et de son voisinage.

Les filtres les plus connus sont les filtres Moyenne, Médiane, Nagao.

2-3-2 Les filtres utilisés

2-3-2-1 Filtre passe-bas

Il est utilisé pour diminuer le bruit sur une image. Le problème avec ce filtre est qu'il atténue également tous les détails de hautes fréquences comme les contours. Ce filtre utilise le masque suivant :

1	1	1
1	1	1
1	1	1

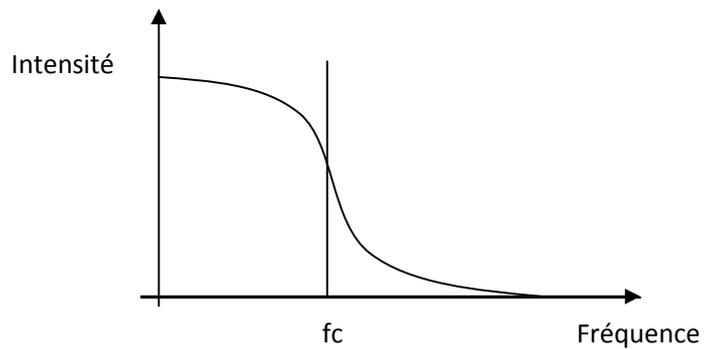


Figure 2.1 Filtre passe-bas

Le coefficient diviseur vaut 9. Ce masque est en fait un moyenneur.

2-3-2-2 Filtre passe-haut

Il est utilisé pour amplifier les détails de hautes fréquences. Il peut permettre par exemple de restaurer des images qui ont été défocalisées et d'accentuer les contours en faisant ressortir les pixels compris entre des zones homogènes.

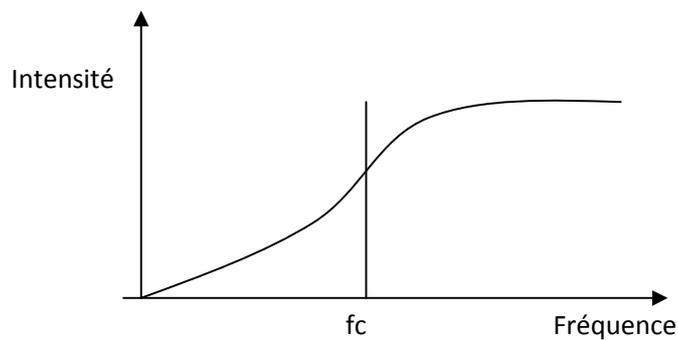


Figure 2.2 Filtre passe-haut

Masque de convolution :

0	-1	0
-1	4	-1
0	-1	0

Le coefficient diviseur vaut 1.

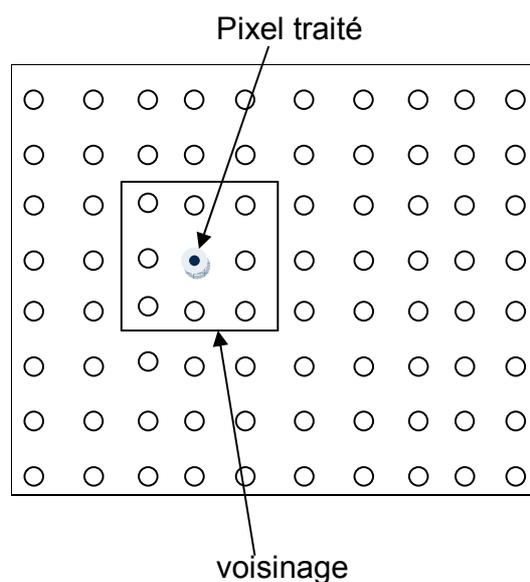
2-3-2-3 Filtre Moyeneur

Le filtrage des bruits blancs conduit à des filtres de moyenne (ou de lissage) qui remplacent le pixel courant par la valeur moyenne des pixels sur une fenêtre 3x3 ou 5x5.

Ce filtre est bien compatible avec notre observation de départ selon laquelle le bruit est caractérisé par d'importantes hautes fréquences. Mais certaines fréquences (assimilées à de fortes variations dans les niveaux de gris) sont définitivement détruites, le lissage agissant comme un filtre passe-bas. L'image est non seulement lissée mais elle est aussi fortement altérée, les contours étant particulièrement dégradés par les filtres de moyenne.

Le filtre de moyenne est une version d'une gamme plus vaste de filtres de lissages dits linéaires. Ici, le masque représentant ce filtre est le suivant :

1	1	1
1	1	1
1	1	1



Ce filtre, normalisé par un facteur 1/9 pour le ramener dans les valeurs normales, génère un lissage fort. Un lissage moyen peut être réalisé par le biais du filtre suivant :

1	2	1
2	4	2
1	2	1

Le facteur de normalisation est sans surprise 1/16 et le coefficient plus élevé du pixel central permet d'atténuer l'effet de lissage.

2-3-2-4 Filtre Médian

Les inconvénients du filtre de Moyenne induisent une orientation vers le filtre dit Médian, qui préserve les contours. Ce filtre vise à remplacer la valeur du pixel central par la valeur médiane de la répartition (on trie les luminances dans l'ordre croissant) des niveaux de gris des pixels situés à l'intérieur de cette fenêtre (figure 3.8)

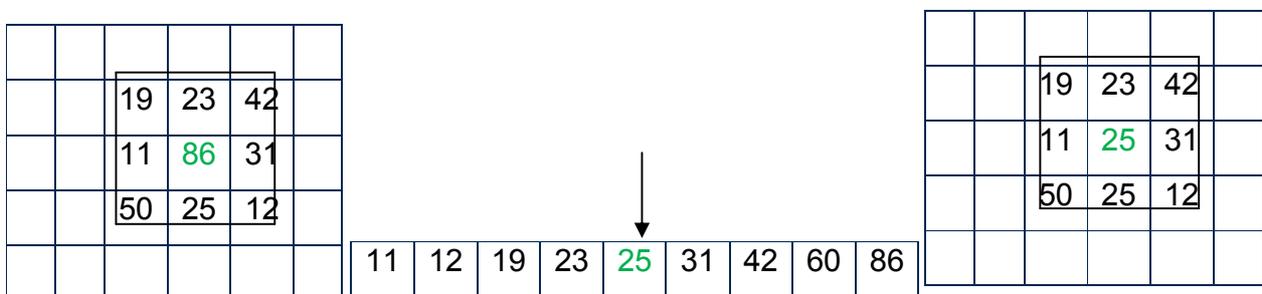


Figure 2-3 Mise en œuvre du filtre Médian

Ce filtre - non linéaire - permet d'éliminer des pixels isolés ou les traînées fines noyées dans des zones assez homogènes.

2-4 Normalisation

Normaliser laisse entendre ramener quelque chose à un standard. Ca peut être corrigé une inclinaison comme ca peut être ramené l'image à une dimension fixe comme on l'a déjà vu au premier chapitre.

2-4-1 Correction de l'inclinaison des lignes

La correction de l'inclinaison des lignes de texte (également appelée correction de "skew"), consiste à redresser horizontalement les lignes d'écriture obliques.

Plusieurs méthodes sont disponibles. Les deux plus populaires sont la transformée de Hough (appliquée sur les centre de gravité des composantes connexes), et les histogrammes de projection.

La méthode des histogrammes consiste à parcourir l'image selon des directions « d » proches de l'horizontale, et à compter le nombre de pixels noirs selon ces direction pour chaque ligne (voir figure 2.4).

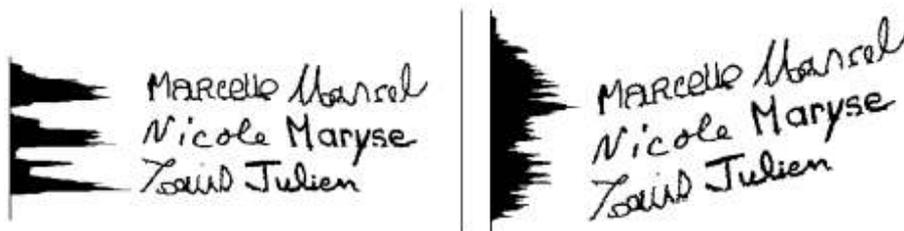


Figure 2.4 Correction de l'inclinaison des lignes à l'aide d'histogrammes de projection horizontale.

Les pics les plus prononcés correspondent à la rotation correcte (figure extraite de [Car 03])

L'angle du document (ou de la ligne), θ , est celui qui correspond à l'histogramme d'entropie maximale. Pour corriger cette inclinaison, il suffit d'appliquer une rotation de l'image d'angle θ :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad 2.7$$

2-4-2 Correction de l'inclinaison des lettres

Certains scripteurs écrivent leurs lettres de façon inclinée par rapport à l'axe vertical. Cette inclinaison de l'écriture est également appelée "slant". Les lettres peuvent être inclinées vers la droite ou vers la gauche. Il convient de corriger cette inclinaison de l'écriture pour la rendre la plus indépendante possible des spécificités d'écriture du scripteur. Plusieurs approches ont été proposées. Ci-dessous est décrit l'un d'eux.

Le contour de l'image de l'écriture est utilisé [Miy 00] [Bri 00]. On parcourt ce contour en comptant le nombre de fois qu'on se déplace dans trois directions privilégiées : n_1 , n_2 et n_3 (voir figure 2.5). $n_1 - n_2$

L'angle d'inclinaison de l'écriture est obtenu en calculant :

$$\theta = \tan^{-1}\left(\frac{n_1 - n_3}{n_1 + n_2 + n_3}\right) \quad 2.8$$

La correction de l'inclinaison se fait par translation des lignes :

$$x' = x - y \cdot \tan^{-1}(\theta) \quad 2.9$$

$$y' = y$$

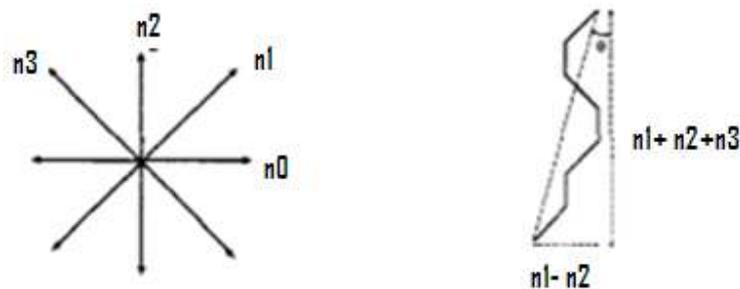


Figure 2.5 – Inclinaison moyenne de l'écriture évaluée sur le contour de l'image (figure extraite de [Miy 00]). A gauche : n_0 , n_1 , n_2 et n_3 sont compteurs associés à 4 directions privilégiées. A droite : l'angle d'inclinaison se déduit en parcourant le contour.

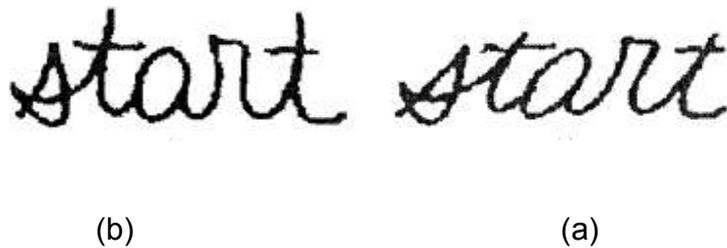


Figure 2.6 – Correction du Slant (figure extraite de [Boz 89])
 (a) image originale.
 (b) image corrigée.

2-4-3 Normalisation de l'Echelle (Ré échantillonnage)

La taille de l'image (écriture) peut varier d'un scripteur à un autre, ce qui peut causer une instabilité des paramètres. Une technique naturelle de prétraitement consiste à ramener ces images à la même taille. Une technique de normalisation toute naturelle est de trouver pour chaque pixel de l'image originale son équivalent dans la nouvelle image : A chaque coordonnée du pixel d'origine, on trouve l'équivalent dans la nouvelle image juste par une règle de trois.

Cette opération donne lieu à une nouvelle image avec des vides entre pixels, surtout s'il s'agit d'une nouvelle dimension plus grande que l'ancienne. Un remplissage, par interpolation, de ces vides s'impose alors.

Soit $f(i,j)$ avec $1 \leq i \leq N$, $1 \leq j \leq M$ l'image originale et $g(k,l)$ l'image résultante d'un changement d'échelle. Soit « F » le facteur d'échelle à appliquer. Nous avons :

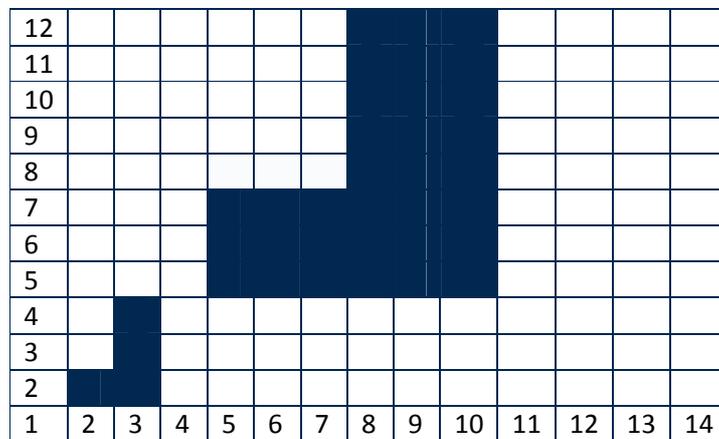
$$k = F.i \quad 1 \leq k \leq F.N \quad 2.10$$

$$l = F.j \quad 1 \leq l \leq F.M$$

Deux situations se présentent alors :

1. $F \geq 1$, il s'agit d'un agrandissement. Si on se contente uniquement de l'application des deux expressions ci-dessus, la nouvelle image contient le même nombre de pixels que l'image originale. La seule différence est que les pixels de la nouvelle image n'occupent pas les mêmes positions que leur homologue de l'image d'origine. Un deuxième passage est opéré. Il tient compte cette fois-ci du facteur d'échelle de la manière suivante : Chaque pixel

de la nouvelle image est entouré par un nombre de pixels correspondant au facteur d'échelle, lui y compris. Si par exemple le facteur d'échelle est égal à trois, chaque pixel de la nouvelle image se voit entouré de huit pixels comme le montre la figure suivante.



Figures 2.7 Agrandissement F=3

2. $F < 1$, il s'agit d'une réduction. Ici un deuxième passage n'est pas obligatoire. En effet à partir des expressions de l'équation 2.10, plusieurs pixels de l'image originale se trouvent superposés dans la nouvelle image : il n'y a pas de trous.

Pour la mise en œuvre de cette technique deux possibilités sont offertes à l'utilisateur : Il peut choisir les dimensions de la nouvelle image en pixels, ou à partir d'un facteur d'échelle par rapport à l'image d'origine. Dans les deux cas, le programme se charge de traduire l'information en facteur d'échelle, puis il procède à l'application des expressions de l'équation 2.10.

Un problème se pose cependant : il s'agit de la déformation des certains caractères fins comme le " ا " 'alif', par rapport à des caractères épais comme 'ض'. Plusieurs algorithmes ont été proposés pour contourner ce problème [Hus 72], [Sri 89] .

2-5 Les techniques de changement de représentation

Avant de parler reconnaissance, il faut parler plutôt extraction de primitive. En effet la première étape de toute opération de reconnaissance est l'étape d'extraction des caractéristiques de l'image (écriture) pour pouvoir l'utiliser par la suite dans l'étape de reconnaissance. Il est tout naturel alors de débiter ce chapitre par les différentes étapes d'extraction de ces caractéristiques

A partir de l'image numérique brute, on relève quatre types d'extraction de primitive. Il s'agit de :

- 1- L'extraction des composantes connexes ;
- 2- L'extraction du contour ;
- 3- La squelettisation ;
- 4- Les transformations mathématiques qui opèrent globalement sur l'image.

Ces techniques de changement de représentation de l'information seront détaillées ci dessous. C'est à partir d'une de ces représentations qu'est ensuite réalisée l'étape d'extraction des primitives proprement dite.

2-5-1 Extraction des composantes connexes

L'extraction des composantes connexes, procédure également appelée capture des connexités ou étiquetage des pixels, est largement utilisée en Reconnaissance des Formes (RdF) pour segmenter les images binaires. La technique consiste à regrouper les pixels voisins dans un ensemble appelé composante connexe. Chaque ensemble est disjoint des autres et peut ensuite être aisément isolé. La 4-connexité est distinguée de la 8-connexité suivant que le critère de voisinage comprend les 4 ou les 8 voisins d'un pixel.

Il existe deux principaux algorithmes pour accomplir cette tâche :

➤ le premier est basé sur une procédure de suivi de contour [LER 91] : en parcourant le contour d'un objet et en revenant au point de départ, une composante connexe est délimitée, à l'exclusion cependant des contours intérieurs correspondant aux éventuels trous.

➤ le second algorithme procède par une propagation d'un étiquetage des pixels lorsque l'on effectue un balayage des lignes et des colonnes de l'image.

Dans [DAR 94], DARGENTON Patrice élabore un algorithme fonctionnant en une seule passe, suivant le critère de 4-connexité. Dans cette méthode, la propagation de l'étiquette des pixels suivant les colonnes (verticalement) est prioritaire sur la propagation suivant les lignes (horizontalement). Il procède de la manière suivante :

En parcourant une ligne horizontale de gauche à droite, on associe un numéro (une étiquette) à chaque pixel de telle sorte que tous les pixels voisins portent le même numéro (le numéro zéro est réservé pour un pixel "vide"). Lorsque sur cette ligne, le voisinage est interrompu, puis reprend plus loin, le numéro est incrémenté de 1. Les étiquettes sont représentées par les lettres A, B et C sur la figure 2.8a.

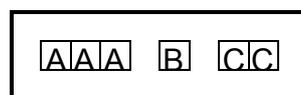


Figure 2.8a : Propagation de l'étiquetage des pixels de gauche à droite sur une ligne horizontale

Lorsqu'une nouvelle ligne est commencée, on propage naturellement l'étiquetage de haut en bas en recopiant le numéro du pixel qui se trouve au-dessus du premier pixel de la nouvelle ligne. S'il n'y a pas de pixel au-dessus, un nouveau numéro est utilisé (figure 2.8 b).

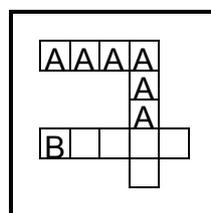


Figure 2.8 b : Propagation de l'étiquetage des pixels de haut en bas sur une colonne verticale

Lorsqu'un conflit se présente entre la propagation horizontale et la propagation verticale des étiquettes, deux cas se présentent alors (figure 2.8c) :

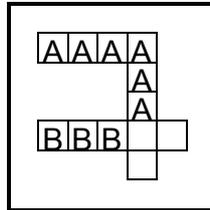


Figure 2.8 c : Conflit entre la propagation horizontale et verticale

1er cas : si le numéro des pixels horizontaux correspond à une nouvelle étiquette, alors il est facile de résoudre le conflit en remplaçant la nouvelle étiquette de tous les pixels à gauche du point de conflit par le numéro prioritaire du pixel de la ligne précédente (figure 2.8d). La nouvelle étiquette est alors annulée ;

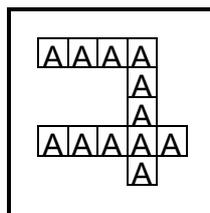


Figure 2.8d : Résolution du conflit correspondant au 1er cas

2e cas : Si le numéro des pixels horizontaux a déjà été propagé à partir de la ligne précédente, il serait trop long de remplacer tous les pixels correspondants. Aussi, on note dans un tableau que les deux étiquettes en conflit désignent une unique composante connexe (figure 2.8 e).

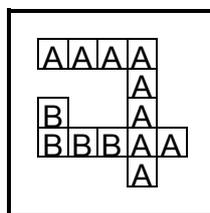


Figure 2.8e : Résolution du conflit correspondant au 2e cas

Les paramètres que l'on peut calculer sur les composantes connexes sont principalement l'aire, la longueur du contour et la délimitation du rectangle englobant la composante connexe.

Le plus souvent, l'extraction des composantes connexes est utilisée comme traitement préalable aux autres types de représentation des données, afin de procéder à une segmentation initiale de l'image. Ainsi l'étape suivante peut être le retour à l'image brute (mais segmentée) sur laquelle on pourra alors extraire les primitives statistiques;

2-5-2 Extraction du contour

L'exploitation du contour est fréquente, tant en reconnaissance de l'écriture qu'en RdF en général, étant donné la facilité de l'extraction.

Le contour est également utilisé comme étape préalable à un changement de représentation de l'information, en tant qu'empreinte des formes contenant une quantité réduite de données.

Dans les images à niveaux de gris, il est intéressant d'extraire le contour à l'aide d'un calcul de gradient. Ce contour est alors d'autant plus marqué que le niveau des pixels résultant du gradient est élevé.

Nous préciserons maintenant quelles sont les différentes représentations possibles auxquelles on peut aboutir à partir du contour.

L'analyse du contour est souvent utilisée en reconnaissance des textes manuscrits pour la segmentation des mots en lettres. Elle sera détaillée dans le paragraphe 5.2.5.

2-5-2-1 La vectorisation

Les pixels du contour sont ramenés à un vecteur d'identité. Pour garantir une invariance par rapport à la translation, c'est-à-dire l'invariance à la position de la forme, la quantité d'information contenue cette liste de pixels est avantageusement réduite à l'aide du code de FREEMAN en une liste de vecteurs unitaires. Le principe est de mémoriser seulement la position de chaque pixel par rapport au pixel précédent (4 ou 8 directions en fonction de la continuité du contour) au lieu des deux coordonnées de tous les pixels dans le plan.

2-5-2-2 La transformation globale du contour

A partir de cette liste de vecteurs, l'objectif de la transformation globale du contour est d'intégrer à la reconnaissance, des propriétés d'invariance à certaines déformations ou transformations que peut subir la forme, telles que :

- La rotation,
- Le changement de taille ou d'échelle,
- Le cisaillement, l'homothétie ou une autre transformation affine,
- Une transformation non linéaire (torsion, distorsion, ...).

Lorsque l'on compare une forme test avec une forme apprise, il est nécessaire de tenir compte d'autres variables telles que, par exemple, le point de départ, du contour, dans la chaîne ou la résolution du contour (nombre de vecteurs pour le décrire, quelle que soit sa taille).

Les principales transformations globales du contour sont les suivantes :

- descripteurs de Fourier
- calcul des moments
- transcodage du contour : par exemple, par rapport au centre de gravité de la forme, la mémorisation du plus grand rayon en fonction de chaque direction échantillonnée.

Ces transformations sont adaptées à des problèmes généraux de reconnaissance des formes, tels ceux rencontrés dans le domaine de l'analyse de scène en robotique, où les formes quelconques peuvent être vues dans toutes les directions, en perspective, avec des parties cachées et souvent du bruit, et aussi en trois dimensions. Dans le domaine de la reconnaissance de l'écriture, les caractères ne sont pas des formes quelconques, elles sont toujours vues à plat, et on ne cherche jamais à les reconnaître à l'envers, dans un miroir, déformées ou partiellement occultées. Il reste cependant des déformations simples qui sont effectivement rencontrées. Par exemple en OCR, le style italique est une déformation modélisable par un cisaillement, qui est d'ailleurs créé de la sorte dans les logiciels de traitement de texte à partir des modèles vectoriels des

caractères droits. Pour l'écriture manuscrite, les déformations sont plus importantes que dans le simple style italique et plus variées.

2-5-2-3 L'approximation du contour : représentation en lignes raccordées (Fitting line segment)

L'objectif de l'approximation du contour est également la reconnaissance. Elle est basée sur l'appariement (la ressemblance) : Les lignes droites fournissent une approximation simple de la courbure des frontières (curve boundary). Un algorithme séquentiel intéressant qui utilise les lignes droites pour le suivi de courbe est le suivant :

Algorithme

1. Approximer la courbe par le segment de droite joignant ces limite (A,B);
2. Si la distance entre ce segment de droite et le point le plus éloigné de la courbe (C) est supérieure à un seuil prédéterminé, joindre AC et BC ;
3. Répéter la procédure pour les nouveaux segments AC et BC et continuer jusqu'à atteindre la précision voulue.

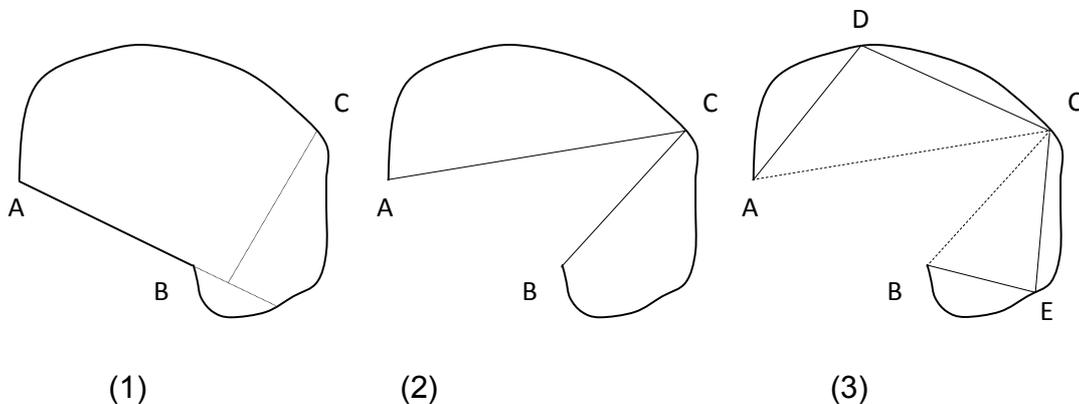


Figure 2.9 : Approximation successive par des lignes droites

Puisque l'opération de reconnaissance qui va suivre est basée sur l'appariement, entre le vecteur contenu dans la base d'apprentissage et le vecteur obtenu par découpage, il est nécessaire alors d'avoir le même nombre de composante pour les deux vecteurs. Le seuil de découpage est choisi alors de sorte à ramener le nombre de segments du vecteur en cours de découpage égale au nombre de segments de la base

de reconnaissance. L'algorithme procède au changement de ce seuil, vers le haut ou vers le bas, jusqu'à trouver un nombre de segment identique entre le vecteur en cours de découpage et ceux de la base.

2-6 Squelettisation

La squelettisation est une opération qui permet de passer d'une image à sa représentation en "fil de fer". Le squelette a un pixel d'épaisseur. C'est une manière de représenter l'information indépendamment de l'épaisseur initiale de l'écriture.

Il permet d'extraire des caractéristiques importantes, comme les intersections et le nombre de tracés, leurs positions relatives. Il est également possible de renormaliser l'épaisseur de l'écriture à partir du squelette.

Il n'existe pas de définition unique du squelette. Le squelette doit seulement remplir certaines conditions :

- ☛ Il doit être aussi fin que possible (typiquement, 1 pixel d'épaisseur)
- ☛ Il doit respecter la connexité
- ☛ Il doit être centré dans la forme qu'il représente ;
- ☛ Il doit conserver les propriétés topologiques de la forme comme le nombre de parties et le nombre de trous ;
- ☛ Il doit respecter les propriétés métriques de la forme.

A chaque composante connexe de la forme, le squelette correspondant ne doit être composé que d'une seule composante connexe d'un pixel d'épaisseur incluse dans la première (voir figure 2.10).

Il existe de nombreuses méthodes de squelettisation. L'une des manières d'évaluer le squelette est de calculer l'axe médian de la forme, comme l'ensemble des centres des boules maximales de cette forme (voir figure 2.11).

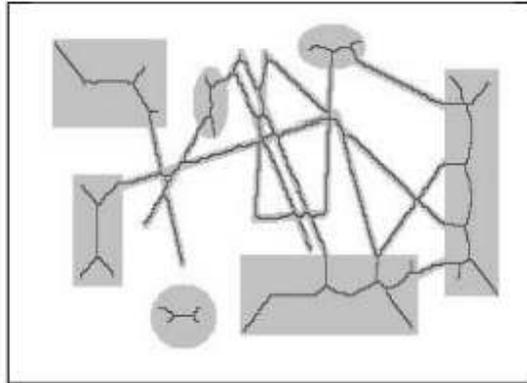


Figure 2.10 – Squelettisation d'une forme quelconque (figure extraite de [Dup 03])

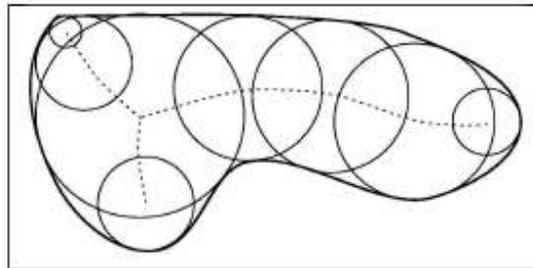


Figure 2.11– Axe médian d'une forme continue (trait pointillé) dont seul le contour est représenté (figure extraite de [Thi 94]).

Un autre moyen d'obtenir le squelette est de procéder par affinages successifs : la forme est "épluchée" (peeling) de manière itérative, en maintenant valide le critère de connexité.

L'algorithme de [Hil 69] exploite ce principe.

On définit le voisinage du pixel P_1 comme :

P_9	P_2	P_3
P_8	P_1	P_4
P_7	P_6	P_5

On définit également :

$A(P_1)$: nombre de transitions 0 vers 1 dans la séquence $P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9$

$B(P_1)$: nombre de pixels noirs dans le voisinage de P_1 (P_1 étant exclu).

$$B(P_1) = \sum_{i=2}^{i=9} P_i \text{ avec } \begin{cases} P_i = 1 \text{ si le pixel est noir} \\ P_i = 0 \text{ si le pixel est blanc} \end{cases} \quad 2.11$$

L'image est parcourue de multiples fois, et un pixel noir est marqué comme effaçable s'il respecte les quatre conditions suivantes :

$$2 \leq B(P_1) \leq 6$$

$$A(P_1) = 1$$

$$P_2 \cdot P_4 \cdot P_8 = 0 \text{ ou } A(P_2) \neq 1$$

$$P_2 \cdot P_4 \cdot P_8 = 0 \text{ ou } A(P_4) \neq 1$$

- Condition1 : $2 \leq B(P_1) \leq 6$. La première inégalité s'assure que P_1 n'est ni un point isolé, ni une extrémité. La deuxième inégalité s'assure que P_1 est bien un point du contour de la forme.
- Condition2 : $A(P_1) = 1$. Cette règle maintient le critère de connectivité.
- Condition3 : $P_2 \cdot P_4 \cdot P_8 = 0$ ou $A(P_2) \neq 1$. Cette règle s'assure que les lignes verticales de 2 pixels de large ne sont pas totalement érodées.
- Condition4 : $P_2 \cdot P_4 \cdot P_6 = 0$ ou $A(P_4) \neq 1$. Cette règle s'assure que les lignes horizontales de 2 pixels de large ne sont pas totalement érodées.

Tous les points marqués comme effaçables sont ensuite effacés, et on réitère l'opération sur la nouvelle image, jusqu'à ce que plus aucun point ne soit effaçable.

De manière générale, les méthodes d'obtention du squelette par érosions successives fonctionnent selon le principe suivant :

1. répéter
2. Appliquer masques et règles pour déterminer les pixels noirs effaçables

3. Changer tous les pixels marqués comme effaçables de noir vers blanc
4. jusqu'à ne plus trouver de pixel n'a été effacé

En générale, il existe trois grandes classe d'algorithmes de squelettisation : les algorithmes à critères topologiques, ceux à critères métriques et ceux dérivés de la vectorisation. Nous nous contentons ici à donner un exemple d'algorithme général à critère topologique.

- ☛ Ces algorithmes appelés aussi algorithmes de « pelage » ou « thinning algorithmes », travaillent d'une manière itérative en supprimant à chaque étape, le long de la frontière de la forme, les points appelés **inessentiels** (c'est-à-dire n'appartenant pas au squelette). Ils utilisent souvent des fonctions booléennes opérant sur des voisinages de points, déterminant à chaque passage la validité des points frontières.

L'algorithme général

```

début
  répéter
    inessentiel ← faux
    pour tout poit dans image faire
      si POINTNESSENTIEL(point) alors
        inessentiel ← vrai
        image(point) ← 0
      finsi
    finpour
  jusqu'à inessentiel = faux
fin

```

- ☛ Pour déterminer si un point est inessentiel, [Ser 82] utilise les masques suivants :

0	0	0
X	1	X
1	1	1

X	0	0
1	1	0
X	1	X

L'ensemble des voisinages 3x3 d'un point inessentiel est obtenu à partir de ces configurations par rotation de 90°.

- L'algorithme général est type séquentiel. Il consiste à balayer l'image ligne par ligne et à supprimer les points inessentiels au fur et à mesure qu'ils sont rencontrés. D'autres algorithmes de type parallèle, consistent à décomposer la suppression en sous cycles permettant d'approcher le squelette par raffinement successif.

Pour plus d'information sur les différents algorithmes de squelettisations nous pouvons consulter [Tho 78] [Mar79].

CHAPITRE 3

EXTRACTION DES PRIMITIVES

3-1 Introduction

L'image brute, sous forme de matrice de pixels, est utilisée dans certaines opérations pour déterminer certaines caractéristiques spatiales de l'objet en cours du traitement (l'écriture). Il peut s'agir du filtrage spatial, de la détermination du périmètre (contour), de la détermination de la surface, de la recherche de l'orientation ...etc. Ces opérations sont très utiles dans les étapes de post traitement. Ils préparent en fait l'image au traitement proprement dit. La recherche de l'orientation permet, par exemple de procéder à la rectification de l'inclinaison des objets avant tous traitements.

En plus de ces traitements spatiaux, tout un ensemble de traitements dans d'autres espaces peut avoir lieu. L'image (spatiale) est ramenée dans un autre espace dans lequel d'autres caractéristiques sont plus apparentes. Pour opérer cette opération de changement d'espace, certaines transformations manipulent le corps plein de l'image alors que d'autres manipulent la frontière de celle-ci représentée par son contour. Pour un soucis d'efficacité, toutes ces transformations cherchent en premier lieu une insensibilité (invariabilité) par rapport à l'échelle, à la rotation et enfin à la translation. Ce chapitre traite le détail de ces transformations. Il répond aux couple de questions suivant : Pourquoi et comment est opéré telle ou telle transformation.

3-2 Moments invariants de Hu

Cette méthode est issue des études de Hu [Hu 62] sur les moments. Le concept des moments est très utilisé en mécanique classique et en statistique. Hu a élaboré un théorème énonçant qu'un ensemble infini de moments calculés à partir d'une image permet de la décrire uniquement et complètement. Il a aussi créé une version de ces moments invariants en translation, en rotation et en homothétie.

Les moments sont particulièrement sensibles aux variations de la distribution de masse dans la silhouette de l'image [Ree 88], mais ce n'est pas un problème pour des images binaires [Cast 95]. Les moments ne font pas que tenir compte des contours, mais aussi des « trous » qui se trouvent à l'intérieur de l'objet. Ils ont donc l'avantage de bien représenter les formes complexes. Les moments de Hu sont utilisés pour la reconnaissance d'avions [Ree 88] et de poissons [Cast 95] etc....

Les moments d'ordre $p+q$ en deux dimensions d'une fonction $f(x,y)$ se définissent par la relation 3.1 :

$$m_{pq} = \iint_{-\infty-\infty}^{+\infty+\infty} x^p y^q f(x,y) dx dy \quad (3.1)$$

où p et q sont l'ordre du moment par rapport à x et y respectivement. Dans le cas d'une image binaire, cette relation devient.

$$f(x,y) = \begin{cases} 1 & \text{pour } (x,y) \in \text{objet} \\ 0 & \text{sinon} \end{cases} \quad (3.2)$$

$$m_{pq} = \iint_{-\infty-\infty}^{+\infty+\infty} x^p y^q dx dy \quad (3.3)$$

Le moment de l'image binaire d'un objet peut être alors simplifié pour s'exprimer selon la relation (3.3).

$$m_{pq} = \sum_x \sum_y x^p y^q \quad (3.4)$$

La mesure doit être invariante en translation. À cette fin, l'image est centrée par rapport à son centre de gravité. Ceci permet d'obtenir une relation invariante en translation (équation 3.4) :

$$\mu_{pq} = \sum_x \sum_y (x - x_c)^p (y - y_c)^q \quad (3.5)$$

où x_c et y_c sont les centres de gravité de l'image de l'objet et μ_{pq} est le moment central. À partir de ces moments centraux, de nouveaux moments qui sont invariants par homothétie, c'est-à-dire qu'ils sont invariants aux changements d'échelle, peuvent être évalués par la relation (3.5).

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\frac{p+q}{2}+1}} \quad (3.6)$$

À partir de cette dernière équation, Hu a dérivé des moments qui sont aussi invariants en rotation. Les développements mathématiques menant à ces moments invariants en rotation étant fastidieux, les lecteurs sont référés à [Hu 62] pour plus de détails. Seuls les sept premiers sont détaillés ici aux équations (3.7) à (3.13), car ceux d'ordre plus élevés ont de très faibles valeurs en pratique et s'avèrent instables.

$$\varphi_1 = \eta_{20} - \eta_{02} \quad (3.7)$$

$$\varphi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (3.8)$$

$$\varphi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (3.9)$$

$$\varphi_4 = (\eta_{30} + 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (3.10)$$

$$\varphi_5 = (\eta_{30} - 3\eta_{12}) \cdot (\eta_{30} - \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (3.11)$$

$$\varphi_6 = (\eta_{20} - \eta_{02}) [(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} - \eta_{03}) \quad (3.12)$$

$$\varphi_7 = (3\eta_{21} - \eta_{03}) \cdot (\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (3.13)$$

Ce sont ces moments qui remplacent l'image dans son nouvel espace : Ils représentent en quelque sorte l'image de l'image et ils portent en eux toutes les informations que contient l'image mère. Dans le cas de notre étude, ces moments sont utilisées pour la reconnaissance de l'écriture (voir chapitre 6).

3-2-1 Application de la méthode des moments pour la reconnaissance

La première étape pour la reconnaissance des caractères manuscrits, consiste à extraire les moments géométriques, qui permettent de trouver les caractéristiques propres aux formes des caractères à reconnaître, l'exploitation de ces moments pourrait permettre de distinguer les différents caractères entre eux.

En premier temps une base de donnée est écrite par un scripteur, contenant les échantillons (Caractères), et pour échantillons chaque les sept moments sont obtenues comme suit :

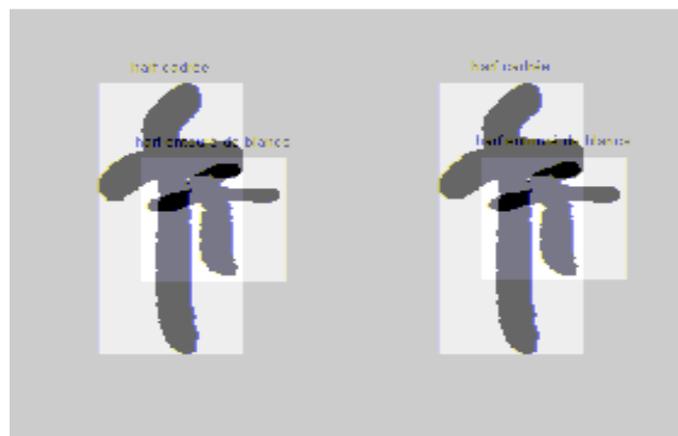
- Lecture du modèle de caractère à partir de la base de données.
- calcul des moments m_{pq} .
- calcul des moments centraux μ_{pq}
- calcul des moments invariants φ_{ik}

k : Représente le nombre des échantillons

Tous les résultats stockés, servent à construire une bibliothèque de référence.

Exemples d'application

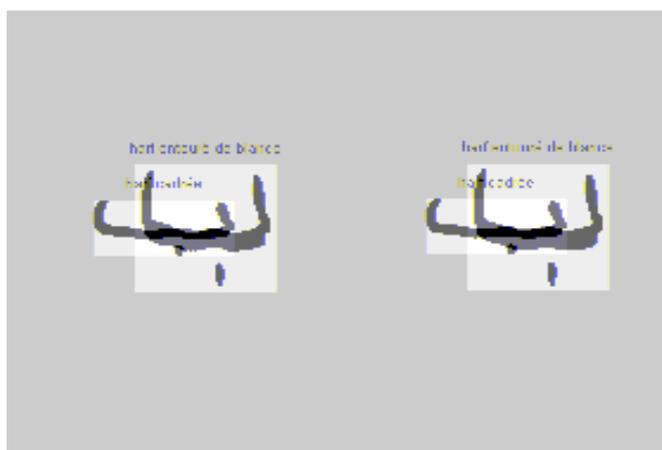
Voici un exemple des sept moments invariants extrait de trois caractères arabes.



ALIF cadrée (64x64)

φ_1	0.30398
-------------	---------

φ_2	0.018872
φ_3	0.013612
φ_4	1.4275
φ_5	0.12213
φ_6	0.19611
φ_7	-0.038017



BA cadrée (64x64)

φ_1	0.32807
φ_2	0.026063
φ_3	0.0004732
φ_4	1.844
φ_5	0.036422

φ_6	0.2977
φ_7	-0.0085768



AIN cadrée (64x64)

φ_1	0.33095
φ_2	0.027197
φ_3	0.010404
φ_4	1.9043
φ_5	0.083704
φ_6	0.31404
φ_7	0.26792

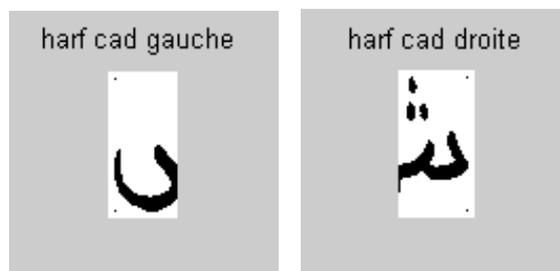
3-2-2 Découpage de l'image du caractère

Comme on l'a signalé auparavant une nouvelle technique de reconnaissance est mise en œuvre. Cette technique opère un découpage de l'image (caractère dans notre application) dans plusieurs directions. D'abord le centre de gravité de l'image est retenu comme centre de découpage à partir duquel la dite image est découpée :

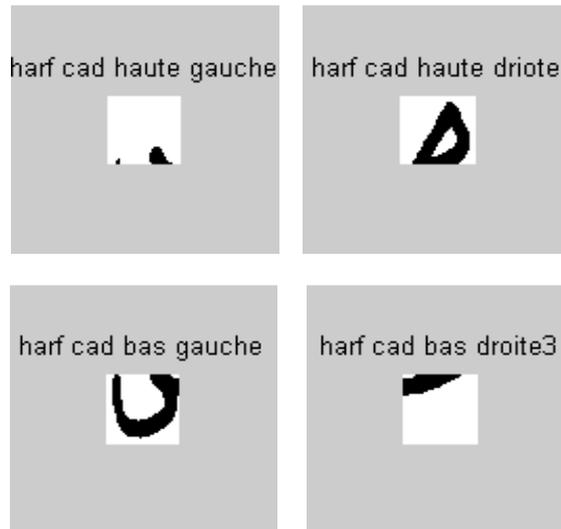
- Soit horizontalement ce qui donne naissance à deux sous images : une sous image du haut et une sous image du bas. Chacune de ces sous image va faire l'objet d'une identification à part ;



- Soit verticalement ce qui donne naissance à deux sous images : une sous image de gauche et une sous image de droite. Chacune de ces sous image va faire l'objet d'une identification à part ;



- Soit dans les deux sens ce qui donne naissance à quatre sous images : une sous image haut gauche, une sous image haut droite, une sous image bas gauche et une sous image bas droite. Chacune de ces sous image va faire également l'objet d'une identification à part.



Le passage d'un découpage à l'autre est opéré selon les besoins de l'opération de reconnaissance. En effet si cette opération de reconnaissance de l'image complète a échouée, avec un certain pourcentage, on passe aux différents découpages dans un ordre bien définit.

Ci dessous on donne des exemples de découpages du caractère « Ain » dans les trois directions citées ci-dessus.

3-2-2-1 Découpage horizontal

- **Partie haute**



AIN HH cadrée

φ_1	0.33095
φ_2	0.027197
φ_3	0.010404
φ_4	1.9043
φ_5	0.083704
φ_6	0.31404
φ_7	0.26792

- **Partie basse**



AIN HB cadrée

φ_1	1.2873
φ_2	1.3461
φ_3	51.164
φ_4	76.268
φ_5	4527.8
φ_6	88.398

φ_7	170.68
-------------	--------

3-2-2-2 Découpage vertical

- **Partie gauche**



AIN VG cadrée

φ_1	0.14058
φ_2	0.019752
φ_3	0.011817
φ_4	0.011676
φ_5	0.00013528
φ_6	0.001636
φ_7	-6.7066e-008

- **Partie droite**



AIN VD cadrée

φ_1	0.084209
φ_2	0.0069919
φ_3	0.0039138
φ_4	0.0027682
φ_5	5.6114e-006
φ_6	0.00020778
φ_7	-7.3273e-007

3-2-2-3 Découpage horizontal et vertical

- **Partie bas gauche**



AIN GB cadrée

φ_1	0.73143
φ_2	0.53306
φ_3	15.101
φ_4	15.14
φ_5	220.66
φ_6	11.053
φ_7	0.035996

- **Partie haut gauche**



AIN GH cadrée

φ_1	0.90669
φ_2	0.81655
φ_3	28.669
φ_4	28.782
φ_5	797.44

φ_6	26.006
φ_7	0.26299

- **Partie bas droite**



AIN DB cadrée

φ_1	0.41977
φ_2	0.17603
φ_3	3.9267
φ_4	3.9298
φ_5	15.323
φ_6	1.6487
φ_7	0.00030065

- **Partie haut droite**



AIN DH cadrée

φ_1	0.32588	3-3 Les mo men ts de Zer nike
φ_2	0.10605	
φ_3	2.1726	
φ_4	2.1744	
φ_5	4.6909	
φ_6	0.70808	
φ_7	0.00014624	

3-3-1 Introduction

Comme pour les moments géométriques, les moments de Zernike sont également utilisés pour la reconnaissance et l'identification. La naissance de ces moments revient à [Tea 00] qui a proposé tout au début un ensemble de moments basés sur la base de polynômes orthogonaux appelées par la suite moments de Zernike. Il est bien connu qu'une fonction discrète d'image puisse être reconstruite par les moments de Zernike. En particulier les moments de Zernike se sont avérés intéressants quant à la robustesse par rapport au bruit et l'invariance par rapport à la rotation. Un ensemble relativement petit de moments de Zernike peut caractériser la forme globale d'un objet, Les moments d'ordre bas représentent la forme globale d'un objet et celles d'ordre plus supérieur le détail de cet objet. Plus on fait appel à des moments de Zernike d'ordre plus élevé, plus est la précision de reconstruction de l'image de l'objet en cours du traitement. La reconstruction dépend du nombre de polynômes de Zernike impliqués. Pour des images binaires la reconstruction peut être réalisée avec un ordre modeste. Cependant, pour les images à niveaux de gris l'ordre de reconstruction est en général plus important. En outre, les valeurs de la partie imaginaire des moments de Zernike pour des images de niveaux gris sont plus hautes que pour des images binaires.

3-3-2 Moments complexes de Zernike

Les moments de Zernike sont fréquemment utilisés pour plusieurs tâches en traitement d'image. Le polynôme de Zernike d'ordre (p,q) est défini sur un disque de **rayon unité** par l'équation :

$$V_{pq}(x, y) = R_{pq}(\rho) \exp(jq\theta) \quad (3.14)$$

Où $\rho = \sqrt{x^2 + y^2}$ est la longueur du vecteur depuis l'origine jusqu'au pixel (x,y), $\theta = \arctan (y/x)$ est l'angle entre ce vecteur et l'axe des abscisses.

- Tous les traitements à base de polynôme de Zernike sont considérés à partir d'un disque unité. Avant tout traitement, il faut donc ramener l'image à l'intérieur de ce cercle. Ceci est réalisé par une normalisation des rayons de tous les pixels de l'image : Partant du centre de gravité, on balaye tous les pixels de l'image à la recherche du plus grand rayon. Ce rayon ainsi déterminé est considéré comme unitaire. Tout autre rayon est divisé par ce rayon maximum pour inclure le pixel correspondant dans le cercle unité ainsi défini.

Le polynôme de Zernike $R_{pq}(\rho)$ est donné par l'expression :

$$R_{pq}(\rho) = \sum_{l=0}^{(p-|q|)/2} (-1)^l \frac{(p-l)!}{l! \left(\frac{p+|q|}{2} - l\right)! \left(\frac{p-|q|}{2} - l\right)!} \rho^{p-2l} \quad (3.15)$$

Une image $f(x,y)$ peut être développée en polynôme de Zernike, sur le disque unité, comme suit :

$$f(x, y) = \sum_{p=0}^{\infty} \sum_{q=-p}^p r_p A_{pq}(\rho) V_{pq}(x, y) \quad (3.16)$$

Le nombre de moment représente le nombre de couple (p,q) tels que:

$$\text{Où } \begin{cases} (p - |q|) & \text{pair} \\ (p - |q|) \geq 0 \end{cases}$$

A_{pq} est le moment de Zernike d'ordre p avec une répétition q.

$$A_{pq} = \iint f(x, y) V_{pq}^*(x, y) dx dy \quad (3.17)$$

La version en coordonnées polaires de A_{pq} est :

$$A_{pq} = \int_0^{2\pi} \int_0^1 f(\rho \cos \theta, \rho \sin \theta) R_{pq}(\rho) \exp(-jq\theta) \rho d\rho d\theta \quad (3.18)$$

Les moments de Zernike sont invariants par rapport à la rotation. Si la fonction $f(x, y)$ est tournée d'un angle α alors les moments de Zernike A'_{pq} de cette image (tournée) sont données par :

$$A'_{pq} = A_{pq} e^{-jq\alpha} \quad (3.18)$$

Il paraît donc que l'amplitude des moments de Zernike peut être utilisée comme caractéristique invariable par rapport à la rotation. La version discrète de A_{pq} est donnée par :

$$\hat{A}_{pq} = \sum_{x_i^2 + y_i^2 \leq 1} \sum h_{pq}(x_i, y_i) f(x_i, y_i) \quad (3.19)$$

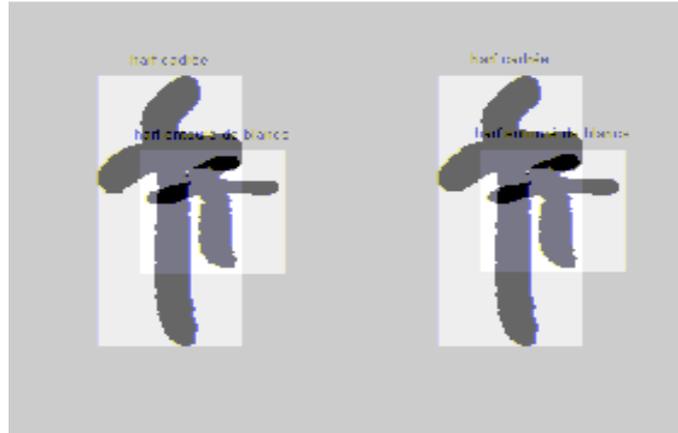
Où

$$h_{pq}(x_i, y_i) = \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} \int_{y_i - \frac{\Delta y}{2}}^{y_i + \frac{\Delta y}{2}} V_{pq}^*(x, y) dx dy \quad (3.20)$$

3-3-3 Application de la méthode des moments de Zernike sur une image

Voici un exemple des sept moments de Zernike extrait de trois caractères arabes.

➤ **moments de Zernike d'ordre 9**



ALIF cadrée (64×64)

6.48267E+00 4.66546E-02 2.98927E+00 3.56259E-02 4.80250E-02 4.55287E-02
 6.71584E-01 2.60542E-02 6.54544E-02 7.38996E-02 3.47393E-02 2.17921E-02
 1.52007E-01 1.02016E-01 7.20295E-03 1.10217E-02 7.56861E-02 9.52039E-02
 1.28191E-02 3.21720E-02 1.89829E-01 1.10088E-01 1.10221E-01 3.94962E-03
 9.96129E-03 1.19179E-01 1.14716E-01 4.21916E-02 1.95689E-02 3.71160E-04



BA cadrée (64×64)

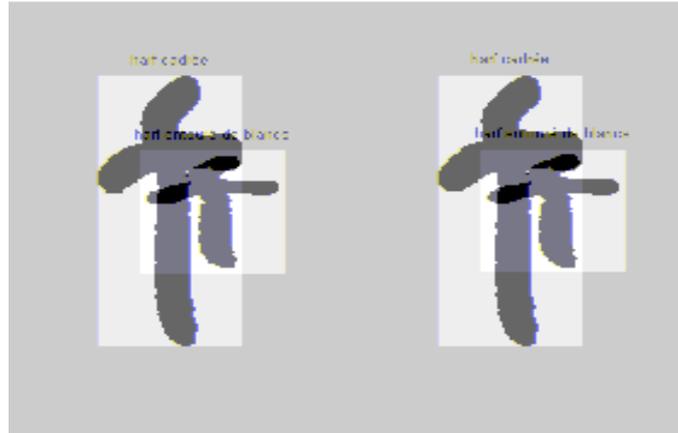
2.39766E+00 7.21348E-02 1.05568E+00 4.00765E-02 3.84958E-02 7.24040E-02
 3.59160E-01 2.55155E-02 1.25579E-01 6.71805E-02 4.99140E-02 6.04977E-02
 7.50686E-01 9.94227E-03 1.11038E-01 9.69236E-02 3.00454E-02 2.81494E-02
 1.01177E-01 2.01998E-02 2.53688E-01 3.31683E-02 1.06584E-01 1.00736E-01
 4.25520E-02 4.95867E-02 3.77823E-02 4.49430E-02 6.51177E-02 4.52994E-02



AIN cadrée (64×64)

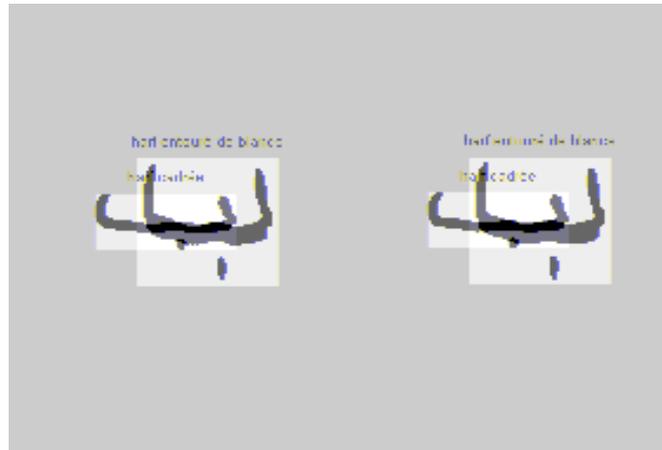
1.89316E+00	5.10457E-02	1.29161E+00	4.15040E-02	7.94517E-02	3.09606E-03
3.94129E-01		8.35166E-02	4.05646E-03	7.74624E-02	1.00695E-02
1.23024E-03	3.70254E-01		8.28911E-02	1.44618E-02	3.07572E-04
5.51468E-02	1.91786E-02	5.71540E-03		9.95457E-05	
	6.93927E-01		4.97871E-02	2.65342E-02	1.61649E-03
7.54275E-05		3.07813E-02		2.71780E-02	1.43033E-02
	5.46621E-04	3.10769E-05			

➤ **moments de Zernike d'ordre 11**



ALIF cadrée (64×64)

6.48267E+00 4.66546E-02 2.98927E+00 3.56259E-02 4.80250E-02 4.55287E-02
 6.71584E-01 2.60542E-02 6.54544E-02 7.38996E-02 3.47393E-02 2.17921E-02
 1.52007E-01 1.02016E-01 7.20295E-03 1.10217E-02 7.56861E-02 9.52039E-02
 1.28191E-02 3.21720E-02 1.89829E-01 1.10088E-01 1.10221E-01 3.94962E-03
 9.96129E-03 1.19179E-01 1.14716E-01 4.21916E-02 1.95689E-02 3.71160E-04
 3.56988E-01 1.01596E-01 1.91227E-01 1.95808E-02 1.62484E-02 7.22042E-03
 1.60727E-01 1.63584E-01 5.77132E-02 5.40338E-02 1.42133E-02
 2.90317E-02



BA cadrée (64×64)

2.39766E+00 7.21348E-02 1.05568E+00 4.00765E-02 3.84958E-02 7.24040E-02
 3.59160E-01 2.55155E-02 1.25579E-01 6.71805E-02 4.99140E-02 6.04977E-02
 7.50686E-01 9.94227E-03 1.11038E-01 9.69236E-02 3.00454E-02 2.81494E-02
 1.01177E-01 2.01998E-02 2.53688E-01 3.31683E-02 1.06584E-01 1.00736E-01

4.25520E-02 4.95867E-02 3.77823E-02 4.49430E-02 6.51177E-02 4.52994E-02
 2.47025E-01 1.67086E-02 9.76461E-02 9.09993E-02 3.72504E-02 4.43313E-02
 5.21758E-02 3.75795E-02 7.10493E-02 2.77267E-02 7.22754E-02
 3.26364E-02



AIN cadrée (64×64)

1.89316E+00 5.10457E-02 1.29161E+00 4.15040E-02 7.94517E-02 3.09606E-03
 3.94129E-01 8.35166E-02 4.05646E-03 7.74624E-02 1.00695E-02 1.23024E-03
 3.70254E-01 8.28911E-02 1.44618E-02 3.07572E-04 5.51468E-02 1.91786E-02
 5.71540E-03 9.95457E-05 6.93927E-01 4.97871E-02 2.65342E-02 1.61649E-03
 7.54275E-05 3.07813E-02 2.71780E-02 1.43033E-02 5.46621E-04 3.10769E-05
 5.43172E-01 7.70745E-02 3.04340E-02 4.40733E-03 5.16094E-04 1.70108E-05
 2.28821E-02 2.99010E-02 2.49970E-02 1.54859E-03 2.07810E-04
 9.99862E-06

3-4 Descripteurs de Fourier

Le contour d'une image est tout simplement une fonction à deux dimensions. Il peut être considéré comme l'empreinte d'une image ce qui veut dire qu'il peut être utilisé pour reconnaître cette image. A part certaines caractéristiques telles la longueur, l'orientation des pixels etc... qui peuvent être retiré directement de ce contour, on peut ramener ce contour vers un autre espace pour lui faire sortir plus d'information concernant l'objet en cours du traitement. La transformée de Fourier est l'un des outils adéquats pour opérer ce changement d'espace. Elle est appliquée au contour de

l'image exactement comme s'il s'agit de son application sur une quelconque fonction à deux dimensions. L'image, représentée par son contour, est ainsi ramenée depuis l'espace cartésien vers l'espace de Fourier. Les distances sont par conséquent ramenées aux fréquences spatiales. Les fameuses expressions, bien connues, de la transformée de Fourier sont évidemment toujours valables.

Considérons donc l'image des caractères arabes (figure 3-1). On leur applique d'abord un détecteur de contour [Leg 01] (Canny, Sobel, Robet,...) comme le montre la figure 3-2. On note bien que ces contours sont fermés ce qui signifie en d'autre terme que ces contours sont une sorte de boucles et par conséquent on est en présence d'une périodicité. En langage de traitement de signal, ces contours sont donc des fonctions périodiques à lesquelles il est tout autorisé d'appliquer la transformée de Fourier. C'est donc à ces contours qu'on applique maintenant cette transformée de Fourier comme suit :

ا ب ح د ر ط ك ل م ن ص
ع ف ق س ه و ي

Figure 3-1 : caractères arabes

ا ب ح د ر ط ك ل م ن



Figure 3.2 : Ensemble des lettres arabe avec un seul contour

La transformée de Fourier discrète (DFT) d'une séquence $f(j,k)$ a deux dimensions est donnée par :

$$F(u,v) = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j,k) \exp\left[-\frac{2\pi i}{N}(uj+vk)\right] \quad (3.21)$$

Où : $i = \sqrt{-1}$ et la transformée de Fourier inverse est donnée par :

$$f(j,k) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v) \exp\left[\frac{2\pi i}{N}(uj+vk)\right] \quad (3.22)$$

Les variables (u,v) sont les fréquences spatiales. La transformée de Fourier possède des propriétés intéressantes :

La composante spectrale à l'origine est :

$$F(0,0) = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j,k) \quad (3.23)$$

Elle est égale $1/N$ fois la moyenne spatiale de l'image.

Considérons la transformation de variable suivante :

$$u = u + mN, v = v + nN$$

où m et n sont des constantes. L'équation (3.21) donne :

$$F(u+mN, v+nN) = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j, k) \exp\left[-\frac{2\pi i}{N}(uj+vk)\right] \exp[-2\pi i(mj+nk)] \quad (3.24)$$

Pour tout entier m et n la deuxième exponentielle est égale à 1 et le domaine de la transformée se trouve périodique :

$$F(u+mN, v+nN) = F(u, v) \quad (3.25)$$

3-4-1 Calcul des descripteurs de Fourier

3-4-1-1 La méthode complexe

Supposant que le contour d'une forme particulière a N pixels numérotés de 0 à N-1. Le *k*-ième pixel au long du contour a comme position (*xk*, *yk*). Ainsi on peut décrire le contour par deux équations :

$$x(k) = x_k$$

$$y(k) = y_k$$

Si on calcule la transformée de Fourier de chaque fonction, on aboutit à deux spectres de fréquence :

$$a_x = F(x(k)), \quad a_y = F(y(k))$$

Pour un nombre fini de pixels d'un contour on utilise simplement la Transformée de Fourier Discrète (DFT). Rappelons que la DFT traite le signal comme périodique, dans notre cas le contour est lui-même périodique puisqu'il est fermé.

L'étape suivante est de considérer les coordonnées (*x*, *y*) du point non comme des coordonnées Cartésiennes mais comme des coordonnées dans un plan complexe :

$$u(n) = x(n) + jy(n) \quad n=0, 1, \dots, N-1$$

Cette équation, pour un contour fermé, est périodique de période N . Sa représentation en DFT est :

$$u(n) = \frac{1}{N} \sum_{k=0}^{N-1} a(k) \exp\left(\frac{2j\pi kn}{N}\right) \quad 0 \leq n \leq N-1 \quad (3.26)$$

$$a(k) = \frac{1}{N} \sum_{n=0}^{N-1} u(n) \exp\left(-\frac{2j\pi kn}{N}\right) \quad 0 \leq k \leq N-1 \quad (3.27)$$

Les coefficients $a(k)$ sont les descripteurs de Fourier (FDs) du contour. Les FDs sont coefficients infinis de la série de Fourier développée de la fonction continue $u(t)$ d'un contour donné. l'équation 4 nous donne :

$$a(k) = \sum_{n=0}^{N-1} (x(n) + jy(n)) \left(\cos\left(-\frac{2\pi kn}{N}\right) + j \sin\left(-\frac{2\pi kn}{N}\right) \right)$$

$$a(k) = \sum_{n=0}^{N-1} \left[(x(n) \cos\left(\frac{2\pi kn}{N}\right) + y(n) \sin\left(\frac{2\pi kn}{N}\right)) \right. \\ \left. + j \sum_{n=0}^{N-1} \left[-(x(n) \sin\left(\frac{2\pi kn}{N}\right) + y(n) \cos\left(\frac{2\pi kn}{N}\right)) \right] \right]$$

$$a_{\text{Réel}}(k) = \sum_{n=0}^{N-1} \left[(x(n) \cos\left(\frac{2\pi kn}{N}\right) + y(n) \sin\left(\frac{2\pi kn}{N}\right)) \right] \quad (3.28)$$

$$a_{\text{Im}}(k) = \sum_{n=0}^{N-1} \left[-(x(n) \sin\left(\frac{2\pi kn}{N}\right) + y(n) \cos\left(\frac{2\pi kn}{N}\right)) \right] \quad (3.29)$$

La méthode des Descripteurs de Fourier n'agit que sur une seule courbe à la fois, de plus cette courbe doit d'être fermée. Pour cette raison l'ensemble des caractères arabes sur lesquelles on peut appliquer cette méthode, après extraction du contour, est limité à l'ensemble montré sur la figure 3.2 et l'entrée au réseau de neurone est un vecteur composé des FD d'un contour unique.

Comme pour les moments invariants, les coefficients obtenus à partir de cette transformation vont jouer le rôle de l'empreinte du contour de l'image. Ces coefficients sont ce qu'on appelle Descripteurs de Fourier (DF).

Une fois ces coefficients (descripteurs) obtenus on peut les utiliser pour reconnaître les objets. Il est à noter cependant que le nombre de descripteurs utilisés est fonction du domaine d'application. Pour déterminer ce nombre on opère en sens inverse : On trouve les coefficients de Fourier de l'objet à traiter. On reconstitue cet objet à partir de ces coefficients par application de la transformée de Fourier inverse. On augmente le nombre de coefficients intervenant dans cette transformation inverse jusqu'à obtenir satisfaction quant à la qualité de l'objet reconstitué.

3-4-2 Propriété des descripteurs de Fourier

Les descripteurs de Fourier calculés selon l'équation (3.28) et (3.29) sont invariables à la translation, rotation et changement d'échelle.

3-4-2-1 La translation

La translation peut être décrite comme addition de tout nombre complexe z à chaque élément des éléments de z_n qui contient les coordonnées de frontière. la transformée de Fourier discrète est :

$$\sum_{n=0}^{N-1} (z_n - z)e^{-2\pi i k n / N} = \sum_{n=0}^{N-1} z_n e^{-2\pi i k n / N} + z \sum_{n=0}^{N-1} e^{-2\pi i k n / N} \quad (3.30)$$

Le premier élément (descripteur) de la sommation de l'équation 3.30 au-dessus est égal à l'élément non translaté de z_n , alors que le deuxième élément est

$$\sum_{n=0}^{N-1} e^{-2\pi i k n / N} = \begin{cases} \frac{1 - e^{-2\pi i k N}}{1 - e^{-2\pi i k}} = 0, & k \neq 0 \\ N - 1, & k = 0 \end{cases} \quad (3.31)$$

Selon équation (3.31) la translation affectera seulement la valeur du premier élément de la transformée de Fourier discrète, et l'invariance à la translation peuvent être obtenues simplement en tronquant le premier élément z_0 .

3-4-2-2 La rotation

La rotation d'un angle φ peut être décrite comme la multiplication de chaque élément de z_n avec $e^{-\varphi i}$. La transformée de Fourier discrète de z_n est

$$\sum_{n=0}^{N-1} (z_n e^{\varphi i}) e^{-2\pi i k n / N} = e^{\varphi i} \sum_{n=0}^{N-1} z_n e^{-2\pi i k n / N} \quad (3.32)$$

D'après l'équation 3.32 On voit que la rotation affecte la phase de tous les descripteurs.

Comme $|e^{-\varphi i}| = 1$, pour réaliser l'invariance de la rotation, il est suffisant de prendre la valeur l'absolu de chaque élément (descripteur).

3-4-2-3 Changement d'échelle

Le Changement d'échelle peut être décrite comme la multiplication de chaque élément de z_n avec la constante c . La transformation discrète de Fourier de z_n est :

$$\sum_{n=0}^{N-1} (cz_n) e^{-2\pi i kn/N} = c \sum_{n=0}^{N-1} z_n e^{-2\pi i kn/N} \quad (3.33)$$

On note à partir de cette équation que chaque élément de la TF discrète est multiplié par c.

Soit « I » une image quelconque. $a_0, a_1, a_2, a_3, \dots, a_n$ les modules des coefficients correspondants aux différentes harmoniques, de la transformée de Fourier, de cette image. Soit « αI » la version de « I » affecté d'un facteur d'échelle α . Soit « βI » la version de « I » affecté d'un facteur d'échelle β . D'après l'expression (3.33) ci-dessus les amplitudes des transformées de Fourier correspondant aux images αI et βI sont respectivement :

$$\alpha a_0, \alpha a_1, \alpha a_2, \alpha a_3, \dots, \alpha a_n \text{ et } \beta a_0, \beta a_1, \beta a_2, \beta a_3, \dots, \beta a_n$$

Prenons un élément quelconque, le deuxième élément par exemple, de chaque vecteur et déterminons un nouveau vecteur par une division de tous les éléments par l'élément choisi c'est dire :

$$V = [\alpha a_1, \alpha a_2, \alpha a_3, \dots, \alpha a_n]^T \rightarrow \frac{V}{\alpha a_1} = \left[1, \frac{a_2}{a_1}, \frac{a_3}{a_1}, \dots, \frac{a_n}{a_1} \right]^T \quad (3.34)$$

$$W = [\beta a_1, \beta a_2, \beta a_3, \dots, \beta a_n]^T \rightarrow \frac{V}{\beta a_1} = \left[1, \frac{a_2}{a_1}, \frac{a_3}{a_1}, \dots, \frac{a_n}{a_1} \right]^T \quad (3.35)$$

De ces deux expression on note que ce rapport est bien fixe c'est-à-dire invariant par rapport à un changement d'échelle c'est ce qu'on cherche.

- Noter bien qu'on peut choisir n'importe quel élément comme diviseur. Exclu de ce choix le premier élément pour des raisons de dépendance, du module de cet élément, de la translation ;

- Noter également que ce ne sont pas les descripteurs de Fourier qui sont réellement utilisés pour la reconnaissance. C'est plutôt leur image représentée par le rapport de chacun de ces coefficients sur l'un d'entre eux.
- Noter surtout que le passage dans le plan de Fourier depuis le plan de l'image, ne prend en considération que le contour principal de cette image. Ils sont exclus donc de ce transfert certains paramètres de l'écriture arabe tels les points et les contours internes (contours secondaires). Cela provient de la définition de la transformée de Fourier elle-même. En effet cette transformée manipule un seul contour fermé à la fois. Dans le cas d'images à plusieurs contours comme c'est le cas pour l'écriture arabe, il faut choisir celui qui fera l'objet de cette transformation. Il est choisi tout naturellement le contour le plus représentatif de l'image complète : le contour principal ;
- Pour une reconnaissance de tout le caractère, il est tout évident d'ajouter ces paramètres, misent à l'écart pendant ce transfert entre repères c'est-à-dire les points. Pour ce paramètre, et spécialement pour l'écriture arabe, on peut même ajouter deux caractéristiques : le nombre de points et leur position par rapport au centre de gravité.
- On peut pour certains d'autres lettres ajouter d'autres caractéristiques telles le nombre de contours.

3-4-2-4 Choix du point de départ z_0

Se déplacer à partir d'un point de départ le long du contour, il peut être décrit comme la multiplication par $e^{2\pi i k m / N}$ d'où m correspond au nombre de points entre l'ancien et le nouveau point de départ. On peut écrire :

$$\sum_{n=0}^{N-1} z_{(n-m)} \bmod N e^{-2\pi i k m / N} = e^{-2\pi i k m / N} \sum_{n=0}^{N-1} z_{(n-m)} \bmod N e^{-2\pi i k (n-m) / N}$$

La fonction $e^{2\pi i k m / N}$ est une fonction périodique et l'équation (6 dessus) peut être réécrite

$$e^{2\pi i k m / N} \sum_{n=0}^{N-1} z_{(n-m) \bmod N} e^{-2\pi i k ((n-m) \bmod N) / N} = e^{2\pi i k m / N} z_k \quad (3.36)$$

Le choix du point départ n'affecte que la phase des descripteurs.

Comme nous prenons la valeur absolue des coefficients discrets de la transformée de Fourier on calculant les descripteurs de Fourier, le point de départ choisi n'affectera pas leur valeur parce que $|e^{2\pi i k m / N}| = 1$.

La méthode des Descripteurs de Fourier est appliquée sur un nombre limité de caractères arabes, il s'agit des caractères qui ont une seule courbe à la fois, de plus cette courbe doit d'être fermé. Cette méthode n'est pas suffisante pour la détermination des Descripteurs de Fourier de tous les caractères. Donc il est nécessaire d'ajouter des paramètres aux vecteurs d'entrée, soit par exemple les informations sur l'ensemble de tous les contours, le contour principale, le nombre de contours secondaires qui représente les points du caractère ou les régions interne par exemple les caractères ha et waw (voir figure 3.4), et la position des contours secondaires par rapport au contour principale, cette combinaison de caractéristique va nous permettre de différencier entre les caractères ayant un corps similaire tel que les caractères ayant même contour principale et se distinguent par le nombre de points 'Ba'(2contours), 'Ta'(3contours), et 'Tha' (4contours), (voir figure 3.3).



Figure 3.3 : Trois caractères ayant le même contour principal et se distinguent par le nombre de points



Figure 3.4 : deux caractères (ha et waw) ayant le nombre de contours internes différents

Après cette décomposition le vecteur caractéristique est représenté par les paramètres suivants :

- 1- Les FDs du contour principal.
- 2- Le nombre de contours secondaires.
- 3- La position des contours secondaires par rapport au contour principale (position centre de gravité de chaque contour par rapport au contour principal)
- 4- nombres de contours internes.

3-5 Caractérisation du contour par le Code de Freeman

Un suivi de contour est alors réalisé de manière à obtenir les directions entre pixels successifs. Ces derniers sont codés par l'intermédiaire du code de Freeman [10] [11]. Les occurrences de ces directions sont alors dénombrées pour chaque zone individuellement. Nous présentons dans le tableau ci-dessous, des exemples de cette technique d'extraction. Pour ces caractéristiques, la normalisation s'effectue en divisant chaque compteur de direction d'une zone par le nombre de pixels de contours présents dans cette zone. Le vecteur des caractéristiques associées à cet espace de représentation contient alors huit composantes par zone.

Dans notre application nous avons choisi le vecteur des caractéristiques des histogrammes de direction. L'extraction de ces caractéristiques est réalisée sur l'image des contours de chaque caractère. Un exemple des caractères (ب, ج, ع, ل) est donné dans le tableau ci-dessous :

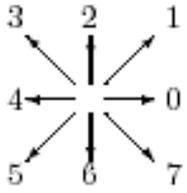
Caractères	ج	ع	ح	ب
	0.092692	0.18443	0.19466	0.2807
	0.0802	0.1286	0.1272	0.0421
	14	0.1099	3	05
	0.2780	8	0.0903	0.1087
	7	0.0862	31	7
	0.0481	94	0.0674	0.0614
	28	0.1624	3	04
	0.0909	4	0.2302	0.2877
	09	0.1438	8	2
	0.0802	2	0.1132	0.0438
14	0.1049	3	6	
0.2834	1	0.0877	0.1052	
2	0.0778	86	6	
0.0445	34	0.0865	0.0666	
63		14	67	

Tableau 3.1 : Caractéristiques des histogrammes de direction

3-6 Caractérisation par squelettisation

La squelettisation est une opération qui nécessite un traitement de l'information avant la reconnaissance. En effet, le squelette est une structure de données brutes que l'on ne peut pas exploiter directement. Elle nécessite une certaine interprétation dont l'objectif est de normaliser l'ensemble des données en un tout cohérent, ceci indépendamment de l'étape de reconnaissance. Plusieurs techniques ont été adoptées à ce niveau. L'une d'entre elles découpe le squelette en ses segments caractéristique. Ces segments construiront par la suite l'identité de l'image « caractère » et peuvent servir pour le reconnaître.

Une autre technique [Abd 02] remplace le squelette par un ensemble de points caractéristiques. Il s'agit des points fins, des points de croisements, des points de

branchements etc. Ces points constitueront également par la suite un vecteur d'identité qui servira dans l'étape de reconnaissance

Prenons un exemple qui détermine les points caractéristiques du squelette du caractère (sad). Trois types de points caractéristiques peuvent être extraits du squelette du tracé (figure 1).

- Le point isolé : c'est le pixel noir qui ne possède aucun voisin.
- Le point de fin de trait (extrémité) : c'est le pixel noir qui possède un seul voisin de même type.
- Le point de branchement: c'est le pixel noir qui possède trois voisins de même type.
- Le point de croisement : c'est le pixel noir qui possède quatre voisins de même type.

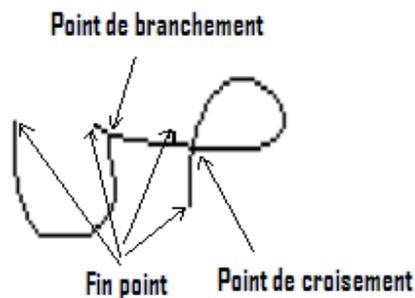


Figure 3.5 - Points caractéristiques du squelette du caractère (sad)

Un algorithme de suivi de squelette permet de déterminer les segments d'un caractère. Ces segments sont classés selon trois catégories (figure 3.5):

- Branche 1 : c'est un trait qui se situe entre deux points de fin de trait ou un point de fin de trait et un point de branchement (ou de croisement).
- Branche 2: c'est un trait de liaison qui se situe entre deux points de branchement (ou croisement) ou entre un point de branchement et un point de croisement.

- Branche 3 (boucle) : c'est un trait qui représente une boucle, autrement qui part et revient au même point de départ.

Cette segmentation se fait par un suivi de squelette en partant des points caractéristiques et en cherchant à chaque fois le point voisin suivant. Pour cela, cette segmentation permet une première disposition des points de chaque segment. Pour faciliter cette opération, on commence par l'extraction des branches de type 1 puis les branches de type 2 et 3 en éliminant à chaque fois les segments localisés sur le squelette. Dans la figure ci-dessous on a une boucle de type 3, une branche de type 2 et quatre branches de type 1 (nombre de fin de points). Ces points constitueront par la suite un vecteur d'identité qui servira dans l'étape de reconnaissance



Figure 3.6 - Segmentation du caractère (sad)

3-8 La représentation du contour par une Chaîne de code (chain code)

Dans cette représentation le vecteur direction entre deux pixels successifs de la frontière est codé (encoded). Par exemple, une chaîne de code couramment utilisé utilise huit directions qui peuvent être codés par trois bits comme le représente la figure 3.7 ci-dessus.

Ainsi pour l'orientation des pixel du contour on a : (A), 76010655432421

Pour le chaîne de code : A 111 110 000 001 000 110 101 101 110 011 010 100 010 001

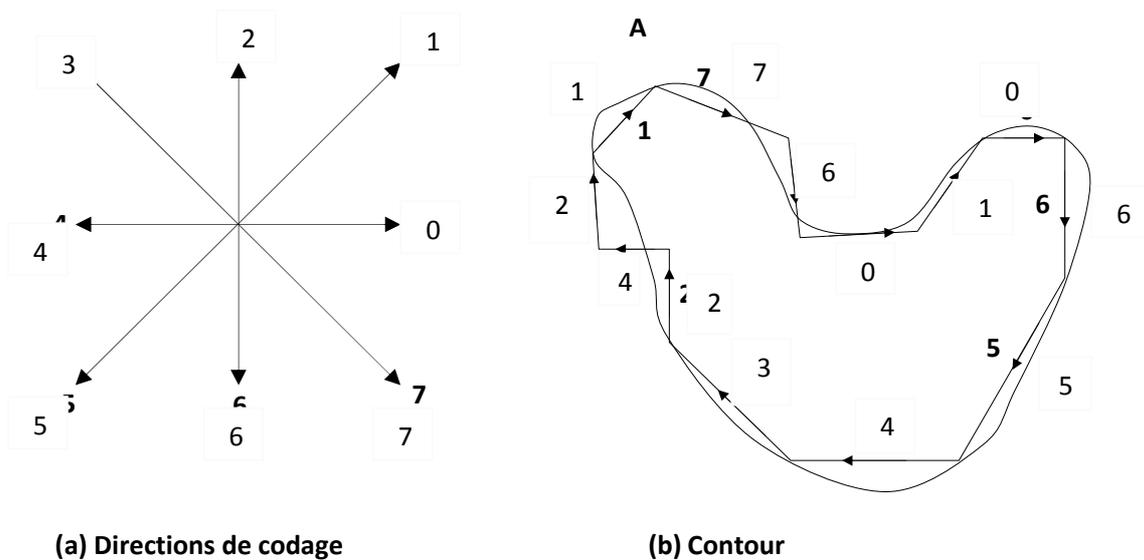


Figure 3.7 - Chain code pour la représentation du contour

Pratiquement la chaîne des codes contient l'adresse du pixel de départ suivi par une chaîne de mots représentant les codes des différentes directions. L'algorithme est donc :

Algorithme

Prendre un point quelconque de la frontière comme point de départ ;

Trouver le pixel suivant et coder son orientation ;

Continuer jusqu'à avoir balayé tous les pixels du contour (frontière).

3-8 La représentation en lignes raccordées (Fitting line segment)

Les lignes droites fournissent une approximation simple de la courbure des frontières (curve boundary). Un algorithme séquentiel intéressant qui utilise les lignes droites pour le suivi de courbe est le suivant :

Algorithme

Approximer la courbe par la segment de droite joignant ces limite (A,B);

Si la distance entre ce segment de droite et le point le plus éloigné de la courbe (C) est supérieure à un seuil prédéterminé, joindre AC et BC ;

Répéter la procédure pour les nouveaux segments AC et BC et continuer jusqu'à atteindre la précision voulue.

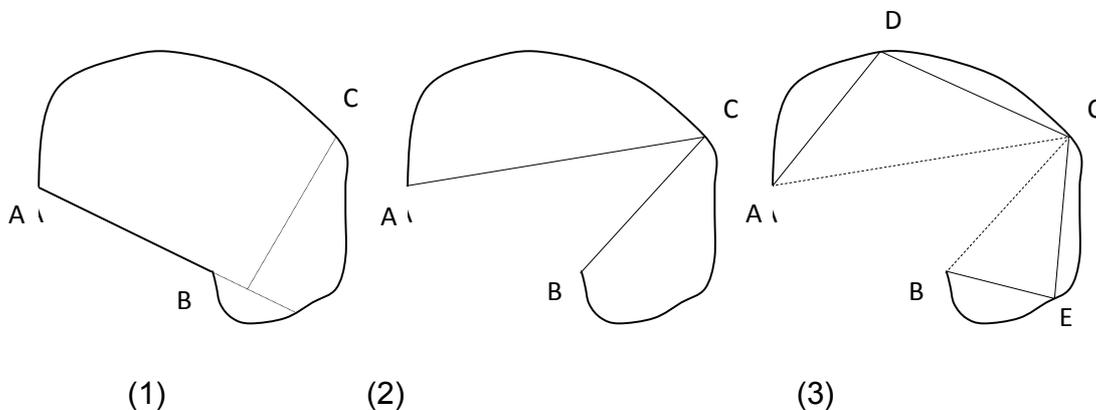


Figure 3.8 - Approximation successive par des lignes droites

3-9 les moments affines invariants (AMI)

Nous avons choisi un modèle de base invariant pour une identification de marques. Les moments affines invariants (MAI) sont les descriptions des caractéristiques d'un objet, qui restent invariable dans les situations, quand un objet est tordu. Les Invariants différent dans l'ensemble de déformations qu'elles sont invariables à plusieurs genres de caractéristiques ont été employés pour la description d'objet dans les travaux récents, tels que les vecteurs de forme [Pel 81], des matrices de forme [Gos 85], des descripteurs de Fourier [Zah 72], des invariants différentiels [Weiss 88] et des invariants de moment [Hu 62]; [Bel 91]; [Pro 92]. La plupart d'entre eux peuvent décrire les objets binaires seulement et, d'ailleurs, sont invariable à la translation, rotation et changement d'échelle. Récemment, [Flu 93] ont dérivé AMIs, qui sont invariables sous transformations affine :

$$u = a_0 + a_1x + a_2y$$

$$v = b_0 + b_1x + b_2y$$

(x,y) et (u, v) sont les coordonnées dans le plan d'image avant et après la transformation, respectivement. L'invariance de l'AMIs a été démontrée théoriquement et expérimentalement en (voir [Flu 93] et [Suk 94]).

Les AMIs peuvent identifier correctement les objets égaux, tordus avec la projection de perspective.

$$u = \frac{a_0 + a_1x + a_2y}{1 + c_1x + c_2y}$$

$$v = \frac{b_0 + b_1x + b_2y}{1 + c_1x + c_2y}$$

Une transformation de perspective décrit exactement la projection d'un objet à 3D d'une position générale dans le plan d'image 2D, une fois capturée par un appareil-photo de trou d'épingle. Quand la distance entre l'appareil-photo et l'objet est

sensiblement plus grande que la taille de l'objet, les projectifs transforment peuvent être bien rapprochés par l'affinage transforment et l'AMIs peut fournir une classification correcte des objets. Ce type de stabilité d'AMIs était la raison de notre décision de les employer. Ils sont assez robuste même dans le cas de l'additif zéro-signifiant le bruit aléatoire. Ces propriétés gentilles d'AMIs sont prouvées dans la section 5, où les expériences que nous avons faites avec la base de données donnée de marque et le modèle proposé d'identification sont décrites.

Les six premiers AMIs suivent en forme explicite:

$$I_1 = \frac{1}{\mu_{00}^4} (\mu_{20}\mu_{02} - \mu_{11}^2) \quad (3.37)$$

$$I_2 = \frac{1}{\mu_{00}^{10}} (\mu_{30}^2\mu_{03}^2 - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 4\mu_{30}\mu_{12}^3 + 4\mu_{03}\mu_{21}^3 - 3\mu_{21}^2\mu_{12}^2) \quad (3.38)$$

$$I_3 = \frac{1}{\mu_{00}^7} (\mu_{20}(\mu_{21}\mu_{03} - \mu_{21}^2) - \mu_{11}(\mu_{30}\mu_{03} - \mu_{21}\mu_{12}) + \mu_{02}(\mu_{30}\mu_{03} - \mu_{21}^2)) \quad (3.39)$$

$$\begin{aligned} I_4 = \frac{1}{\mu_{00}^{11}} & (\mu_{20}^3\mu_{03}^2 - 6\mu_{20}^2\mu_{11}\mu_{12}\mu_{03} - 6\mu_{20}^2\mu_{02}\mu_{21}\mu_{03} + 9\mu_{20}^2\mu_{02}\mu_{12}^2 + 12\mu_{20}\mu_{11}^2\mu_{21}\mu_{03} \\ & + 6\mu_{20}\mu_{11}\mu_{02}\mu_{12}\mu_{30}\mu_{03} - 18\mu_{20}\mu_{11}\mu_{02}\mu_{21}\mu_{12} - 8\mu_{11}^3\mu_{30}\mu_{03} - 6\mu_{02}^2\mu_{20}\mu_{12}\mu_{30} \\ & + 9\mu_{02}^2\mu_{20}\mu_{21}^2 + 12\mu_{11}^2\mu_{02}\mu_{30}\mu_{12} - 6\mu_{02}^2\mu_{11}\mu_{21}\mu_{30} \\ & + \mu_{02}^3\mu_{30}^2) \end{aligned} \quad (3.40)$$

$$I_5 = \frac{1}{\mu_{00}^6} (\mu_{40}\mu_{04} - 4\mu_{31}\mu_{13} + 3\mu_{22}^2) \quad (3.41)$$

$$I_6 = \frac{1}{\mu_{00}^9} (\mu_{40}\mu_{04}\mu_{22} + 2\mu_{31}\mu_{22}\mu_{13} - \mu_{40}\mu_{13}^2 - \mu_{04}\mu_{31}^2 - \mu_{22}^2) \quad (3.42)$$

où μ_{pq} est le moment centré d'ordre $(p + q)$. il est défini comme :

$$\mu_{pq} = \iint (x - x_t)^p (y - y_t)^q f(x, y) dx dy$$

(3.43)

Ou (x_t, y_t) sont les coordonnées du centre de gravité de l'objet. les détails de la dérivation peut être trouvée dans [Flu 93].

CHAPITRE 4

ETAT DE L'ART DE LA RECONNAISSANCE

4-1 Caractéristiques de la langue arabe

La langue arabe a vu ses racines naître dans la péninsule Arabique et les premières traces de l'écriture arabe, telle qu'on la connaît ne remontent qu'au VI^e siècle. Cette langue a vécu une expansion extrêmement rapide et a relié un immense empire recouvrant le Proche-Orient, l'ensemble de l'Afrique du nord, l'Espagne et la Sicile. L'expansion et le développement de cette langue sont intimement liés à la naissance et la diffusion de l'islam. Néanmoins cette langue ne s'est pas limitée au texte coranique, mais elle est devenue une langue de culture, de philosophie, de sciences et de techniques allant jusqu'à supplanter les autres langues locales lors de l'expansion arabo-musulmane. La langue Arabe est une langue parlée par à peu près 300 millions de personnes. Elle est la langue officielle ou figure parmi les langues officielles de 19 pays. Il y a deux types d'écritures possibles pour la langue arabe :

- l'écriture classique qui correspond à l'écriture du Coran et la littérature classique,
- l'écriture universelle du monde arabe actuel.

En ce qui concerne l'arabe parlé, chaque pays ou région possède un arabe parlé dialectal. Il arrive que cet arabe parlé dialectal recouvre en fait plusieurs dialectes différents tous issus de l'arabe classique. L'arabe dialectal se subdivise en trois grands groupes :

- Les dialectes arabiques : parlés dans la péninsule arabique,
- Les dialectes maghrébins : algérien, marocain et tunisien,
- Les dialectes proche-orientaux : égyptien et syro libano palestinien.

L'arabe appartient à la famille des langues sémitiques comme l'hébreu et l'araméen, au sein desquelles essentiellement les consonnes sont représentées en écriture, néanmoins nous notons la présence de voyelles.

- L'alphabet arabe est composé de 28 lettres ;
- Les mots sont écrits sur une ligne de base horizontale dite encore ligne de référence ou d'écriture ;
- Cette écriture est de nature cursive et se fait de la droite vers la gauche ;
- Les chiffres sont écrits de la gauche vers la droite ;
- les formes minuscules et majuscules des lettres sont inexistantes ;
- les pays arabes font usage de deux types de graphies pour représenter les chiffres : les chiffres arabes et les chiffres hindous .
- La lettre arabe change de forme selon sa position dans le mot. Elle ne s'écrit donc pas de la même manière au début, au milieu ou en fin de mot ;

Alif	ا			ا
Ba	ب	ب	ب	ب
Ta	ت	ت	ت	ت
Tha	ث	ث	ث	ث
Jim	ج	ج	ج	ج
Ha	ح	ح	ح	ح
Kha	خ	خ	خ	خ
Del	د			د
Dhel	ذ			ذ
Ra	ر			ر
Zay	ز			ز
Sin	س	س	س	س
Shin	ش	ش	ش	ش
Sad	ص	ص	ص	ص
Dhad	ض	ض	ض	ض
Taa	ط	ط	ط	ط
Dha	ظ	ظ	ظ	ظ
Ain	ع	ع	ع	ع
Ghin	غ	غ	غ	غ
Fa	ف	ف	ف	ف
Kaaf	ق	ق	ق	ق
Kaf	ك	ك	ك	ك
Lem	ل	ل	ل	ل
Mim	م	م	م	م
Noun	ن	ن	ن	ن
Haa	ه	ه	ه	ه
Waw	و			
Ya	ي	ي	ي	ي

Tableau 4.1 L'alphabet Arabe dans différentes formes

- Plus de la moitié des caractères arabes (16) incluent dans leur forme des points qui peuvent être au nombre de 1, 2 ou 3. Ces points peuvent se situer au dessus ou en dessous du corps du caractère, mais jamais en haut et en bas simultanément.
- Certains caractères arabes incluent une boucle qui peut avoir différentes formes

ه ه ه ف ف ف ق ق ع ع ص ص ط م م م و ة لا

Figure 4.1 - Exemple de formes de boucles

- Certains caractères ne peuvent être rattachés à leur gauche et de fait ne peuvent se trouver qu'en position isolé ou finale. En présence de ces caractères, le mot est composé d'une ou de plusieurs parties qu'il est convenu d'appeler généralement PAW (Peace of Arabic Word) ou encore Pseudo-mot [AlBa 95]. Un PAW correspond donc à une chaîne d'un ou de plusieurs caractères (tableau 4.2). L'écriture arabe est ainsi semi cursive plutôt que totalement cursive.

AWs/mot	Ws/mot	PAWs/mot	PAWs/mot	V/mot
الأزهار	الانسان	الحساب	حرف	جملة

Tableau 4.2- Exemple de mots composés de la droite vers la gauche de 1,2,3,4 et 5 PAWs respectivement..

- Les dimensions des caractères sont variables, même s'il s'agit des différentes formes d'un caractère voir figure ci-dessous ;
- Pour des raisons de justification de texte et/ou d'esthétique, les ligatures horizontales peuvent être allongées en insérant entre les caractères d'une même chaîne une ou plusieurs elongations « matta » (ou tatwil), correspondant au symbole

«-». L'élongation se situe toujours à gauche du caractère courant. Si le trait d'allongement est associé à un caractère en position de début ou finale, le caractère prend sa forme de milieu et voit sa chasse augmenter du nombre de « matta » insérées (Tableau 4.3) [Ben 99]. Au niveau du PAW, l'insertion de traits d'allongement affecte uniquement sa largeur, la morphologie reste la même comme indiqué dans la figure II-3 [Tre 97].

avec 6 élongations	avec 4 élongations	avec 2 élongations	Sans élongation
ت	ت	ت	ت
ج	ج	ج	ج
ق	ق	ق	ق
م	م	م	م

Tableau 4.3- Exemples de caractères avec et sans élongations

Le tableau (4.1) illustre les différentes lettres arabes ainsi que leurs formes au début, au milieu ou en fin de mot. Il est à noter la présence des points diacritiques rattachés aux lettres dont le nombre varie de un à trois points. Un nombre important de mots en arabes sont composés uniquement de consonnes. Néanmoins, nous notons la présence de voyelles au sein des mots. Les voyelles se subdivisent en deux catégories : voyelles longues et brèves.

Les voyelles longues sont composées de trois lettres alif (أ), waw (و) et yaa (ي). En revanche, les voyelles brèves sont facultatives. Nous distinguons une forme particulière d'une voyelle brève à savoir la double voyelle. La figure 1.9 illustre les types de voyelles de la langue arabe.

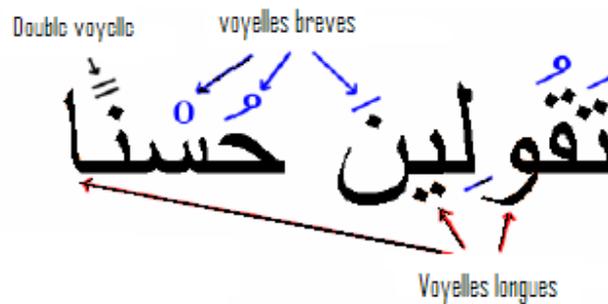


Figure 4.2 : Les voyelles de la langue arabe.

Il y a un cas particulier dans la composition des mots en langue arabe : le cas de la chadda. La présence d'une chadda au dessus d'une lettre indique que cette lettre est doublée. La figure 4.3 montre le dédoublement de lettre : chadda.

$$ا + ل + ر + ر + ح + م + ا + ن = الرَّحْمَان$$

Figure 4.3 : Le dédoublement de lettre : chadda.

4-2 Etat de l'art de la reconnaissance

4-2-1 METHODES STRUCTURELLES

Elles ressemblent beaucoup aux primitives topologiques. La différence est qu'elles sont généralement extraites non pas de l'image brute, mais à partir du squelette ou du contour de la forme. Ainsi, on ne parle plus de trous, mais de boucles ou de cycles dans une représentation filiforme du caractère.

Almuallim et Yamaguchi [Alm 94] proposent une technique de reconnaissance structurelle pour les mots arabe manuscrits Khorsheed et. Clocksin [Kho 92] présentent une technique d'extraction de caractéristiques structurelles de l'écriture arabe cursive. Dans cette étude le squelette de l'image binaire du mot est décomposé en un nombre de segments selon un certain ordre. Chaque segment est transformé en un vecteur de caractéristique. Les caractéristiques recherchées sont les courbatures des segments,

leurs longueurs relatives, et la position du segment par rapport au centre de gravité du squelette. Les résultats de cette préparation sont appliqués pour l'entraînement des Modèles Cachés de Markov (HMM) pour opérer la reconnaissance [Abd 02].

4-2-2 METHODES TOPOLOGIQUES

Le terme métrique désigne la mesure d'une distance. La topologie est "l'étude des propriétés de l'espace (et des ensembles) du seul point de vue qualitatif". Concrètement, la topologie consiste, à l'aide de sondes appliquées directement sur l'image "brute", à effectuer par exemple sur l'échantillon les mesures et les tests suivants :

- 1- compter dans une forme le nombre de trous,
- 2- évaluer les concavités,
- 3- mesurer des pentes et autres paramètres de courbures et évaluer des orientations principales,
- 4- mesurer la longueur et l'épaisseur des traits,
- 5- détecter les croisements et les jonctions des traits,
- 6- mesurer les surfaces, les périmètres,
- 7- déterminer le rectangle délimitant l'échantillon, ou le polygone convexe,
- 8- évaluer le rapport d'élongation (ou allongement) longueur/largeur, ...
- 9- rendre compte de la disposition relative de ces primitives.

Dans leur étude M. Maged et M. Fahmy présentent le détail d'extraction de caractéristiques géométriques d'un caractère arabe issu d'un mot introduit à partir d'une tablette graphique 3. [Mag 87] : est présentée une autre approche géométrique. Il s'agit d'appliquer la technique des histogrammes directionnels sur le contour de l'image du mot (caractère), et d'utiliser le code de Freeman pour coder cet histogramme.

4-2-3 LES PRIMITIVES STATISTIQUES

L'histogramme, qui représente le nombre de pixels sur chaque ligne ou colonne de l'image, en est un exemple classique et simple à calculer. On peut citer également l'approche basée sur un moyennage des pixels situés à l'intérieur d'un masque

rectangulaire : on construit une matrice de masques recouvrant la totalité de la forme qui permet une représentation statistique des valeurs correspondant à chaque masque.

W.F. Clocksin et M.S. Khorsheed [cloc 98] présentent une nouvelle méthode pour la reconnaissance des mots arabe manuscrits. Cette approche n'exige pas la segmentation du mot en ses caractères. Elle est appliquée à l'écriture arabe manuscrite cursive, où les ligatures, la superposition et la variation du style présentent un défi pour les méthodes qui utilisent la segmentation. Cette méthode ne demande pas à trouver les contours des mots. Elle ramène chaque mot « de l'écriture » dans un système de coordonnées polaire, et lui applique la transformée de Fourier. Le spectre obtenu est invariant en rotation, translation et à l'échelle. Chaque mot est représenté par un simple vecteur comprenant les amplitudes du spectre de Fourier. La reconnaissance est basée sur la distance Euclidienne de ce vecteur.

Dans [13], le système divise le texte arabe en ses mots constituants, à partir du blanc qui sépare ces mots. Un vecteur de caractéristiques est calculé, pour chaque mot. Ce vecteur est invariant par rapport à la translation, la rotation et l'échelle. Partons d'une image d'un Scanner en format bitmap du mot (qala= قال) par exemple voir figure (4.4).

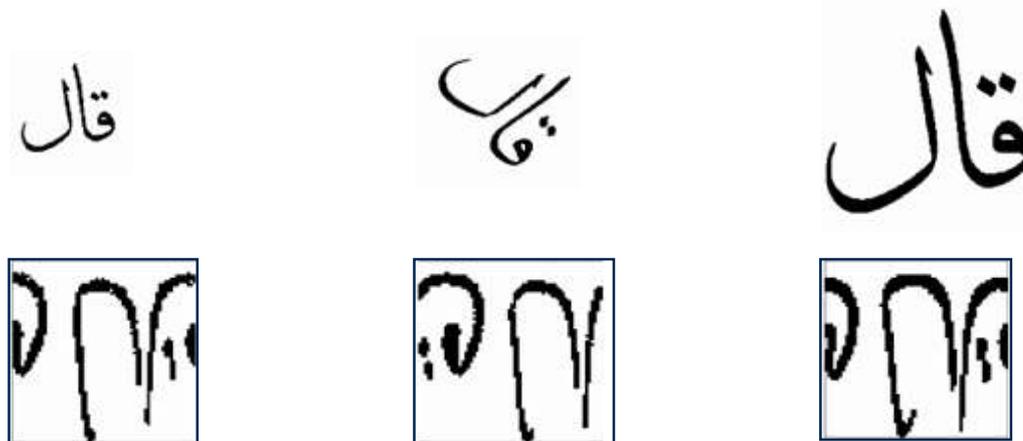


Figure 4.4: Trois images du mot qala et leurs correspondances en coordonnées polaires normalisées

Chaque pixel noir de l'image peut être représenté par un nombre complexe $Z = x+jy$, où $0 \leq x \leq X$, $0 \leq y \leq Y$. Soit $O(O_x, O_y)$ le centre de gravité des pixels noir du mot de l'image et D_{max} la distance Euclidienne maximale entre O et tous les pixels.

$$D_{max} = \max_{x,y} (\|Z - O\|) = \max_{x,y} \left(\sqrt{(x - O_x)^2 + (y - O_y)^2} \right) \quad 4.1$$

A chaque pixel de l'image, effectuons la transformation en coordonnées suivantes :

$$U(Z) = r + j\theta = \frac{\sqrt{(x-O_x)^2+(y-O_y)^2}}{D_{max}} + j \tan^{-1} \left(\frac{y-O_y}{x-O_x} \right) \quad 4.2$$

Où : $0 \leq r \leq 1, -\pi \leq \theta \leq \pi$

Cette transformation ramène l'objet dans l'espace des coordonnées polaires normalisé. Soit $f(x,y)$ l'image correspondante dans le nouveau espace avec une dimension de $N \times N$ pixels. L'idée principale derrière cette transformation est de s'affranchir du changement d'échelle et de ramener la rotation en une translation.

Comme nous l'avons déjà signalé, la transformée de Fourier est appliquée au corps de toute l'image. Ceci est dû à la complexité du contour obtenu si jamais on cherche à appliquer ces descripteurs au contour d'un mot de l'écriture. Si par contre l'image de l'écriture est constituée de simples caractères isolés, comprenant un seul corps chacun, les FDs sont favorablement appliqués au contour de ces caractères. Un détecteur de contour est alors appliqué au corps de l'image initiale, et une fonction 2D qui représente ce contour est alors obtenu. A cette fonction on applique maintenant les DFs. Pour s'échapper aux problèmes de rotation et de changement d'échelle, on peut toujours appliquer l'idée du paragraphe précédent.

4-2-3-1 Caractéristique spectral

Une fois l'objet ramené dans l'espace polaire, les descripteurs de Fourier (FDs) peuvent lui être appliqués. Ces descripteurs sont largement utilisés dans le domaine de la reconnaissance des formes et ils ont montrés un bon succès dans le domaine de la reconnaissance de l'écriture. Le contour de l'image est considéré comme une fonction à partir duquel un ensemble de caractéristiques spectrales invariantes peuvent être obtenues. Cette opération réussie très bien dans le cas de caractères isolés. Pour sa bonne réussite dans la reconnaissance des mots, une segmentation fiable est alors exigée. C'est une opération assez difficile à obtenir en pratique. En plus si les FDs sont souhaitable pour des caractères à simple contour, ils ne peuvent être directement appliqués aux contours des mots entiers du fait de la complexité de la courbe qui lie les différents caractères. C'est pour cette raison que les FDs sont appliqués au corps de toute l'image et non pas au contour de ses caractères.

$$F(\omega_x, \omega_y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) e^{-j\frac{2\pi}{N}(x\omega_x + y\omega_y)} \quad 4.3$$

$$0 \leq \omega_x, \omega_y \leq N$$

En plus, puisqu'on travail dans l'espace polaire, et que toute rotation de l'image est ramenée à une translation, au niveau des DFs ceci se traduit par une variation de phase comme le montre l'expression suivante :

$$f(x - x_0, y - y_0) \rightarrow F(\omega_x, \omega_y) e^{-j\frac{2\pi}{N}(x_0\omega_x + y_0\omega_y)} \quad 4.4$$

De cette expression, on note bien une indépendance des amplitudes, des différentes harmoniques, du décalage. Si de cette expression on garde uniquement ces amplitudes on assure alors une indépendance du spectre d'amplitude obtenu par rapport au décalage donc par rapport à la rotation d'origine.

CHAPITRE 5

CONTRIBUTION A LA RECONNAISSANCE DE L'ECRITURE ARABE MANUSCRITE

5-1 Reconnaissance des caractères arabes manuscrits isolés

5-1-1 Introduction

Ce chapitre concerne la mise en œuvre des différentes techniques décrites jusqu'ici. Il s'agit de la simulation de ces techniques et la comparaison des résultats obtenus. Nous mettons en zoom tout particulièrement notre nouvelle technique de reconnaissance et sa comparaison avec les autres.

5-1-2 Réseaux de neurones et classification

Avant de parler mise en œuvre proprement dite, parlons du moyen qui a permis à cette mise en œuvre : les réseaux de neurones. En quoi il s'agit ?

Les réseaux de neurones sont la réplique des neurones biologique c'est-à-dire les neurones qui existent en vous et moi. Un ensemble de cellules sont inter connectées d'une manière à répondre d'une manière bien spécifique à une excitation. La figure 5-1 ci-dessous donne l'exemple de l'organisation de ces cellules dans une structure réseau.

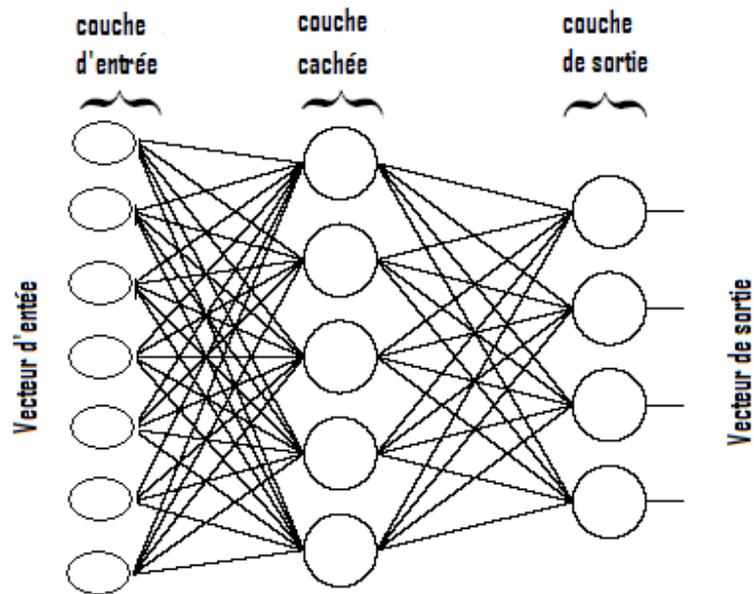


Figure 5.1 schéma du réseau de neurone

Comment ça travail ?

- ☛ Des couches de cellules sont disposées parallèlement une après l'autre comme le montre la figure (5.1) ci-dessus. Les cellules de la couche d'entrée vont recevoir le vecteur d'activation. Les sorties de ces cellules sont connectées aux entrées des cellules de la deuxième couches et ainsi de suite : les sorties de la couche « i » constituent les entrées de la couches « $i+1$ ». La première couche est appelée « couche d'entrée ». C'est elle qui recevra le vecteur d'activation. La dernière couche est appelée couche de sortie. Ces sorties donnent le vecteur réponse du réseau. Les couches intermédiaires sont appelé couches cachées et sont numérote de 1 jusqu'à « N » si le réseau dispose de « N » couches.
- ☛ Le neurone de la couche « i » qui rassemble les sorties des neurones de la couches « $i-1$ » donne en sa sortie une réponse qui dépend de la fonction de transfert dont il est équipé. Ces fonctions sont à choisir selon l'application. On trouve entre autres des fonctions de transfert de type sigmoïde, de type gaussienne etc....
- ☛ Les liaisons entres neurones sont matérialisées par des poids ;
- ☛ La détermination de ces poids est effectuée dans une première phase : la phase d'apprentissage. Une fois la réponse du système est conforme à la sortie prévue,

ces poids seront sauvegardés et utilisés dans la deuxième phase : la phase de reconnaissance.

- ☛ L'apprentissage proprement dit est opéré en effectuant une comparaison entre la sortie calculée du réseau et la sortie prévue. La différence entre les deux sorties, qui constitue le vecteur différence, constitue l'entrée d'un algorithme de minimisation qui réajuste les poids du réseau jusqu'à avoir un vecteur différence inférieur à un certain seuil prédéfini.
- ☛ Parmi les algorithmes de réajustement des poids, couramment utilisés, on peut citer entre autres l'algorithme de rétro propagation.

5-1-3 Application

Pour une première étude, nous nous sommes proposés de bâtir un réseau de reconnaissance des caractères arabes isolés. Les vingt huit caractères de l'écriture arabe ont été enregistrés pour un ensemble de scripteurs égale à vingt.

5-1-3-1 Extraction des primitives (caractéristiques)

Les trois méthodes, Moments invariants, Fourier et Zernike, détaillées dans le chapitre III, ont été utilisées pour le calcul des primitives de tous les caractères rassemblés.

5-1-4 Préparation du réseau

5-1-4-1 Structure du réseau

Le réseau utilisé est de type perceptron (MLP) à trois couches à rétro propagation de l'erreur.

- Le nombre de neurones de la couche de sortie est égale au nombre de situations à distinguer. Dans notre application il s'agit de faire la distinction entre les 28 caractères de l'écriture arabe, le nombre de neurones de cette couche est fixé donc à 28 neurones. Ce nombre peut être réduit en adoptant

un codage binaire de ces 28 sorties ce qui donne un total de 5 bits donc cinq neurones en sortie ;

- Le nombre de neurones de la couche cachée est fixé, après plusieurs essais, à quarante neurones. Ce nombre est celui qui donne la meilleure réponse du système : meilleure au sens de taux de reconnaissance.
- Le nombre de neurones de la couche d'entrée dépend de la méthode utilisée :
 - Pour la méthode des moments invariants, le nombre de ces moments est sept à qui s'ajoute un indicateur pour le nombre de points, zéro si le caractère n'en contient pas, un autre indicateur pour la position de ces points par rapport au centre de gravité (1 au dessus, 0 en dessous) et un dernier indicateur sur la présence ou l'absence des boucles dans la structure du caractère (1 présence de boucle, 0 absence de ces boucles). Le total des identificateurs est donc 10. C'est le nombre de neurones de la couche d'entrée ;
 - Pour la méthode de Fourier, le nombre de descripteurs de la transformation est fixé, après plusieurs essais, à 20. S'ajoute à ce nombre les trois paramètres correspondant aux points et au contour. Le nombre de neurones de la couche d'entrée est égal donc à 23.
 - Pour la méthode de Zernike, le nombre de neurones, également après plusieurs essais, est fixé à 42 pour un ordre de développement égale 9.

5-1-4-2 Phase d'apprentissage

Chaque scripteur a écrit les vingt huit caractères de la langue arabe. Les moments géométriques invariants sont calculés pour chacun de ces caractères. Deux possibilités nous sont déjà offerte quant à la manière d'ordonner les moments géométriques invariants des différents scripteurs pour entamer l'opération d'apprentissage :

- Une première méthode consiste à prendre tous les moments de tous les caractères du premier scripteur suivis de tous les moments de tous les caractères du deuxième scripteur, et ainsi de suite jusqu'au dernier scripteur : Si on désigne par ϕ_{ijk} le

moment d'ordre « i ; $1 \leq i \leq 7$ » correspondant au caractère « j ; $1 \leq j \leq 28$ » du scripteur « k ; $1 \leq k \leq 7$ », le fichier d'apprentissage aura l'allure suivante:

```

Φ111 Φ211 Φ311 Φ411 Φ511 Φ611 Φ711 1
Φ121 Φ221 Φ321 Φ421 Φ521 Φ621 Φ721 2
Φ131 Φ231 Φ331 Φ431 Φ531 Φ631 Φ731 3
.....
Φ1,28,1 Φ2,28,1 Φ3,28,1 ..... Φ7,28,1 28
Φ112 Φ212 Φ312 Φ412 Φ512 Φ612 Φ712 1
Φ122 Φ222 Φ322 Φ422 Φ522 Φ622 Φ722 2
Φ132 Φ232 Φ332 Φ432 Φ532 Φ632 Φ732 3
.....
Φ1,28,2 Φ2,28,2 Φ3,28,2 ..... Φ7,28,2 28
.....

Φ1,1,10 Φ2,1,10 Φ3,1,10 ..... Φ7,1,10 1
Φ1,2,10 Φ2,2,10 Φ3,2,10 ..... Φ7,2,10 2
Φ1,3,10 Φ2,3,10 Φ3,3,10 ..... Φ7,3,10 3
.....
Φ1,28,10 Φ2,28,10 Φ3,28,10 ..... Φ7,28,10 28

```

- Une deuxième méthode consiste à prendre tous les moments du premier caractère pour tous les scripteurs, suivi de tous les moments du deuxième caractère pour tous les scripteurs et ainsi de suite jusqu'au dernier caractère. Si on garde la même notation ci-dessus, le fichier d'apprentissage aura l'allure suivante :

```

Φ111 Φ211 Φ311 Φ411 Φ511 Φ611 Φ711 1
Φ112 Φ212 Φ312 Φ412 Φ512 Φ612 Φ712 1
.....
Φ1,1,10 Φ2,1,10 Φ3,1,10 ..... Φ7,1,10 1
Φ121 Φ221 Φ321 Φ421 Φ521 Φ621 Φ721 2
Φ122 Φ222 Φ322 Φ422 Φ522 Φ622 Φ722 2
.....
Φ1,2,10 Φ2,2,10 Φ3,2,10 ..... Φ7,2,10 2

```

Φ131	Φ231	Φ331	Φ431	Φ531	Φ631	Φ731	3
Φ132	Φ232	Φ332	Φ432	Φ532	Φ632	Φ732	3
.....							
Φ1,3,10	Φ2,3,10	Φ3,3,10	Φ7,3,10			3
Φ1,28,1	Φ2,28,1	Φ3,28,1	Φ7,28,1			28
Φ1,28,2	Φ2,28,2	Φ3,28,2	Φ7,28,2			28
.....							
Φ1,28,10	Φ2,28,10	Φ3,28,10	Φ7,28,10			28

Opérer l'apprentissage avec l'un ou l'autre de ces deux fichiers, influe d'une manière considérable sur la rapidité (le temps) de l'opération et même sur les valeurs de poids calculés, ce qui influencera d'une manière directe sur la qualité et les résultats final pendant la phase de reconnaissance. Les essais ont montrés que c'est la première organisation qui fini par donner des banques au bon pourcentage de reconnaissance. C'est donc cette configuration qui sera retenu dans le restant de ce travail.

Ces moments ont constitués donc les entrées du réseau, et l'apprentissage est exécuté en mode supervisé.

5-1-4-3 Activation des sorties

A un instant donné, c'est une et uniquement une seule sortie du réseau qui est active. Elle correspond à l'ordre du caractère dont les moments viennent juste d'attaquer l'entrée de ce réseau. Pour aider le réseau à connaître, parmi toutes ses sorties, celle qui sera active à un moment donné, le fichier d'apprentissage est configuré de sorte qu'il porte cette information : C'est le contenu de la dernière colonne de ce fichier (voir ci dessus). Un (1) désigne la première sortie, deux (2) désigne la deuxième sortie etc.

5-1-4-4 Fichier de reconnaissance

Nous l'avons déjà signalé, le fichier de reconnaissance est la combinaison de plusieurs opération d'apprentissage correspondante chacune à une catégorie de scripteurs et donnant naissance à une banque spécialisée. A la fin de chaque opération d'apprentissage, le programme se charge d'enrichir (d'ajouter) ce fichier avec la

nouvelle banque qui vient d'être déterminée et mettre à jour le paramètre « N ». On peut schématiser la structure de ce fichier comme suit :

Nombre de population : N
Nombre de couches =
Nombre de neurones de chaque couche (7,40,28);
Coefficients d'apprentissage (Alpha) =

Momentum (Beta) =

Banque1

Banque2

.....

BanqueN

La Banque1 correspond à la première catégorie de scripteurs (population1), la *Banque2* correspond à la deuxième catégorie de scripteurs (population2)...etc.

Une Banque « i » contient les poids de liaisons entre tous les neurones des différentes couches.

5-1-4-5 Phase de reconnaissance

Il s'agit dans cette phase de savoir quelle est la sortie correspondante à une entrée attaquée par un caractère « X ». Le déroulement de l'opération est schématisé ci-dessous.

Dans ce schéma les moments qui attaquent le réseau sont représentés par le caractère. Ces moments attaquent à tour de rôle chacune des banques. Un Buffer de sortie permet d'additionner, dans l'ordre, les réponses en sorties de toutes ces banques.

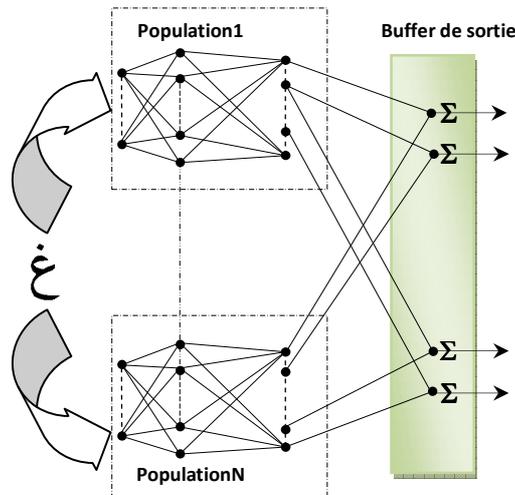


Figure 5.2 : Réorganisation de la structure du réseau en populations

Si « k » sorties d'ordre « i », de ces banques, sont activées (=1) en présence d'un caractère, la sortie d'ordre « i » du système entier est égale à : $\sum_k 1 = k$.

5-1-4-6 Résultats et discussions

Une banque spécialisée, est normalement le résultat d'apprentissage d'un ensemble d'individus de même catégorie qu'on a appelé population. Pour simuler cette population, on peut demander à la même personne d'écrire l'ensemble des caractères plusieurs fois.

Dans sa configuration la plus simple, une population peut contenir un seul individu. C'est ce qu'on a adopté pour mettre en œuvre cette nouvelle technique.

Pour mener une étude comparative, deux méthodes d'apprentissage ont été appliquées : La méthode classique et la nouvelle méthode qu'on propose. Un ensemble de scripteurs a servi à déterminer une banque de reconnaissance générale unique selon le procédé classique. C'est-à-dire que les moments de ces scripteurs sont disposés les uns à la suite des autres, et ont servi à l'apprentissage du réseau.

1. Ils ont été utilisés pour créer des banques individuelles spécialisées pendant la phase d'apprentissage. Toutes ces banques ont été rassemblées dans une banque unique qui va être utilisée pendant la phase de reconnaissance.

Ces deux opérations étant effectuées, un ensemble de scripteurs étrangers, c'est à dire qui n'a pas servis à l'apprentissage du réseau, est utilisé dans la phase de reconnaissance.

5-1-5 Méthode de découpage de l'image

Nous avons constatés que le nombre sept de moments invariants, utilisés comme descripteurs d'un caractère, peut se révéler insuffisant. On a cherché donc de nouvelles caractéristiques qui peuvent se joindre à ces moments pour augmenter le nombre de descripteurs et donc augmenter les chances de reconnaissance. Une nouvelle technique est alors mise à jour. Il s'agit de la technique de découpage (chapitre 3) : Une image caractère est découpée, par rapport à son centre de gravité, en un ensemble de sous images. Trois directions sont proposés pour réaliser ce découpage : Horizontalement, Verticalement et Horizontalement-Verticalement. Chacune des ces sous images donne naissance à sept moments invariants. Le nombre des moments invariants résultants est donc 7×2^n , où n est le nombre de sous images obtenues pendant la phase de découpage.

L'opération de découpage peut être à plusieurs niveaux : Une sous image obtenu à un niveau « i » peut faire l'objet, elle même, d'un découpage au niveau « H-1 ». C'est une sorte de zoom en profondeur qui permet d'aller de plus en plus vers les détails de l'image. A la limite le détail le plus fin est le pixel. Normalement on arrivera jamais à cette limite parce que l'intérêt du découpage est justement d'éviter de comparer les images pixels par pixel est de gagner donc dans les temps des traitements.

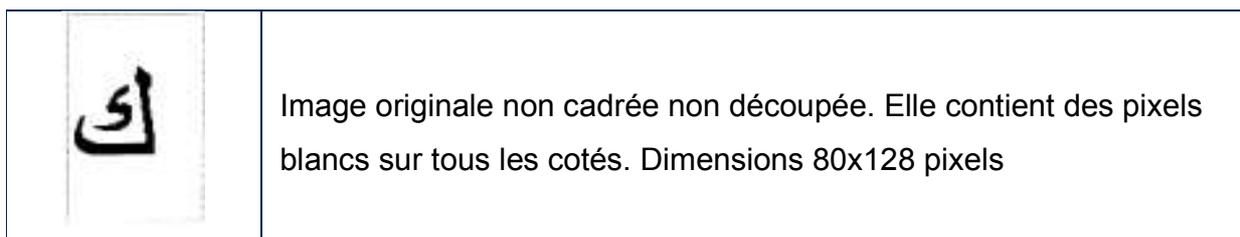
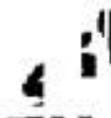


	Image cadrée non zoomée. Dimensions 49x55 pixels
	Image cadrée et zoomée : Dimensions 64x64 pixels. Tous les pixels blancs qui sont en plus sur les cotés sont enlevés. Opérer un zoom, en avant ou en arrière sur cette image, prend en compte donc uniquement les pixels qui font partie du rectangle de contenance de l'image du caractère.
	Les deux sous images gauche et droite de l'image cadrée zoomée. Elles constituent chacune une image à part entière et peut donc subir tous les traitements de l'image mère.
	Les deux sous images haut et bas de l'image cadrée zoomée. Elles constituent chacune une image à part entière et peut donc subir tous les traitements de l'image mère.
	Les quatre sous images obtenues par un découpage, de l'image cadrée zoomée, dans les deux directions. Chacune de ces sous images va subir un traitement d'identification et peut constituer, elle même, une image à part entière pour affiner l'identification si besoin est.
	Découpage de deux niveaux : Dans le premier niveau un découpage, dans les deux directions, est opéré sur l'image entière ; Dans le deuxième niveau un découpage vertical est opéré sur la sous image droite haut.

Le découpage constitue la deuxième phase, si l'identification de l'image entière a échoué : On découpe l'image selon une direction donnée et on détermine les moments invariants de chacune des sous images. Ces moments vont nous servir à l'identification de ces sous images exactement comme nous ont servis les moments invariants de l'image entière pour son identification. Le choix du type de découpage (horizontal, vertical, ou dans les deux directions) de cette deuxième phase n'est pas

encore fixé. Les essais qui vont être menés vont nous permettre de fixé ce choix. Si même à ce niveau l'identification n'est pas encore accomplie, on peut passer à un deuxième niveau de découpage et ainsi de suite. Le tableau ci dessus nous donne l'exemple d'une image découpé en deux niveaux. Dans le premier un découpage, dans les deux directions, est opéré sur l'image entière. Dans le deuxième niveau un découpage vertical est opéré sur la sous image droite haut. Ces sous sous image vont subir un traitement d'identification à leur tours. On peut continuer ainsi jusqu'à obtenir une identification sans ambiguïté de l'image complète du caractère non découpé.

Un programme de découpage est développé en conséquence. Il permet le choix du type de découpage et d'opérer ce découpage. Les sous images, obtenues de ce découpage, sont automatiquement sauvegardés.

Tous ces programmes sont rassemblés et mise à la disposition de l'utilisateur à travers une seule interface graphique interactive développée à l'aide de Borland C++ Builder. Cette interface permet à l'utilisateur, à l'aide d'un menu constitué de plusieurs fenêtres spécialisées:

- ,/ De charger une image quelconque. Il s'agit dans notre cas de l'image d'un caractère;
- ✓ D'opérer un Zoom de l'image chargée ;
- ,(De calculer les moments Absolus, Centrés, Centrés Normalisés ou Invariants du caractère chargé; ,(D'opérer un découpage de l'image chargée ;
- ✓ De procéder à l'apprentissage ou à la reconnaissance.

5-2 Reconnaissance des caractères arabes manuscrits à partir d'un texte

5-2-1 Introduction

Après exposition des différentes techniques de traitement de l'écriture arabe, dans toute la chaîne, depuis l'acquisition jusqu'à la reconnaissance, ce chapitre sera consacré aux détails de notre contribution quant à la reconnaissance de l'écriture arabe manuscrite sur deux plans :

Sur le premier plan, nous nous somme fixé comme objectif le développement d'un système de reconnaissance des caractères arabes manuscrits et isolés. La

bibliographie qui traite le sujet est abondante [Rou 04] [Elb 00] et il est même sans intérêt de reprendre ce qui est déjà fait. Mais nous avons vu la chose autrement : Il va sans dire que l'écriture, arabe, manuscrite dépend du scripteur : Son état, son âge, son sexe etc.... Si par un quelconque moyen on fait surgir ses caractéristiques, on peut les utiliser pour la reconnaissance des caractères manuscrits. C'est ce que nous avons développés dans ce travail. L'état, l'Age, le Sexe etc sont des paramètres fictifs. Pour les faire sortir, c'est leurs images qui est utilisées :

- ☛ Des fichiers spécialisés sont créés ;
- ☛ Chaque fichier rassemble les vecteurs des caractéristiques correspondant aux différents caractères de différents scripteurs appartenant à une catégorie donnée (grand, petit, male etc..). C'est Ce fichier joue le rôle de l'image de cette catégorie ;
- ☛ Après apprentissage, du réseau et sur la base de ces fichiers, il y a naissance de banques spécialisées qui contiennent les poids des différents neurones du réseau ;
- ☛ L'ensemble de ces banques donne une banque générale, qui servira dans l'étape de reconnaissance.
- ☛ La configuration du système est telle qu'elle permet d'enrichir cette banque avec d'autres catégories de scripteurs au fur et à mesure. C'est une banque ouverte et extensible.

Le détail de création et la structure même de ces banques sera donnée dans l'Annexe A et B.

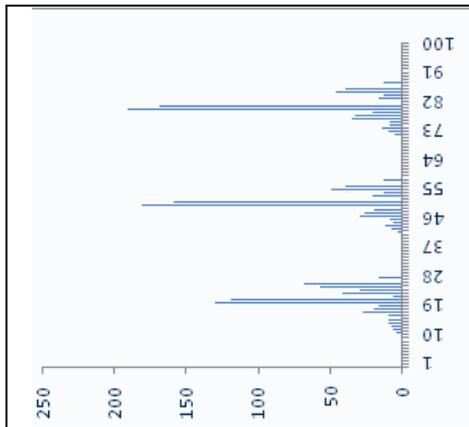
5-2-2 Reconnaissance des caractères arabes manuscrits

Sur un deuxième plan, nous nous sommes proposés de récupérer des caractères à partir d'un texte arabe manuscrit et de les reconnaître. Il est question donc de partir du tout début de la chaîne, c'est-à-dire à partir de l'image d'un texte arabe manuscrit. Nous verrons comment sont obtenus les lignes du texte, comment sont obtenus les mots ou pseudo mots à partir de ces lignes, comment sont obtenus les caractères à partir de ces mots (pseudo mots) et pour conclure comment sont reconnus ces lettres.

5-2-3 Segmentation du texte en lignes

5-2-3-1 Récupération des lignes

L'idée est simple : Les creux (vallées) de l'histogramme horizontal correspondent aux plages d'absence d'écriture. C'est à cet endroit qu'est opéré le découpage. Cela suppose bien sûr que les hampes et les jambages des lignes ne se chevauchent par rapport à un observateur horizontal. La figure 5-3 ci-dessous représente un petit texte contenant trois lignes et l'histogramme horizontal correspondant (voir l'article [ZAH01], ROM95).

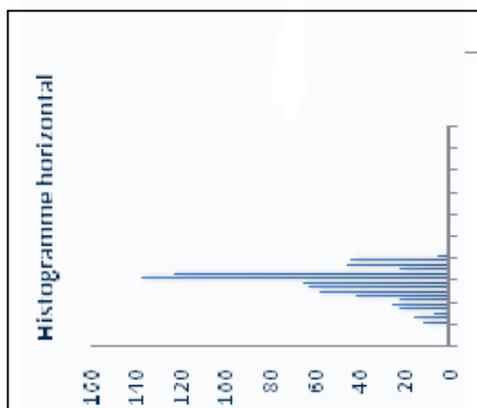


وقل ربي زدني علما
و علمني ما ينفعني
وانفعني بما علمتني

Figure 5-3 Texte avec son histogramme horizontal (pour la séparation des lignes)

5-2-3-2 Détection de la ligne de base

La ligne de base est la ligne sur laquelle repose l'écriture. Elle contient les **corps principaux** de tous les mots ou pseudo mots. Son utilisation sera détaillée dans les paragraphes qui suivent. Pour la détection de cette ligne on s'est servi de l'histogramme horizontal qui présente un pic net au passage par cette ligne (figure 5-4).



الجمهورية الجزائرية الديمقراطية الشعبية

Figure 5-4 : histogramme horizontal correspondant à cette ligne

5-2-4 Segmentation des lignes en mots

La encore c'est l'histogramme qui nous est venu en aide. Mais il s'agit de l'histogramme vertical. Par nature de l'écriture arabe même, comme d'ailleurs pour le latin, un espace blanc important sépare deux mots successives [GILL99]. C'est ce vide qui est exploité pour déterminer les points de découpages correspondant aux vallées de cet histogramme voir figure 5-5.

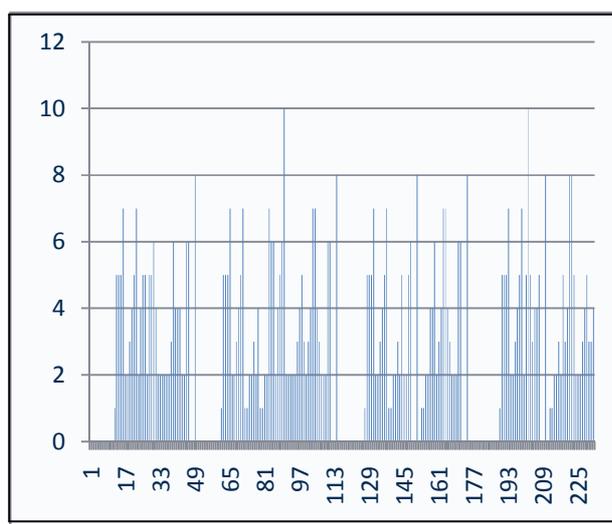


Figure 5-5 : ligne avec son histogramme vertical (pour la séparation des mots)

5-2-5 Décomposition du mot en ses composantes

L'écriture arabe est semi cursive ce qui signifie qu'un mot peut être composé de plusieurs, au maximum cinq, sous mots appelé *pseudo mots* (PAW, composantes connexes). Le résultat de décomposition de chaque ligne en mots peut donner naissance donc à un mot composé, de plusieurs pseudo mots, plutôt qu'un mot unique.

Rappelons que la finalité des choses est la récupération des caractères. On peut tenter d'opérer un découpage à l'aide de l'histogramme vertical maintenant. C'est une opération ramenée à l'échec à coup sûr. Pourquoi ? Considérons l'image du mot " حرية " (la liberté) figure ci-dessous. Il est clair que les caractères (ر) et (ي) sont superposés du coup il est impossible de les séparer à partir du résultat d'une simple projection verticale (histogramme vertical). Il faut disposer d'un moyen beaucoup plus intelligent pour réussir la séparation de ces caractères. Il s'agit de l'algorithme *glissement* détaillé ci-dessous.

L'idée principale de cet algorithme est la suivante: les mots ou les pseudo-mots constituent un corps soudé. Tous les points de ce corps sont rattachés les uns aux autres. A partir d'un seul point (pixel) appartenant à ce corps, on peut par progression successive récupérer l'ensemble du restant du corps. Ce dernier signifie à premier bord l'ensemble de tous les points. Il peut signifier aussi l'image de ce corps représentée en sa forme (contour). C'est cette dernière signification que nous avons adoptée pour la récupération des pseudo-mots (s'il y a lieu). Ce choix est dicté par un souci de simplicité de programmation et de rapidité du traitement. En fait le suivi du contour est une opération plus simple que le suivi de tout le corps.

Considérons à nouveau l'image du mot " حرية " de la figure 5-6 (a) ci-dessous. Le contour lui correspondant est donné en figure 5-6 (b).

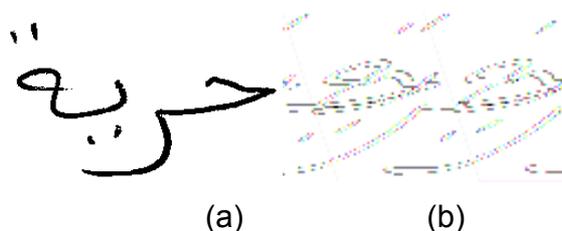


Figure 5-6 (a) : le mot " حرية " (b) : contour correspondant

On entend par corps principal l'ensemble des pixels ayant comme support la ligne de base. Ayant en possession les coordonnées de la ligne de base et le contour du mot, on va procéder maintenant à la récupération des corps principaux (au pluriel). En fait il peut s'agir d'un mot composé de plusieurs corps principaux correspondant chacun à un pseudo mot. La figure 5-7 (a) représente un mot constitué d'un seul corps (un seul pseudo mot). La figure 5-7 (b) représente un mot composé de plusieurs corps (plusieurs pseudo-mots).

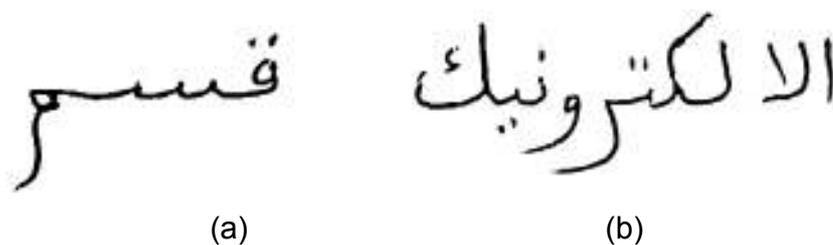


figure 5-7 (a) : mot constitué d'un seul corps

figure 5-7 (b) : mot constitué de plusieurs corps

Algorithme (glissement)

1. Détection de la ligne de base depuis le corps (plein) du mot ;
 2. Détection du contour. Utilisez le dans la suite des opérations;
faire
 3. Suivez la ligne de base horizontalement, de droite à gauche, jusqu'à détecter un pixel ;
 4. Sauvegardez ce pixel dans une nouvelle image et éliminé le de l'image d'origine ;
 5. Chercher le pixel suivant. Sauvegarder le dans la nouvelle image et éliminé le de l'image d'origine ;
 6. Reprendre depuis l'étape 5 jusqu'à ne plus avoir de pixel
 7. Procédez à un découpage du pseudo mot, qui vient d'être récupéré, en composantes principales (caractères);
 8. Reprendre depuis l'étape 3 jusqu'à ne plus avoir de pixels
- fin

Algorithme de récupération des pseudos mots et caractères

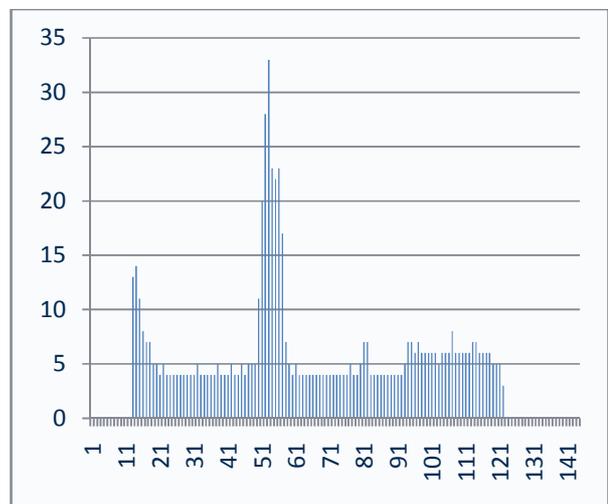
Cet algorithme récupère donc les pseudo-mots un à un. Pour le mot (حرية), c'est le pseudo-mot (ح ر) qui est récupéré. Tout ce passe comme si, pendant cette opération

de récupération, on a fait glisser la lettre (ج) par-dessous la lettre (ـ) d'où le nom algorithme glissement.

- Le découpage des pseudo-mots, récupérés en chaque fin de l'étape 6 de l'algorithme précédent, est opéré après remplissage de ces pseudo-mots dans la nouvelle image. Ce remplissage est dicté par le souci de se donner les meilleures chances de détecter les creux correspondant aux points de découpage. En effet si le découpage est opéré sur le contour du pseudo mot certains caractères se trouvent confondus avec les elongations naturelles d'autres caractères (voir figure ci-dessous).



(a)



(b)

Figure 5-8 (a) : a le mot (جبل)

Figure 5-8 (b) le corps principal du mot (جبل) et l'histogramme vertical correspondant

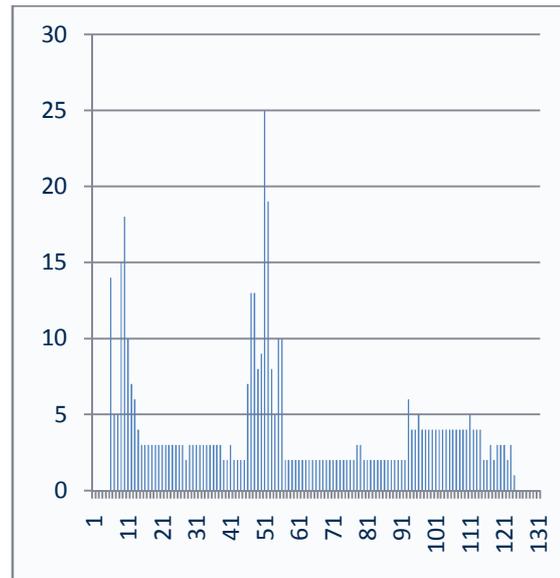


Figure 5-9 Le contour du corps principal du mot (جبل) et l'histogramme vertical correspondant

Il n'est pas garanti du tout qu'au niveau du dent correspondant à la lettre (ـبـ) on trouve un pic front au niveau de l'histogramme. Il est même probable de trouver dans toute la région définie par la fin de la lettre (ـجـ) et le début de la lettre (لـ) un histogramme aplati qui absorbe la lettre (ـبـ) et qui autorise en plus le découpage à ce niveau.

L'algorithme de remplissage est très simple.

Algorithme

1. Chercher la première ligne ;
 - faire
 2. Balayer cette ligne de l'image jusqu'à détection d'un pixel noir;
 3. Mettre au noir, tous les pixels blancs, sur cette ligne, jusqu'à la rencontre d'un pixel noir ;
 4. Passer à la ligne suivante ;
- Tant qu'il ya des lignes ;

Algorithme de remplissage des pseudos mots

- A ce stade, si l'objectif de l'opération est une reconnaissance des mots, on peut dore et déjà opérer une reconnaissance des pseudos mots récupérés pour remonter petit à petit, par association de ces pseudo mots, vers le mot complet.

Pratiquement, l'opération de découpage proprement dite est opérée après le remplissage des pseudos mots. Il s'agit de décomposer le pseudo mot en ses composantes principales qui sont les caractères.

A ce stade, on a fait appel à l'algorithme de découpage décrit dans [ZAH 98]. Dans cet algorithme, on procède d'abord par élimination de la ligne de base déjà trouvée. Ce qui donne un pseudo mot avec une ligne horizontale blanche (dépourvue de pixels) au milieu. Ce découpage donne également des espaces blancs verticalement. Au niveau de l'histogramme vertical, ces espaces blancs se traduisent par un manque de pixels (des creux). C'est au niveau de ces creux qu'est opéré le découpage (voir figure 5-10). Les segments ainsi obtenus peuvent correspondre à des caractères complets, ou à des fragments de caractères. Chaque segment est récupéré à part entière. C'est à la charge du module de reconnaissance qu'est décidée la validité ou non de ces segments. Ce module travail en parallèle avec le module de découpage : A la fin de chaque opération de découpage, le module de reconnaissance est activé en lui passant comme argument les fragments un à un.

Les erreurs sont détectés lors de la segmentation, ils sont généralement de type sur-segmentation, c'est à dire qu'un caractère est découpés en deux segments ou plus. Le tableau ci dessous résume le type d'erreurs pouvant survenir lors de la segmentation.

caractères	م	ص	س	د	ب
Caractères segmentés	م	ص	س	د	ب

Pour corriger ces erreurs, les points diacritiques sont réassociés aux caractères, chacun dans son emplacement.

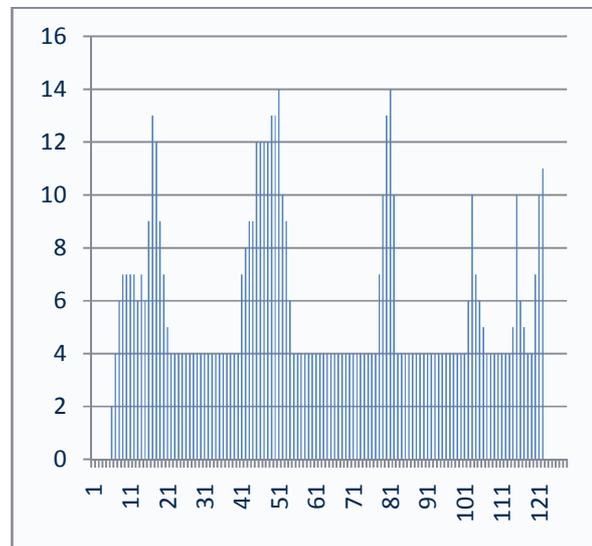
- pour les caractères de la classe (ب ت ث) , et د (ذ). En effet ils se situent toujours en fin du pseudo-mot. Ce type d'erreurs est corrigé en reliant toujours le dernier segment au segment précédent.
- Pour les caractères de la classe (ب ت ث ي ن) , les segments générés ont une forte ressemblance au début ou en milieu du pseudo-mot. La différence entre ces différents cas est les points diacritiques qui apparaissent dans ces caractères mais pas dans les segments générés. Pour corriger ce type d'erreurs on teste sur la ponctuation au dessus ou en dessous du segment. Si le segment ne contient pas de ponctuation il se peut qu'il s'agisse du caractère س, il est relié au segment suivant.

5-2-6 Processus de segmentation

Analysons d'abord les histogrammes verticaux des deux figures ci-dessous.

سبعة

(a)

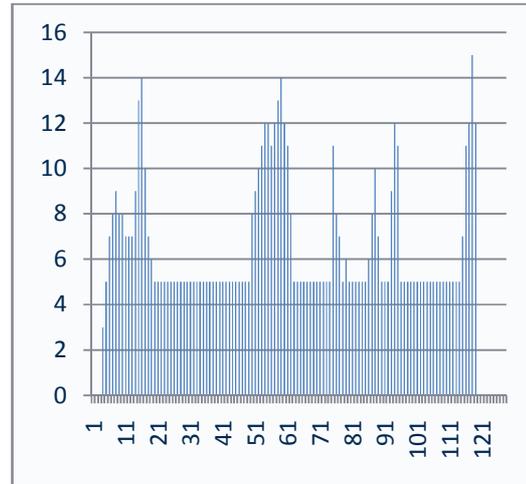


(b)

Figure 5-10 : le mot 'سبعة' et l'histogramme de son corps principal

تسعة

(a)



(b)

Figure 5-10 : le mot 'تسعة' et l'histogramme de son corps principal

تسعة تسعة

Figure 5-10 (c) : le mot 'تسعة' avec la suppression de la ligne base

Considérons comme points de découpage les minimas de ces histogrammes. Il est clair que le découpage, ainsi opéré, donne lieu à deux images dont les corps se ressemblent et peuvent être interprétés comme identiques. Pourtant il s'agit de deux caractères totalement différents. La première image est celle d'*une partie* (graphème) du caractère entier (س) alors que la deuxième est celle du *caractère entier* (ت). Il est nécessaire alors de disposer d'un outil de séparation ou de levée d'ambiguïté dans de telles situations. Cet outil est un module spécialisé qui fait appel aux corps secondaires des mots, les points. En effet les deux images extraites ont été privées de leurs corps secondaires pour des raisons de simplicité de segmentation, mais que la situation, à ce stade, est arrivée à confusion ce sont ces corps secondaires qui vont jouer le rôle de séparateurs. Ce module opère selon l'algorithme suivant :

Algorithme

```

1. xmin ← 0, ymin ← ordonnée de base
2. xmax ← abscisse découpage, ymax ← hauteur de l'image
3. fenêtre de récupération ← (xmin, ymin, xmax,ymax);
4. récupérer le contenu de la fenêtre depuis l'image d'origine ;
5. si (contenu ≠0)
    si (nombre de corps =3)
        c'est un ' نْ '
        passer au découpage suivant
    fin si
    si (nombre de corps =2)
        c'est un ' ن '
        passer au découpage suivant
    fin si
    si (nombre de corps =1)
        c'est un ' ن '
        passer au découpage suivant
    fin si
fin si
sinon
ymin ← 0, ymax ← ordonnée de base
fenêtre de récupération ← (xmin, ymin, xmax,ymax);
récupérer le contenu de la fenêtre depuis l'image d'origine ;
si (contenu ≠0)
    si (nombre de corps =2)
        c'est un ' نـ '
        passer au découpage suivant
    fin si
    si (nombre de corps =1)
        c'est un ' نـ '
        passer au découpage suivant
    fin si
fin si
sinon
    c'est un graphème du ' سد '
    élargir la fenêtre jusqu'au 3 ème point de segmentation
    récupérer le caractère (c'est un ' سد ' ou ' نسد ')
    passer au découpage suivant
fin sinon
fin sinon

```

Algorithme de levée d'ambiguïté des graphèmes de la famille ' sin= سد '

Ce module spécialisé revient donc à l'image originale qui n'est maintenant qu'un ensemble de corps secondaires. Il les récupère au fur et à mesure pour sortir de la

confusion à laquelle est confronté le système. Cette récupération utilise une fenêtre de largeur égale l part d'abord au dessus de la ligne de base pour chercher d'éventuels corps. Dans l'affirmative, il détermine le nombre de ces corps ce qui conduit à l'identification sans ambiguïté du caractère. C'est l'un du premier ensemble de la famille ' sin= سد '. Dans le cas contraire, il poursuit son inspection mais cette fois en dessous de la ligne de base. La encore c'est le nombre de corps secondaire qui, s'ils existent, conduit à l'identification du caractère. C'est l'un du deuxième ensemble de la même famille. Sinon il s'agit bien d'un graphème de caractère ' sin= سد ', le système opère alors une récupération des deux graphèmes restant pour compléter le caractère ' sin= سد ' en entier.

- En entend par famille ' sin= سد ', l'ensemble des caractères qui ont le même corps principal que celui du premier graphème de cette lettre. Ils sont de deux ensembles :
 - Le premier ensemble dispose de corps secondaire au dessus de la ligne de base. Il s'agit de : **ت, ث, ن**;
 - Le deuxième ensemble dispose de corps secondaire au dessous de la ligne de base. Il s'agit de : **ب, ي**.

Pour valider notre démarche, on a pratiqué l'opération de découpage-assemblage sur le mot (حرية), selon la chaîne de traitement suivante :

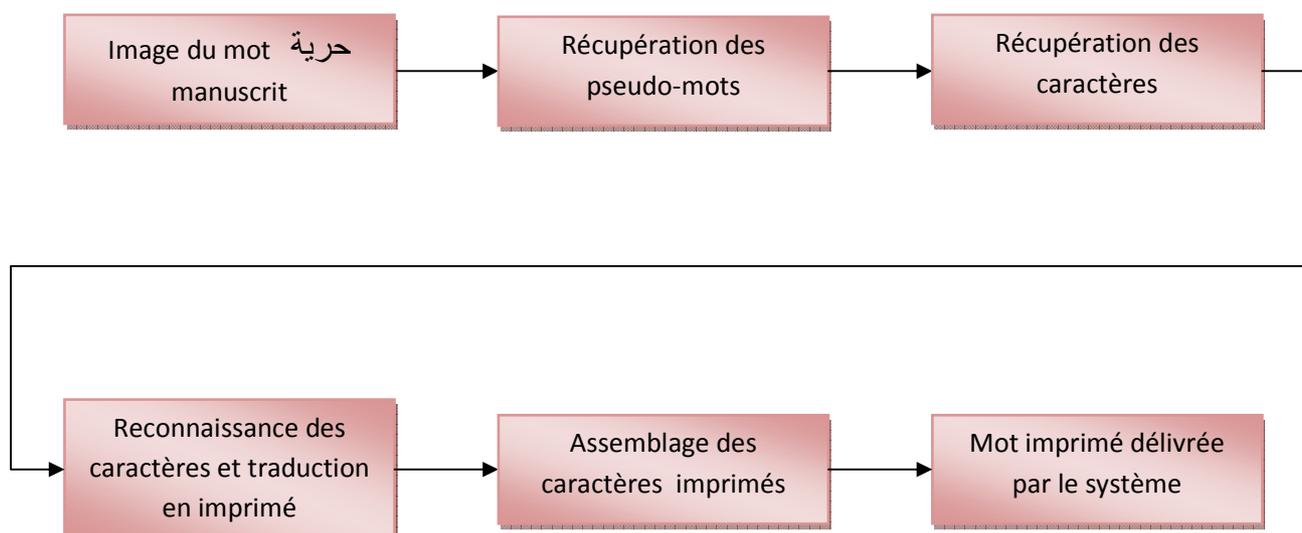


Figure. 5-11 : Chaîne de découpage-Assemblage du mot **حرية**

CHAPITRE 6

SIMULATIONS ET RESULTATS

6-1 Introduction

Les techniques de reconnaissances présentées dans le chapitre précédent vont être exploitées dans ce chapitre pour la reconnaissance des caractères arabes manuscrits. Au préalable, il est important de signaler que ces techniques peuvent être appliquées aux images des caractères entiers, ou à des images représentant des parties de l'image entière du caractère. C'est une nouvelle technique qu'on a mis en œuvre pour la première fois. Il s'agit, au lieu de prendre le caractère entier et lui appliquer un algorithme de calcul quelconque (par exemple calcul des moments), de prendre une partie de ce caractère et lui effectuer ce traitement. C'est une sorte de zoom sur le caractère pour essayer de voir plus finement les détails que contient celui-ci.

Cette idée nous est dictée par un besoin de plus de caractéristiques pour une bonne séparation entre les différents caractères. Ce découpage est opéré dans plusieurs directions, selon le choix de l'utilisateur et est activé si le processus de reconnaissance classique, en servant du caractère entier a échoué. Dans ce cas on ira voir un peu plus profondément dans le caractère en analysant ses différentes régions chacune à part. Cela permet d'augmenter le nombre de caractéristiques du caractère entier puisque chacune de ses régions vas contribuer avec ses caractéristiques propres à elle.

Si on se sert des moments invariants comme vecteur des caractéristiques, par exemple, on aura le tableau des caractéristiques suivant :

Découpage	Nombre de moments invariants
Caractère entier	7
Horizontal bas/ Horizontal haut	14
Vertical droite/Vertical gauche	14
Droite haut/Gauche haut/Droite bas/Gauche bas	28

Tableau 6.1 : Dimension du vecteur caractéristique par la méthode de découpage

Pour le caractère considéré en entier nous disposons donc d'un vecteur d'une longueur égale à 7. Pour le caractère découpé nous disposons d'un vecteur de longueur multiple de celle du caractère père. Il est bien évident, qu'on procédant de la sorte, nous augmentons les chances d'une séparation juste entre les caractères.

6-2 Architecture du réseau de neurone utilisé

Le réseau utilisé est de type perceptron multicouche (MLP) à rétro propagation de l'erreur avec un apprentissage supervisée. On a utilisé un réseau à trois couches (une couche d'entrée, une couche cachée et une couche de sortie).

6-2-1 La couche d'entrée

Le nombre de neurone de cette couche dépend du nombre de paramètres que peut générer chaque méthode d'extraction des caractéristiques. Dans le cas de la méthode des moments invariants par exemple ce nombre est de dix alors que pour la méthode des moments de Zernike ce nombre est fixé, après plusieurs essai, à 30.

6-2-2 La couche cachée

Le nombre de neurones de cette couche est choisit après plusieurs essais. Le meilleur nombre étant celui qui donne le meilleur résultat dans le sens du pourcentage de reconnaissance : avec les moments géométriques on a utilisé 40 neurones, avec les moments de Zernike on a utilisé 30 neurones.

6-2-3 La couche de sortie

Le nombre de neurones de sortie est dicté par le nombre de situation à séparer. Dans notre cas, il s'agit de séparer entre 28 caractères.

- La première solution qui a été adoptée est donc une couche de sortie avec 28 caractères. Chaque sortie correspond donc à un caractère et ce caractère est déclaré reconnu si cette sortie est active. Bien sûr dans le cas de confusion entre caractères, se sont plusieurs sorties qui s'activeront à la fois.
- L'autre possibilité, intéressante quant gain au niveau de l'espace mémoire, est de codé les dites sorties d'une manière à réduire leur nombre. Pour un codage binaire, par exemple, les 28 sorties sont ramenées à 5 bits ($2^5=32$). Le numéro de la sortie active n'est pas donné par l'ordre de cette sortie mais plutôt par la combinaison de toute les sorties actives.

On peut utiliser l'une ou l'autre des deux méthodes reste cependant à signaler que la deuxième est plus exposée à une mauvaise reconnaissance plutôt que la première. En effet dans cette méthode la sortie active est la combinaison de plusieurs sorties. Si jamais il y a confusion de reconnaissance entre caractères, on ne peut pas le savoir. Tout se passe comme si cette confusion n'aura jamais lieu ce qui n'est pas vrai. Dans la première par contre, à chaque caractère est affectée une sortie uniquement pour lui. Si une confusion entre deux caractères ou plus a lieu, le système le signalera par l'activation de plusieurs de ses sorties et une procédure de séparation devrait être alors activée.

On ce qui nous concerne nous nous sommes servis de la première méthode. Les résultats ainsi que les discussions correspondantes sont détaillés ci-dessous.

6-3 Résultat et discussion

6-3-1 Taux de Reconnaissance des caractères arabes manuscrits isolés

Avant de parler reconnaissance proprement dite, disons d'ors et déjà que notre système de reconnaissance est bâti sur l'idée de populations et de banques spécialisées.

- Une banque spécialisée, est normalement le résultat d'apprentissage d'un ensemble d'individus de même catégorie qu'on appelle population. Dans sa configuration la plus simple, une population peut contenir un seul individu.

Nous l'avons déjà mentionné au chapitre VI que le **fichier de reconnaissance** est la combinaison de plusieurs opérations d'apprentissages correspondante chacune à une catégorie de scripteurs et donnant naissance à une **banque spécialisée**. A la fin de chaque opération d'apprentissage, le programme se charge d'enrichir ce fichier avec la nouvelle banque qui vient d'être déterminée et mettre à jour le paramètre « N » (le nombre de population).

Pour mener une étude comparative, deux méthodes d'apprentissage ont été appliquées : La méthode classique et la nouvelle méthode qu'on propose. Pour cela, 20 scripteurs sont utilisés. Chaque scripteur a écrit les 28 caractères arabes puis:

2. 80% (16 scripteurs) de cet ensemble à servi à déterminer une banque de reconnaissance générale unique selon le procédé classique. C'est-à-dire que les moments de ces scripteurs sont disposés les uns à la suite des autres, et sont utilisés pour l'apprentissage du réseau.
3. Ces 80% ont été utilisés pour créer 16 banques individuelles spécialisées pendant la phase d'apprentissage. Toutes ces banques ont été rassemblées dans une banque unique qui va être utilisée pendant la phase de reconnaissance.

Ces deux opérations étant effectuées, les 20% restant, c'est-à-dire qui n'ont pas servis à l'apprentissage du réseau dans les deux cas, sont utilisés dans la phase de reconnaissance.

6-3-1-1 Comparaison entre les deux méthodes

Méthodes	Taux de Reconnaissance		
	Moments Invariants (10)	Moments de Zernike (30)	Descripteurs de Fourier (23)
Méthode classique (une seule banque)	41%	52%	89%
Nouvelle Méthode (plusieurs banques)	75%	88%	96%

Tableau 6.2 : Comparaison des taux de reconnaissance en utilisant la configuration classique et la nouvelle configuration du réseau

Sous forme d'histogramme ce tableau apparaît comme l'indique la figure 6-1

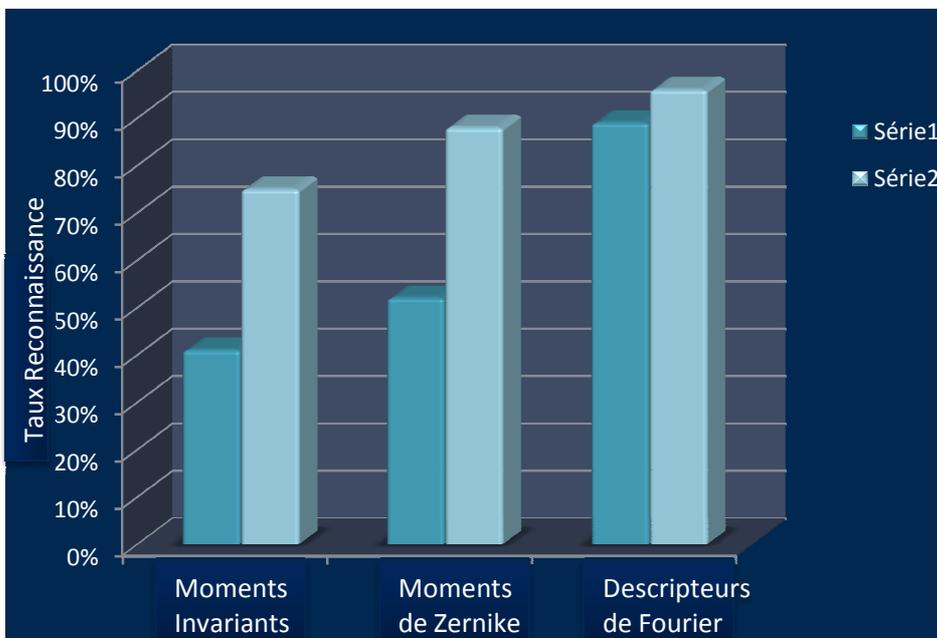


Figure 6-1 : histogramme comparatif des taux de

Reconnaissance de l'ancienne et la nouvelle méthode

série1 : Méthode classique

série2 : Nouvelle méthode

Il est bien clair que :

- ☛ C'est la deuxième configuration qui donne de bons résultats au sens des taux de réussites en reconnaissance ;
- ☛ Que les descripteurs de Fourier gardent toujours leur position en tant que meilleurs descripteurs.

6-3-1-2 Comparaison entre les différents types de découpage

Pour voir le sens, en plus ou en moins, de l'influence du découpage sur le taux de reconnaissance nous avons opérés un découpage des caractères étudiés horizontalement, verticalement et dans les deux sens ensuite :

- ☛ Un apprentissage est exécuté, individuellement, sur chacune des parties des différents caractères. Une banque spécialisée pour cette partie est alors générée. Pendant la phase de reconnaissance, c'est le vecteur des caractéristiques correspondant à cette partie qui est utilisé (voir tableau 6.3);
- ☛ L'autre possibilité, celle que nous avons adoptée, est de rassembler les vecteurs des caractéristiques de chacune des parties du caractère, en un vecteur unique de dimension multiples. Cette configuration permet donc d'augmenter le nombre d'entrées du réseau et laisse croire en une bonne reconnaissance (voir tableau 6.4).

Les résultats obtenus sont résumés dans le tableau ci-dessous.

Découpage	Architecture du réseau	Moments Invariants
Caractère entier (CE)	7x40x28	34%
Horizontal bas (HB)	7x40x28	29%

Horizontal haut (HH)	7x40x28	35%
Vertical droite (VB)	7x40x28	21%
Vertical gauche (VG)	7x40x28	36%
Droite haut (DH)	7x40x28	39%
Droite bas (DB)	7x40x28	42%
Gauche haut (GH)	7x40x28	23%
Gauche bas (GB)	7x40x28	16%

Tableau 6.3 : Comparaison des taux de reconnaissance pour chacune des parties de découpage.

Sous forme d'histogramme ce tableau apparaît comme l'indique la figure 6-2



Figure 6-2 : histogramme comparatif des taux de reconnaissance de chacune des parties de découpage.

Découpage	Architecture du réseau	Moments Invariants
Caractère entier (CE)	7x40x28	35%
Horizontal bas/ Horizontal haut (H)	14x40x28	40%
Vertical droite/Vertical gauche (V)	14x40x28	42%
Droite haut/Gauche haut/ Droite bas/Gauche bas (HV)	28x40x28	50%

Tableau 6.4 : Comparaison des taux de reconnaissance en utilisant la configuration classique et la nouvelle configuration du réseau

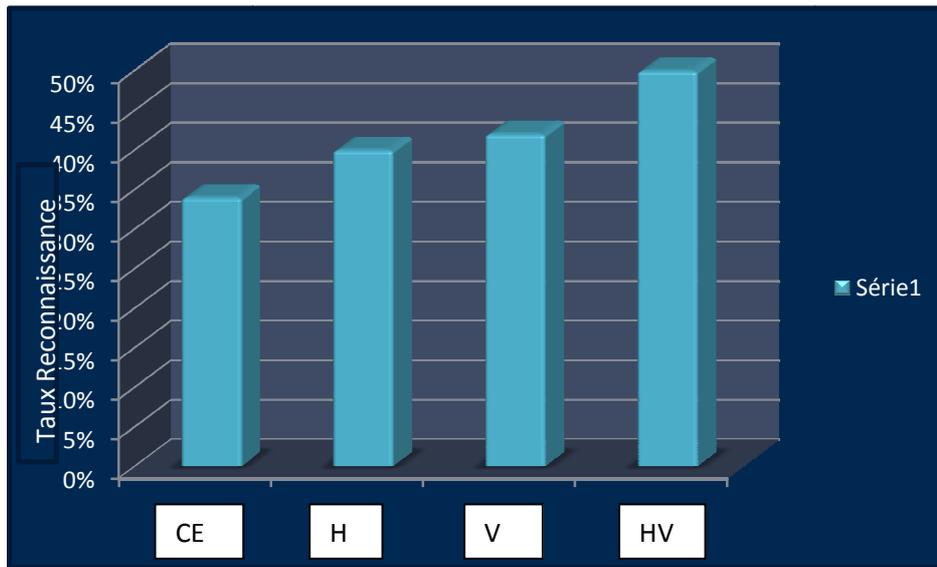


Figure 6-3 : histogramme comparatif des taux de reconnaissance de chacune des parties rassemblées de découpage.

Il apparaît claire que le fait de rassembler les vecteurs des caractéristiques de toutes les parties de l'image en un seul vecteur donne de bien meilleurs taux de reconnaissance. Cette conclusion devrait être affinée en poussant le traitement sur une population plus importante de scripteurs.

Le tableau 6.5 montre l'architecture du réseau correspondant aux trois méthodes).

	Moments Invariants	Moments de Zernike	Descripteurs de Fourier
Architecture des réseaux	10x40x28	30x30x28	23x40x28
Facteur d'apprentissage	0.5	0.5	0.5
Momentum	0.2	0.2	0.2
Erreur RMS	0.01	0.01	0.01
Nombre d'images utilisées	448	448	448
Nombre d'images	112	112	112

De tests			
Test sur les images d'apprentissage	100%	100%	100%
Nombre d'images reconnus parmi l'ensemble du test	84	98	107
Taux de réussite	75%	88%	96%

Tableau 6.5 : Architecture du réseau des trois méthodes

Le réseau a été entraîné sur un nombre de 448 caractères et les tests ont été effectués sur un nombre de 112 caractères. Au vue des résultats obtenus :

- Nous pouvons conclure déjà que le réseau apprend bien ce qu'on lui fait apprendre puisqu'il reconnaît les caractères qui ont servis à sa création ;
- Ensuite que les descripteurs de Fourier sont bien meilleurs au niveau du taux de reconnaissance.

6-3-1-3 Conclusion

L'utilisation de la nouvelle méthode a donnée une amélioration claire (nette) du taux de reconnaissance. Cette amélioration est obtenue juste par utilisation des caractéristiques globales (statistique) des caractères. Un taux de reconnaissance plus élevé peut cependant être obtenu par l'ajout d'autres caractéristiques propre à la langue traitée (telles nombre de points, nombre de corps etc pour la Langue Arabe

6-3-2 Reconnaissance des mots arabes manuscrits

Le paragraphe (5-2-5) du chapitre 5 nous a servis à décrire la technique de segmentation d'un mot arabe manuscrit. Cette technique est bien évidemment valable dans le cas d'une analyse de tout un texte du moment que ce texte est composé de mots et que ces mots sont verticalement séparés par des espaces blancs. En prenant comme

échantillon d'étude un mot, l'algorithme d'écrit ne perd donc en rien de sa généralité. Partons donc du mot arabe « الحاسوب ». Ce mot est composé de quatre pseudos mots. On commence par calculer l'histogramme vertical de ce mot. Le résultat obtenu est celui de la figure (6.4).

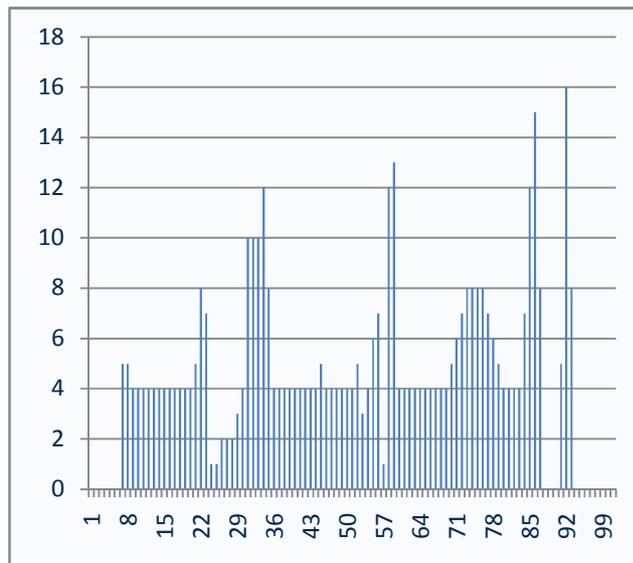


Figure 6.4 : histogramme vertical du mot « الحاسوب »

Opérant maintenant une suppression de la ligne de base au niveau du mot « الحاسوب ». On se retrouve avec une ligne horizontale toute blanche et des blancs qui apparaissent entre les différents caractères d'une part mais entre les dents du caractère « س » d'autre part. L'algorithme de reconnaissance est développé en sortie à rassembler les fragments ainsi obtenus jusqu'à l'obtention d'un caractère valide. Le va et vient est permanent entre le module d'assemblage des fragments et le module de reconnaissance jusqu'à ce que ce dernier signale que le résultat de la reconnaissance est un caractère arabe. Les choses se passent de la manière suivante :

- D'abord le mot est dépourvu de ses points diacritiques ;
- Le mot restant est dépourvu de sa ligne de base ;
- Les points de segmentation sont calculées ;

- Le mot est revêtu de ses points diacritiques ;
- Le processus de reconnaissance-assemblage est activé.

Le résultat de l'exécution de cet algorithme est donné ci-dessous.

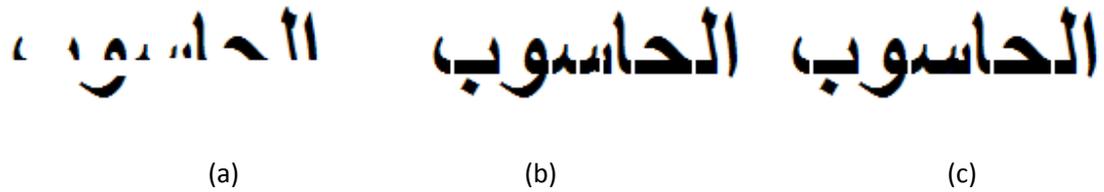


Figure 6.5 : Segmentation du caractère « الحاسوب »

(a) Suppression de la ligne de base

(b) Résultats avant post-traitement

(c) Résultats après post-traitement

Le résultat de l'application du même algorithme sur tout un texte peut être consulté en annexe C.

Pour tester le degré de réussite de cette technique, plusieurs textes ont été segmentés les taux de reconnaissances sont allées jusqu'à 90% dans certains textes.

Conclusions et Perspectives

Dans ce travail de thèse, nous nous sommes principalement intéressés à la reconnaissance de l'écriture arabe manuscrite.

Malgré les efforts et les travaux intensifs réalisés dans le domaine de la reconnaissance Optique de l'écriture, aucun système OCR n'est jugé fiable à 100%. Mais au fur et à mesure Les auteurs essayent d'améliorer les scores par de meilleurs résultats.

L'écriture arabe et l'écriture latine cursive ont de nombreux points communs, qui permettent le transfert vers l'arabe de techniques déjà éprouvées sur le latin. Des prétraitements spécifiques à l'arabe sont nécessaires (détection des signes diacritiques, détection de la ligne de base), mais la segmentation en graphèmes, l'extraction de primitives et le moteur de reconnaissance sont les mêmes que ceux utilisés pour la reconnaissance de l'écriture latine.

Diverses extensions pourraient être envisagées pour améliorer le taux de reconnaissance.

- **Choix des caractéristiques** : dans le chapitre 3 nous avons décrit les différentes techniques d'extraction des caractéristiques à partir de l'image d'un caractère arabe manuscrit. Ce n'est pas une limitation par rapport à un traitement d'un texte entier, parce que la finalité de ce dernier est une décomposition en caractères. Le nombre de caractéristiques donné par une formulation quelconque est limité. Le génie est d'essayer de trouver d'autres caractéristiques, qui permettent une séparation meilleure entre les différents caractères, ou une utilisation particulière de ces caractéristiques. Notre travail a constaté à coupler les deux. C'est-à-dire chercher à augmenter le nombre de caractéristique et utiliser une nouvelle méthode d'utilisation de ces caractéristiques. Pour ce dernier point, en plus des moments en nombre de sept donnés par la formulation utilisée (moments invariants), nous avons utilisés une **nouvelle restructuration du réseau** pendant la phase de d'apprentissage. Cette organisation fait apparaître le coté génération ou population. Pour chaque génération de scripteurs, un réseau spécialisé est crée pendant la phase d'apprentissage. Le réseau utilisé dans la

phase de reconnaissance (banque) est la combinaison de tous les réseaux correspondant aux différentes générations et créés pendant la phase d'apprentissage. On a cherché à développer un système ouvert dans le sens où il prend en compte de nouvelle génération récemment arrivées. On peut donc à tout moment ajouter de nouvelles banques à la suite de celles déjà existantes. C'est un système extensible. Cette organisation est dictée par un souci de donner au réseau les meilleures chances de reconnaître l'écriture de scripteurs de différentes générations. Une comparaison, entre l'ancienne technique et notre nouvelle technique, permet de se contenter quant à l'amélioration des taux de reconnaissance obtenus.

- ☛ D'un autre côté, on a investi une technique toute nouvelle : la **technique de découpage**. Toujours dans un souci d'augmenter le nombre de caractéristiques séparatrices entre les différentes classes, cette technique pratique un zoom à l'intérieur de l'image du caractère. Un découpage de l'image est opéré dans plusieurs directions, choisis par l'utilisateur, et les caractéristiques de chaque région sont extraites individuellement. Ce découpage n'est a opéré, bien sûr, que si le traitement de l'image entière n'a pas permet une séparation nette entre les différents caractères. La également une étude comparative, entre l'utilisation de l'image entière et l'utilisation de sous images, a été menée. Les résultats obtenus sont plutôt encourageant. Reste cependant à augmenter la population de scripteurs pour conclure à l'efficacité (ou non) de cette méthode.

ANNEXE A : INTERFACE UTILISATEUR

1. Simulation et logiciel

Au cours de la simulation de toutes les parties du système, nous avons adoptés une implantation modulaire. Chaque fonction est matérialisée par un programme spécialisée qui réalise la tâche correspondante. Il n'est pas question, dans cette thèse, de donner les programmes proprement dit. Nous nous contentons plutôt de donner l'ensemble des modules avec les fonctions qu'ils réalisent.

Ces programmes sont écrits soit en langage C soit en MATLAB. En voici la liste

2. Module de prétraitement

Ce module s'occupe du prétraitement des images pour les préparer à l'extraction des primitives il contient les modules suivants :

- Sous module qui détermine le cadre qui délimite uniquement l'image (enlève la frontière blanche de l'image si elle existe).
- Sous module qui normalise l'image pour la ramener a une dimension (largeur_normalisee_en_pixel hauteur_normalisee_en_pixel) Quelque soit sa dimension d'origine. Elle comble le vide entre les lignes et les colonnes dans l'image normalisée.
- Sous module qui calcule les moments absolus à partir de ces moments on détermine le centre de gravité d'image .
- Sous module qui opère un découpage de l'image en sous images. Le découpage est effectuée par rapport au centre de gravite de l'image entière
- Sous module qui calcule les histogrammes horizontal et vertical d'une image
- Sous module qui sauvegarde l'image entière dans le cas d'un changement d'échelle ou chacune des sous images dans le cas d'un découpage horizontale
- Sous module qui détermine le contour de l'image.

3. Programme de calcul des moments géométrique d'une image:

Ce Module calcul :

- Moments Absolus
- Moments Centrés Normalisés
- Moments Invariants
- Moments Affines

Ce module calcule les moments centrés normalisés. Les moments résultants sont des moments invariants par rapport à la rotation, à la translation et à l'échelle.

4. Module de calcul des Moments de Zernike et moments affine

- Sous module qui calcule les moments invariants partir des moments centrés normalisés
- Sous module qui calcule les moments invariants affines partir des moments centrés
- Sous module qui sauvegarde tous les moments dans un fichier

5. Module de calcul des descripteurs de Fourier

Ce module est composé de deux sous modules :

- Un sous module qui calcule les FD d'une image.
- Un sous module qui détermine les caractéristiques des contours secondaires.

6. Module de calcul le vecteur des caractéristiques des histogrammes de direction

Module de calcul les caractéristiques du squelette

- Les tâches de ce module sont : l'extraction du squelette et détermination des caractéristiques du squelette.

7. Réseau

C'est un assemblage de plusieurs modules spécialisés qui exécute les deux opérations : apprentissage/reconnaissance.

- Module qui permet de lancer une opération d'apprentissage selon deux formes:
 - a- Initier une opération d'apprentissage
 - b- Poursuivre une opération d'apprentissage
- Module qui permet d'initier une opération d'apprentissage.
- Module qui permet de créer une nouvelle banque.
- Module qui permet de lancer un apprentissage pour enrichir une banque déjà existante avec une nouvelle population.
- Module de lancement de l'apprentissage pour créer une nouvelle banque ou pour enrichir une banque existante.
- Module qui permet de poursuivre un apprentissage interrompu.
- Module qui permet un apprentissage supplémentaire si le nombre d'itération est atteint sans que la précision de calcul ne soit atteinte.
- Module qui permet de lancer un nouvel apprentissage pour enrichir avec une nouvelle population une banque déjà existante.
- Module qui ouvre le fichier qui contient les entrées du réseau (les moments) et les sorties correspondantes pour l'ensemble des échantillons (patterns) d'apprentissage
- Module de récupération des entrées d'attaque du réseau (sorties de la couche d'entrée) et l'état des sorties du réseau pour une opération d'apprentissage
- Module qui réalise une propagation en avant dans le réseau en calculant les sorties des neurones de toute les couches depuis la première couche cachée jusqu'... la couche de sortie y compris
- Module de retour en arrière si la précision de calcul n'est pas atteinte par calcul des erreurs en sortie de chaque neurone
- Module qui vérifie la correspondance entre les sortie désirées et les sorties calculées
- Module de collecte des informations sur les couches du réseau pour une opération d'initialisation d'apprentissage
- Module de création des sorties des neurones et des erreurs de ces Sorties
- Module de création en mémoire des matrices des poids

- Module de création en mémoire des matrices des erreurs des poids pour une opération d'initiation apprentissage ou de poursuite d'apprentissage
- Module d'initialisation des matrices des poids par des valeurs arbitraires pour une opération d'apprentissage. Les matrices des erreurs des poids sont initialisées par des zéros (0)
- Module d'initialisation des matrices des erreurs des poids par des zéros (0) pour une opération d'initialisation d'un apprentissage ou une poursuite d'apprentissage.
- Module qui ajuste les poids pour une nouvelle itération
- Module qui réalise une sauvegarde temporaire de la structure du réseau des matrices des poids et des matrices des erreurs des poids pour poursuivre l'apprentissage ultérieurement.
- Module qui sauvegarde la structure du réseau et les matrices des poids des neurones pour une opération d'apprentissage réussie
- Module de chargement de l'entête de la banque après une opération d'apprentissage réussie ou d'un fichier de sauvegarde transitoire pour la poursuite de l'apprentissage.
- Module de chargement de l'entête du fichier de sauvegarde transitoire pour la poursuite de l'apprentissage
- Module de chargement des matrices des poids et des erreurs des poids du réseau pour une opération de poursuite d'apprentissage
- Module de sauvegarde de l'entête de la banque après une opération d'apprentissage réussie.
- Module de sauvegarde des matrices de poids et erreurs des poids pour poursuivre l'apprentissage ultérieurement.
- Module de sauvegarde des matrices de la population.
- Module qui réalise une opération de reconnaissance en combinant les réponses des réseaux des différentes populations de la banque par rapport à une entrée
- Module de récupération des entrées d'attaque du réseau (sorties de la couche d'entrée) pour une opération reconnaissance.

- Module de masquage des sorties. Elle ne laisse passer que les sorties qui répondent ... certains critères de points et de trous
- Cette interface permet aussi de montrer la progression du réseau pendant l'apprentissage en affichant : le graphe de l'erreur (RMS), le pourcentage du nombre d'itérations par rapport au nombre total et le taux de réussite du réseau.

8. Interface utilisateur

L'interface graphique qui permet l'accès aux différentes fonctions du système est développée sous BUILDER C++. C'est une interface interactive qui permet à l'utilisateur, et en toute simplicité, d'opérer un choix dans l'ensemble des options offertes. Elle est organisée sous forme de bloc spécialisé : Bloc prétraitement, Bloc extraction des primitives, Bloc réseau, ...etc. Le passage entre les différents blocs est effectué par un simple clic de souris. On peut toujours revenir sur une option active pour la désactiver et vice versa. La figure ci-dessous donne une image de cette interface.

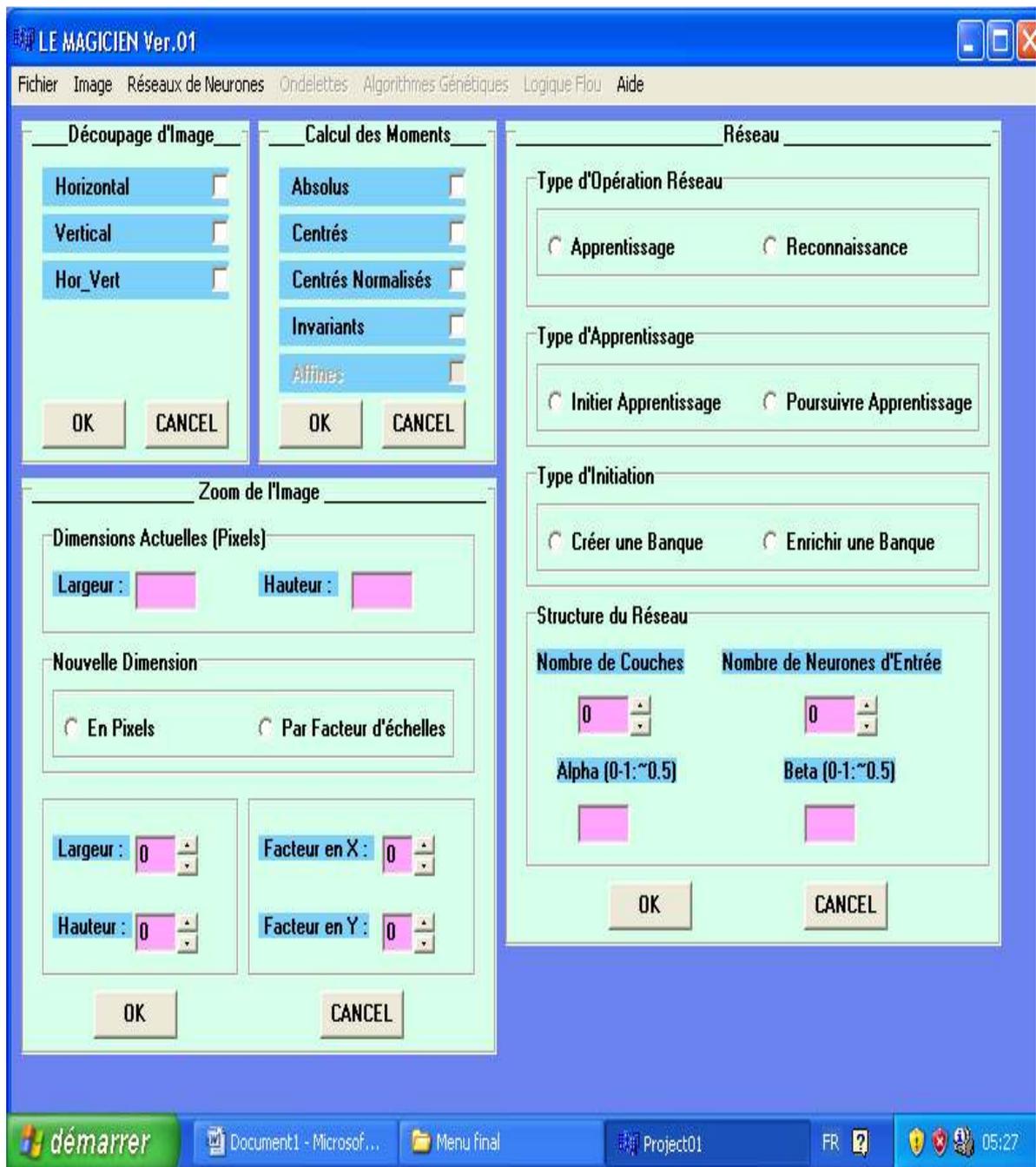


Figure A.1: Interface Utilisateur

ANNEXE B : ORGANIGRAMMES

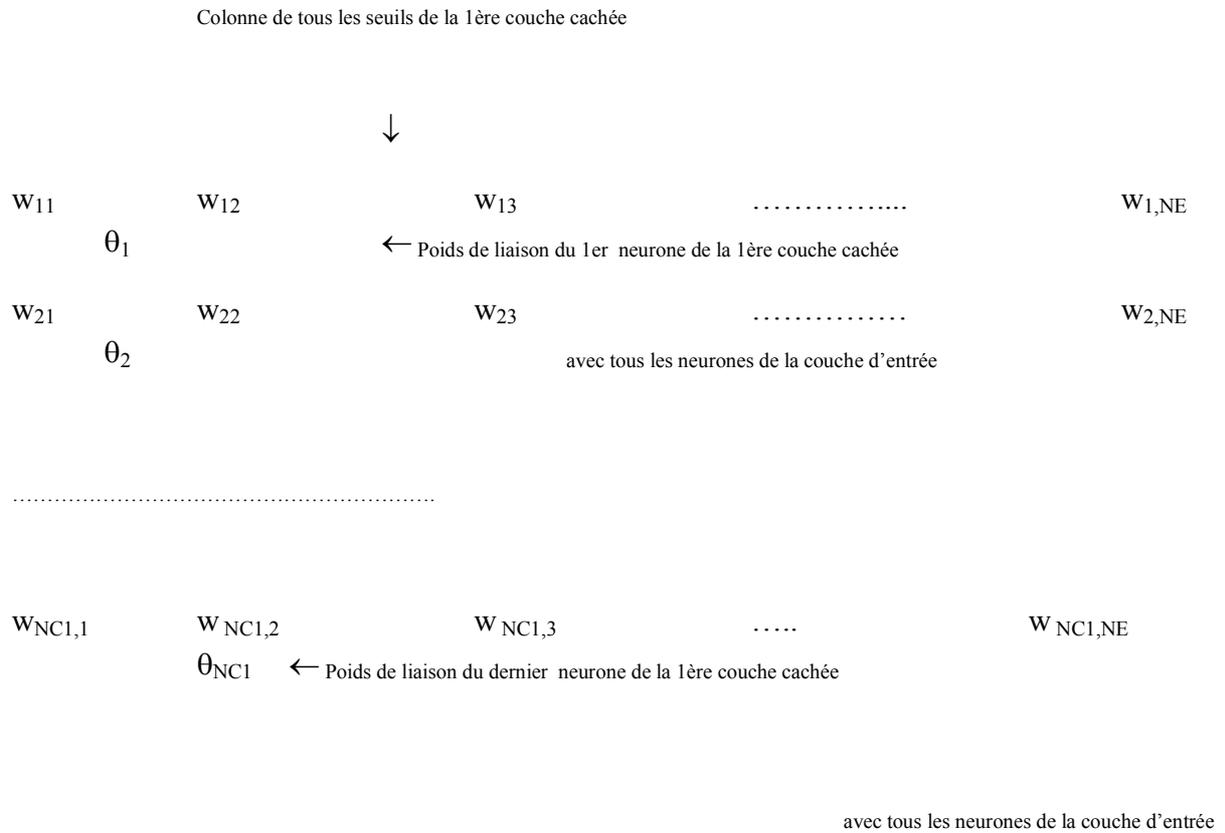
1. La structure du fichier de sauvegarde des poids est la suivante :

- La première ligne contient des informations sur la structure du réseau dans l'ordre suivant :

Nombre de couches total du réseau nombre de neurones de la couche d'entrée nombre de neurones de la 1^{ère} couche cachée nombre de neurones de la 2^{ème} couche cachée ... nombre de neurones de la couche de sortie

- Les matrices des poids comme définis ci-dessous.

2. Les matrices des poids et erreurs des poids ont la configuration suivante

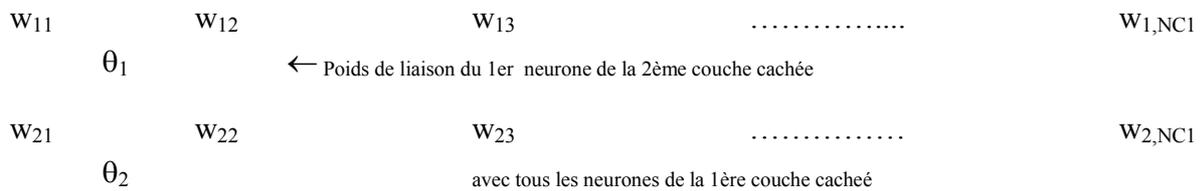


Matrice N°0 de dimension (NC1x(NE+1)) entre la couche d'entrée et la première couche cachée

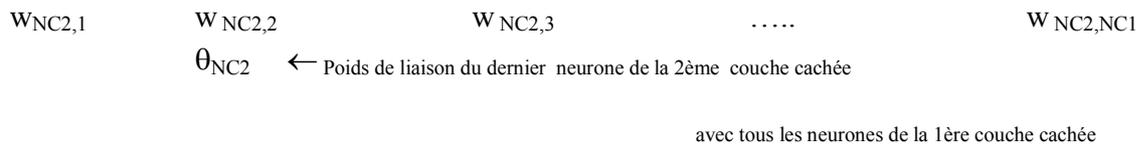
NE : Nombre d'éléments de la couche d'entrée

NC1 : Nombre d'éléments de la première couche cachée

Colonne de tous les seuils de la 2ème couche cachée



.....



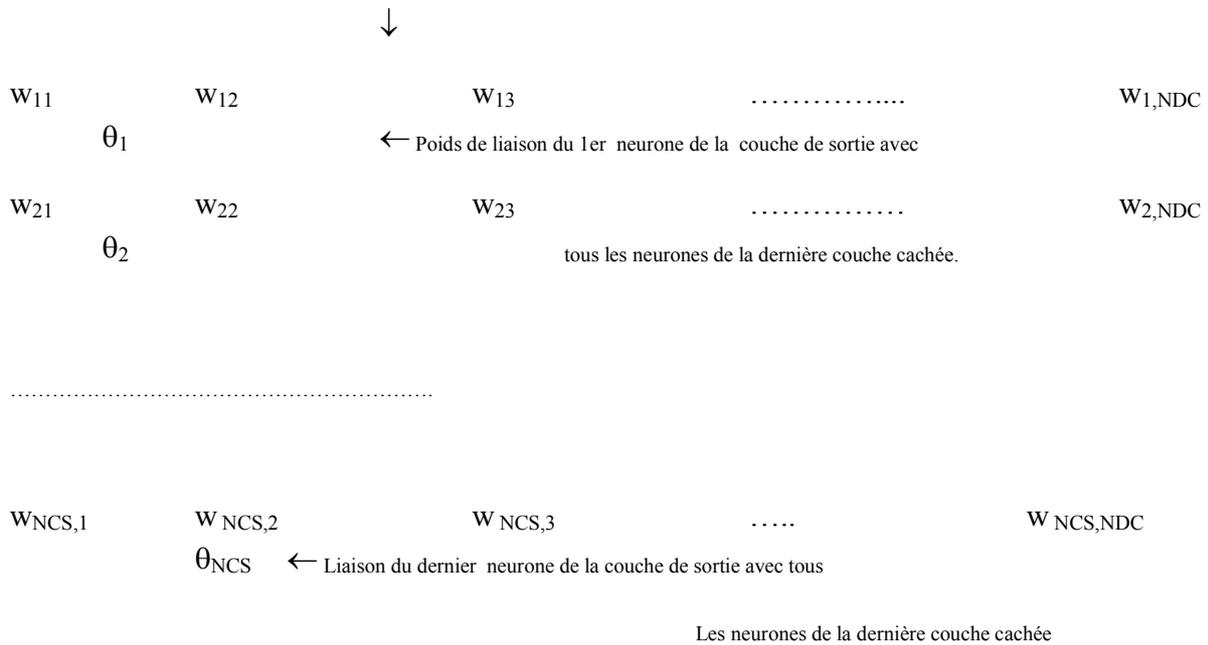
Matrice N°1 de dimension (NC2x(NC1+1)) entre la 1^{ère} et la 2ème couche cachée

NC1 : Nombre d'éléments de la première couche cachée

NC2 : Nombre d'éléments de la deuxième couche cachée

.....

Colonne de tous les seuils de la couche de sortie



Matrice $N^\circ = \text{nombre de couche total} - 2$:

de dimension $(NCS \times (NDC + 1))$ entre la dernière couche cachée et la couche de sortie

NCS : Nombre d'éléments de la couche de sortie

NDC : Nombre d'éléments de la dernière couche cachée

☞ La dernière colonne de ces matrices concernent les seuils des neurones de chaque couche. Il faut garder présent à l'esprit que ces neurones ne sont pas liés entre eux. Il constituent uniquement des entrées aux neurones des couches.

☞ Les matrices des erreurs de liaisons sont disposées de la même manière ;

☞ L'accès aux éléments de ces matrices se fait de la manière suivante :

Poids[I][J][K] où:

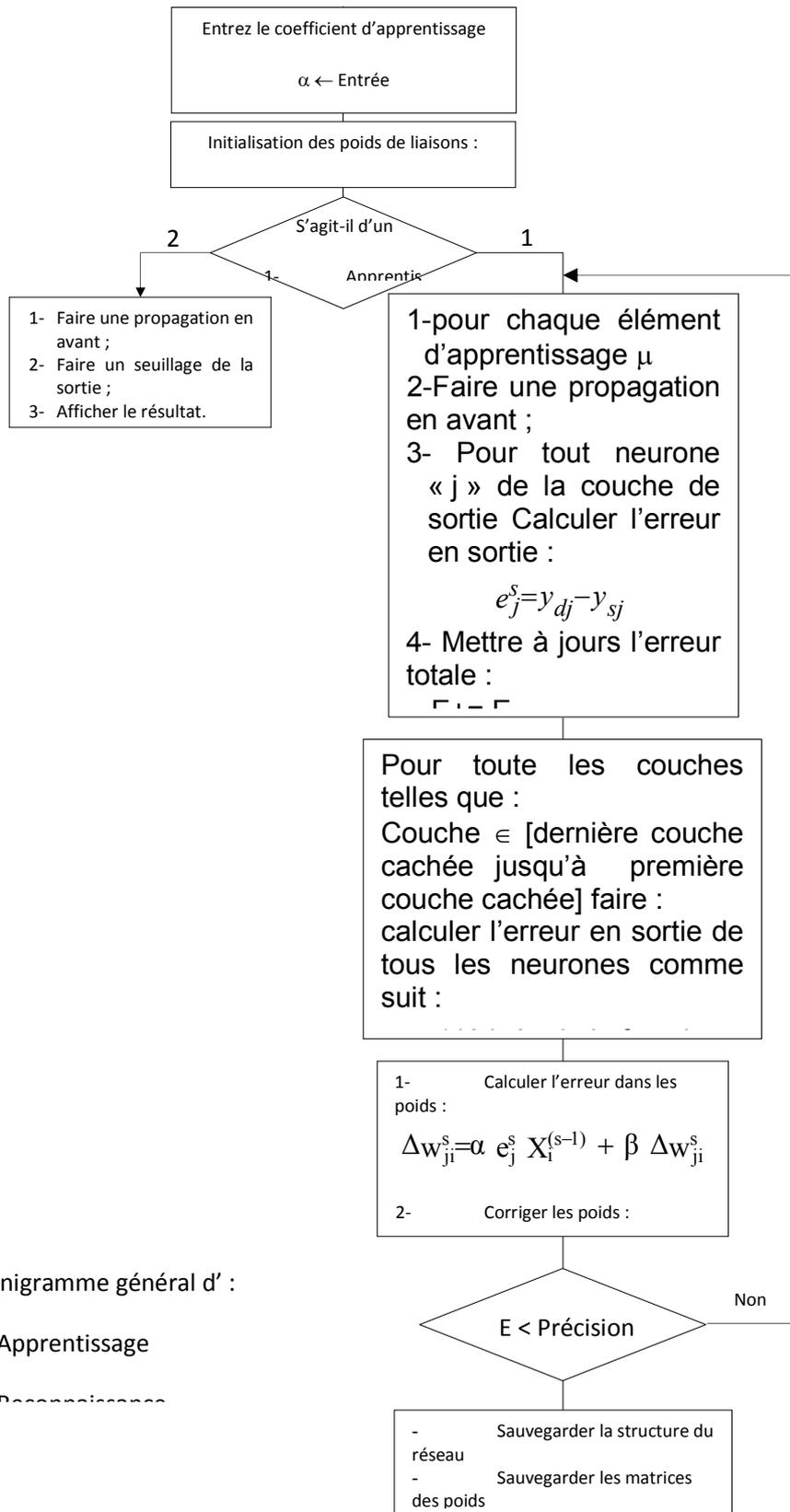
I: désigne le N° de la matrice $I \in [0, \text{nombre de couche total} - 2]$

J :désigne la ligne en cours de traitement à l'intérieur de cette matrice c.à.d le numéro du neurone de la couche (I+1) . $J \in [0, \text{nombre d'éléments de la couche (I+1)}]$

K :désigne la colonne en cours de traitement c.à.d le neurone de la couche I.

$K \in [0, \text{nombre d'élément de la couche I}]$

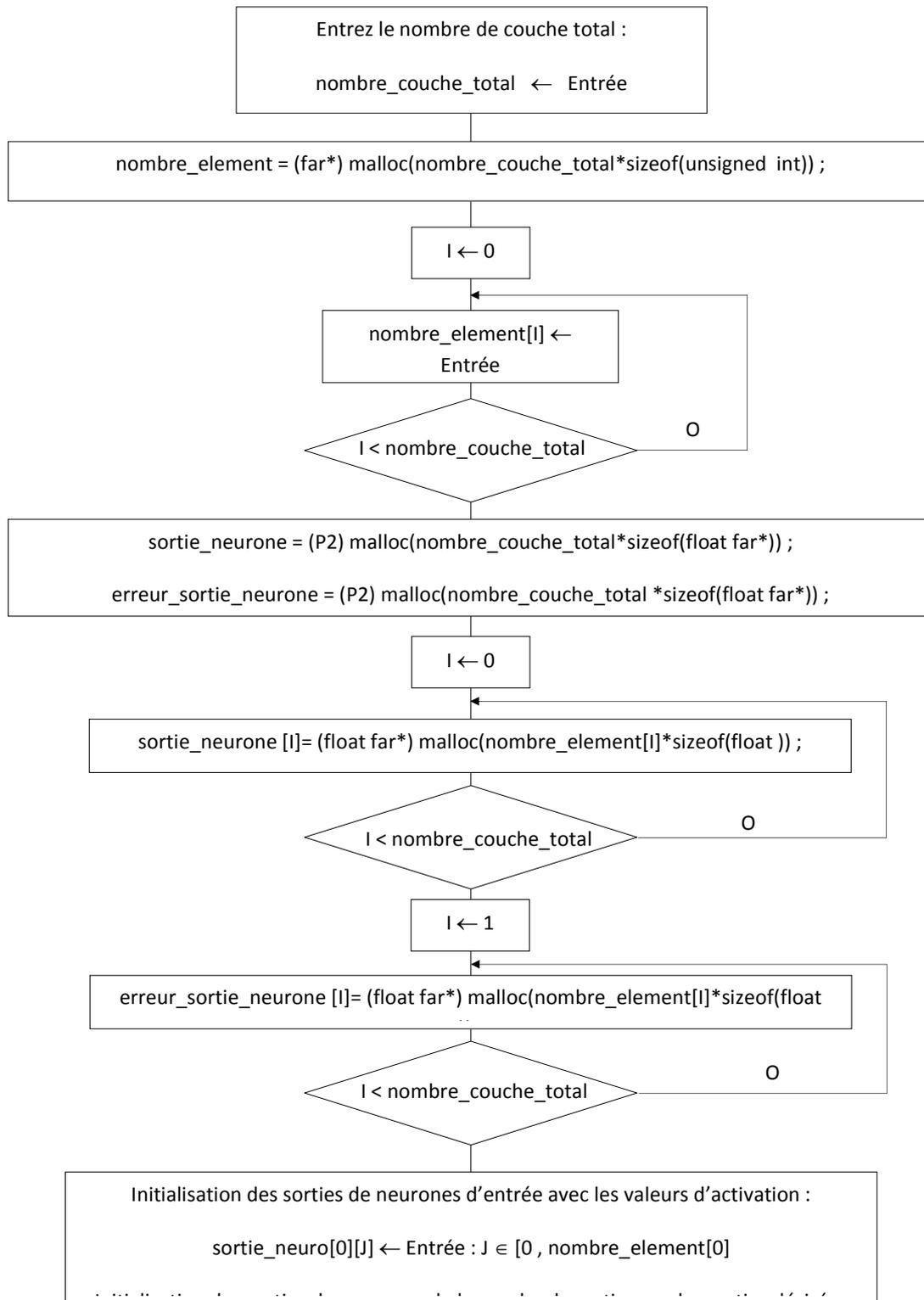
☞ Les erreurs des sorties des neurones concernent toute les couches sauf la couche d'entrée. C'est pourquoi pour les réservations mémoire on a commencé à partir de l'indice 1 (voir organigramme ci dessous)



Organigramme général d' :

- Apprentissage

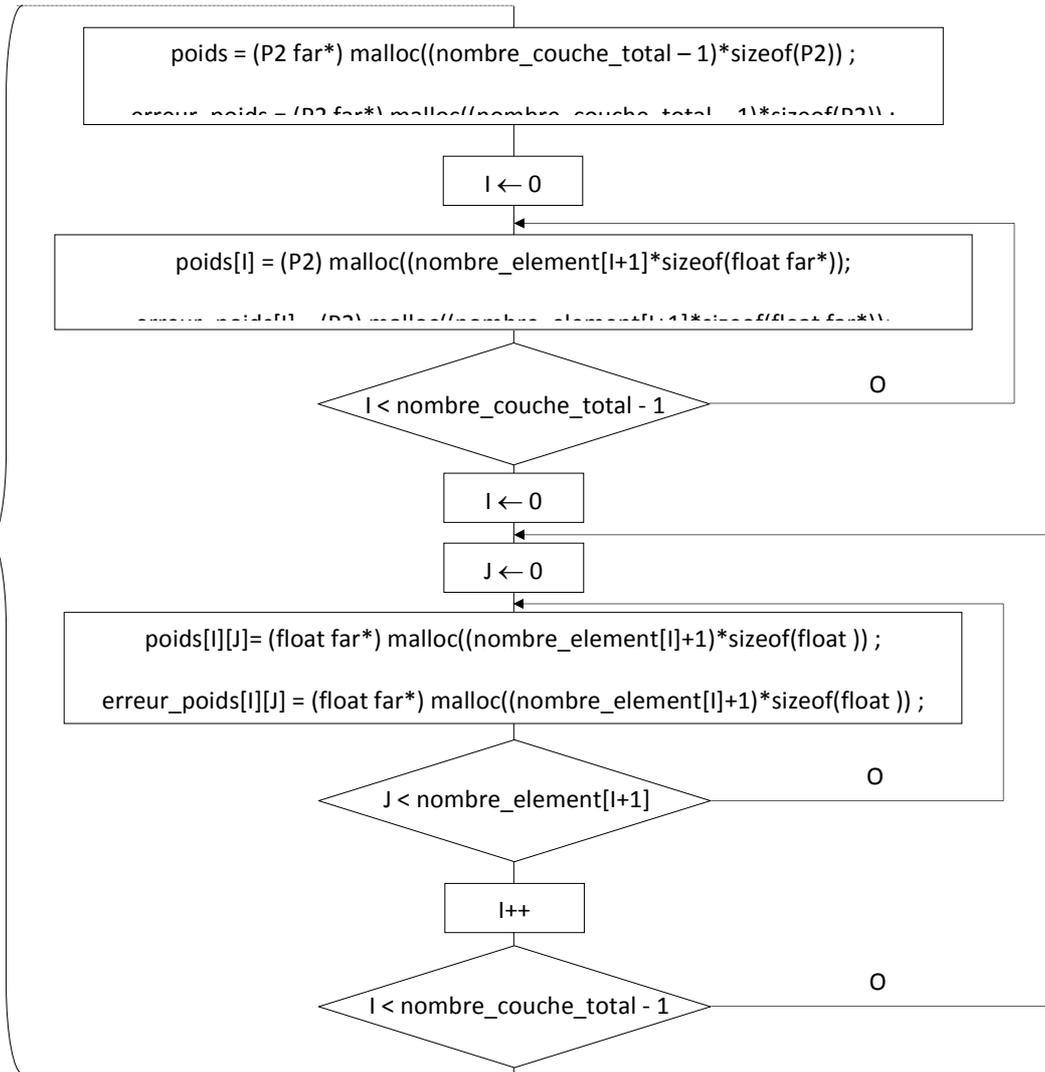
Reconnaissance



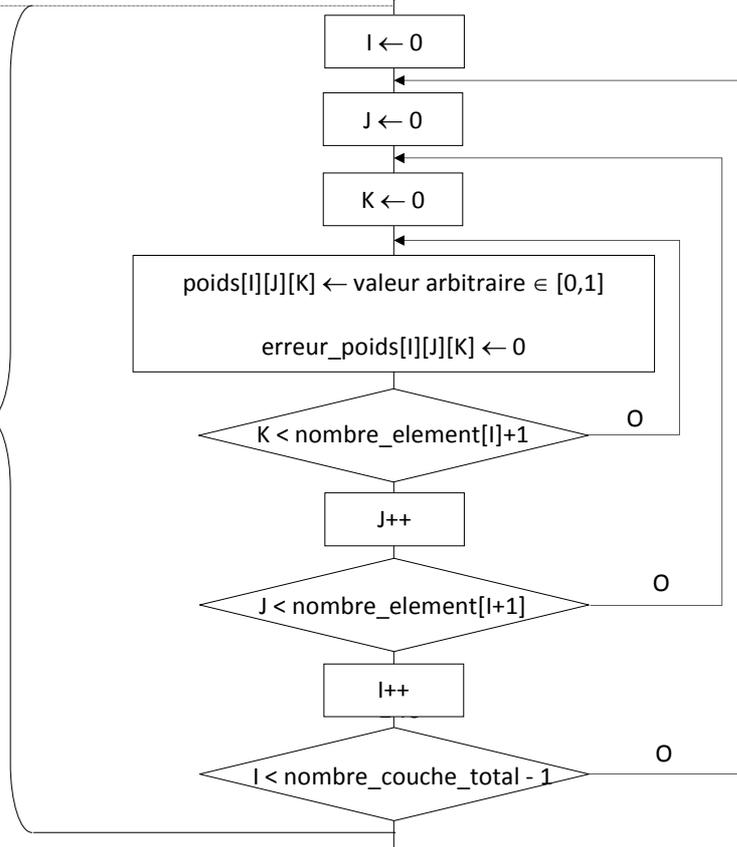
Organigramme de réservation de l'espace mémoire pour les sorties

des neurones, les sorties d'entrée et les erreurs en sortie

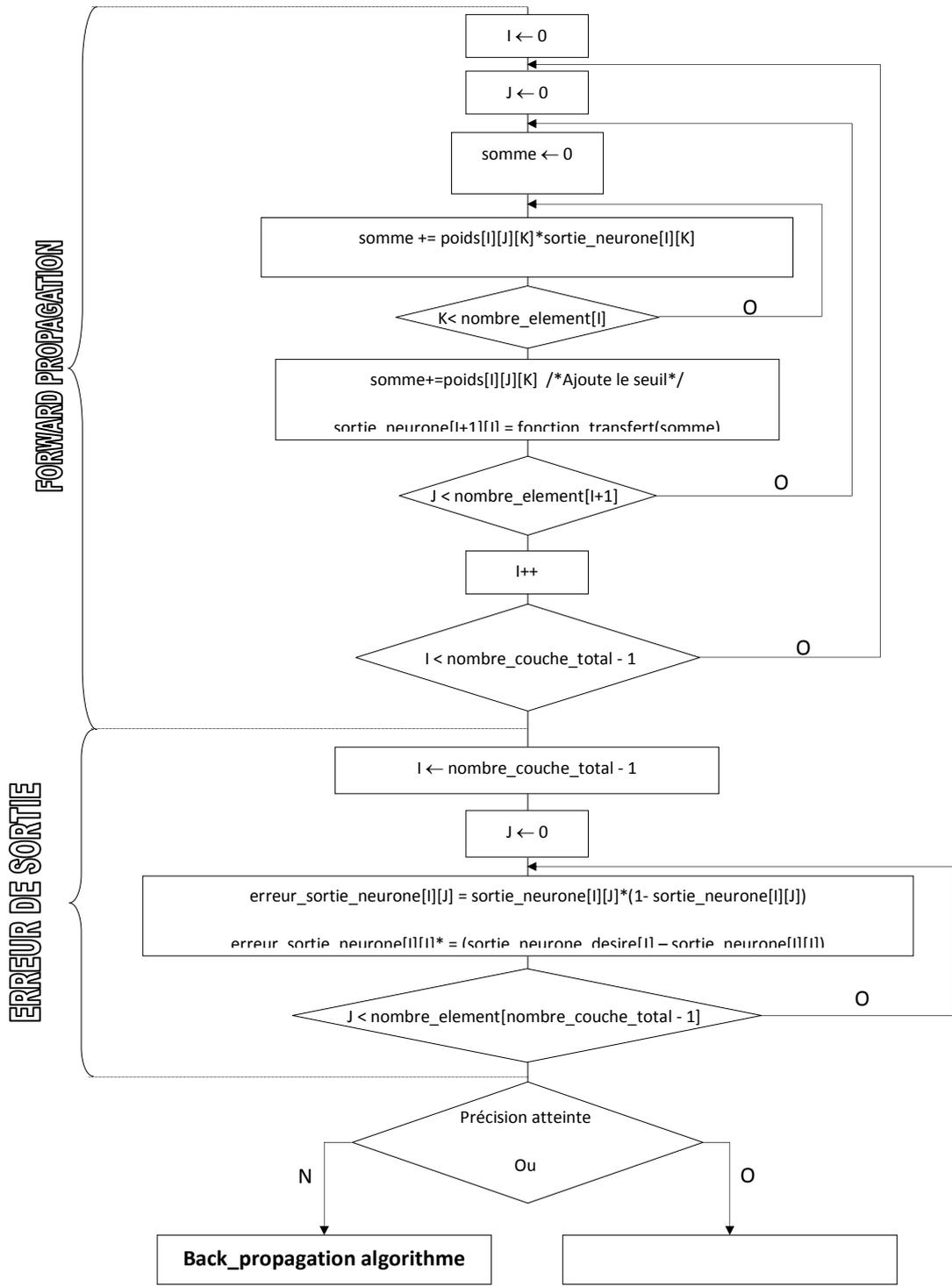
RESERVATION



INITIALISATION



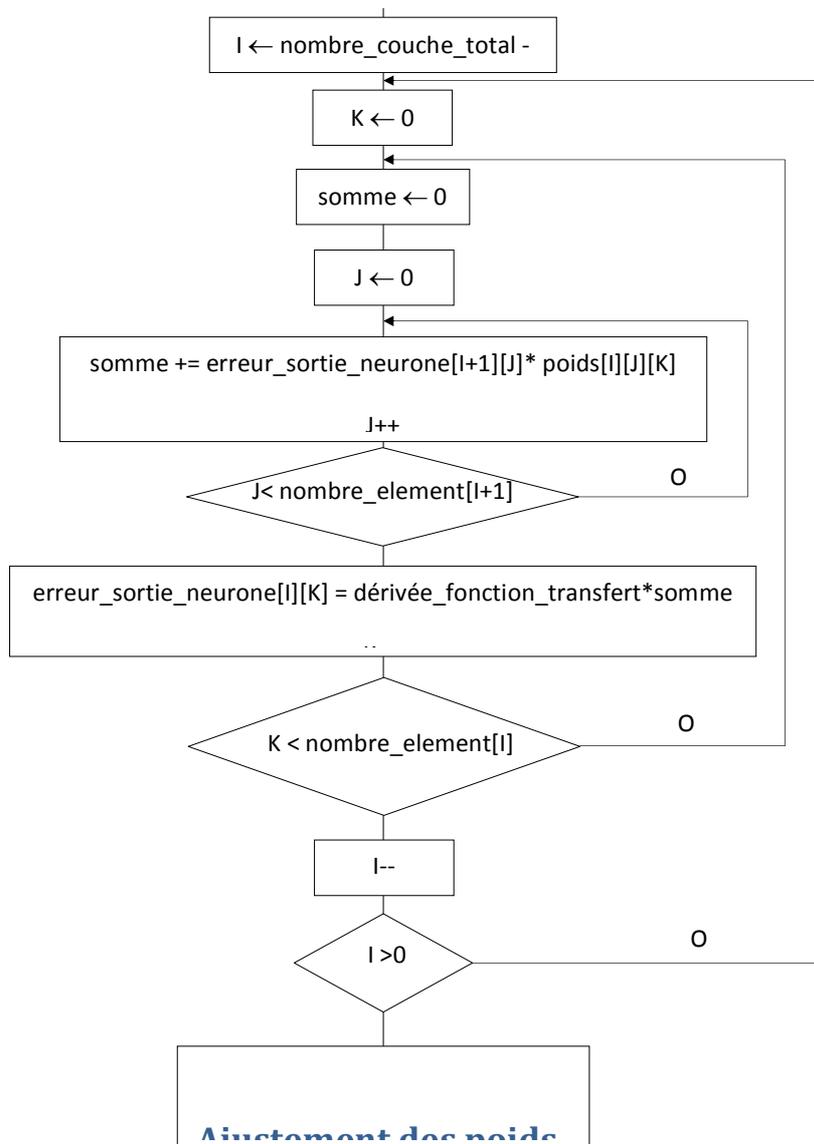
ORGANIGRAMME DE CREATION DES MATRICES DES POIDS ET DES ERREURS DES POIDS ET LEUR INITIALISATION



ORGANIGRAMME DE PROPAGATION AVANT
 ET CALCUL DES ERREURS EN SORTIE

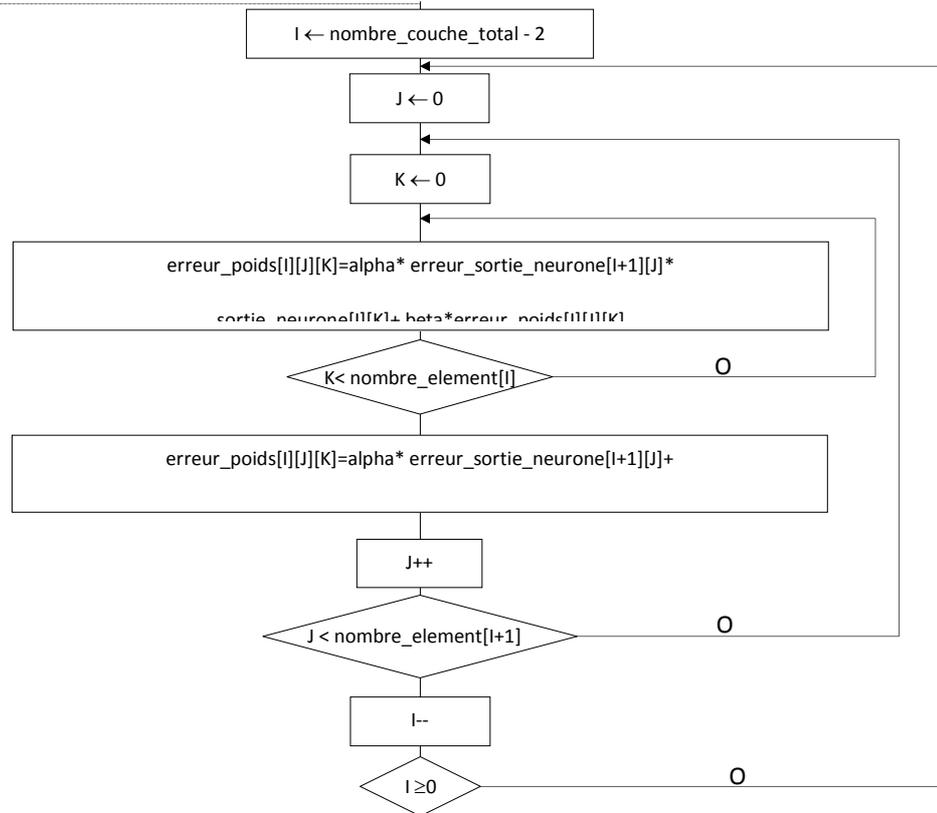
ORGANIGRAMME DE PROPAGATION AVANT ET

CALCUL DE L'ERREUR EN SORTIE

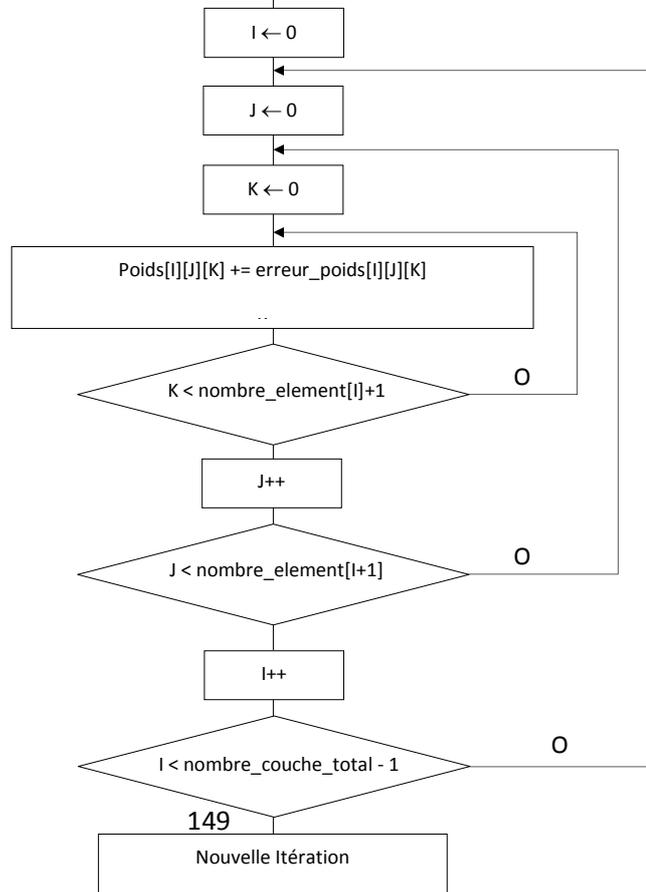


ORGANIGRAMME DE PROPAGATION ARRIERE

CALCUL DES ERREURS DES POIDS: DELTA W_{ij}



CORRECTION DE TOUS LES POIDS



Dans l'étape d'apprentissage chaque échantillon de la bibliothèque d'apprentissage est présenté au réseau et l'ensemble des opérations suivante a lieu :

- 1- Calcul de la contribution de cet échantillon dans l'erreur quadratique globale ;
- 2- Rectification de tous les poids ;
- 3- Passage à l'échantillon suivant.

L'étape d'apprentissage prendra fin quand la précision de calcul est atteinte.

ANNEXE C

Lignes de mots avant segmentation

حکم و امثال
حُسْنُ الخلق يذیب الخطایا كما تذیب الشمس الجلید
یساعد الله الذین یساعدون أنفسهم

Avant Post-traitement

حکم و امثال
حسن الخلق يذیب الخطایا كما تذیب الشمس الجلید.
یساعد الله الذین یساعدون أنفسهم

Après Post-traitement

حکم و امثال
حسن الخلق يذیب الخطایا كما تذیب الشمس الجلید
یساعد الله الذین یساعدون أنفسهم