الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

14/95

وزارة التربية الوطنية
MINISTERE DE L'EDUCATION NATIONALE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT Electronique

PROJET DE FIN D'ETUDES

SUJET

Etape de decision et Reconnaissance des caractères arabes multifaille multifonte par méthode structurelle

Proposé par : Mme HAMMAMI

Etudié par : Mr. ZEROUATI ALI Fayçal
Mr. ZEKRINI Benyahia.

Dirigé par Mme HAMMAMI

PROMOTION
1995

E.N.P. 10, Avenue Hacen Badi El-Harrach - ALGER

École Nationale Polytechnique

Département d'électronique

Projet de Fin d'Etudes

Sujet:

Reconnaissance des Formes
Etape Décision et Reconnaissance utilisant une étape structurelle

Proposé par
Mme. Hammami

étudié Par:
Mr. Zerouati Ali Fayçal.
Mr. Zekrini Benyahia

Soutenu devant le jury composé de:

Mlle. M. GUERTI	Présidente
Mme. L. HAMMAMI	Rapporteur
Mr. BERKANI	Examinateur

le 03 Juillet 1995.

A

ma Mère

mon Père

mes Soeurs

et mes Frères

A. Si Ferroukh

A

mon père

ma mère

mes sœurs

mes frères

et à tous mes amis.

Benzahia.

Remerciements

REMERCIEMENTS

On tient à remercier, vivement Mme L.HAMMAMI, qui nous a confié ce travail, et qui nous a guidé et encouragé tout le long de notre travail.

On désir également témoigner notre sincère reconnaissance a Mlle S.AITDAOUD post-graduante a l'ENP qui nous a toujours aidé avec beacoup de gentillesse, trouve ici notre sincère affection.

On remercie particulièrement Mr. HAMLI, Mr.BOUZETINE et Mlle S.Mounira de l'ICS de Baulieu (EL-HARRACH) de l'intéret qu'il ont apporté a notre travail.

Enfin on remercie notre ami A.BENGHARABI pour son aide qui nous a été précieux.

SOMMAIRE

Introduction Générale

INTRODUCTION GENERALE

La reconnaissance des formes est une étape dans un processus de comprehension de notre univers. Elle permet de passer d'une forme, qui appartient a un monde physique et qui est pleine d'informations utiles et non utiles, a une representation plus simple qui la caracterise.

Donc la R.F s'occupe de la résolution de plusieurs problèmes qui sont liés au codage des formes, a leur paramétrisation et a leur discrimination; mais cette tâche s'avére trés difficile a cause de l'appartenance des formes a un espace physique plein de contraintes qui se repercutent sur la transcription numérique qui devient trés complexe et cela a cause de l'absence de capteurs adaptés a toutes les situations ,et aussi la nature des formes et leur apparence qui varient d'un échantillon a un autre (même au sein d'une même famille), ce qui multiplie la dimension de l'espace de représentation et augmente le temps de décistion.

Aprés la transcription numérique de la forme, la R.F procede à la caractérisation de la formes qui la singularise et qui la rapproche de ses formes voisines d'une part et l'éloigne de ses fausses semblables d'une autre part.. Cette tâche s'appelle l'apprentissage, et constitue des famillles de forme ayant les même caracteristiques.

Pour arriver a ce groupement il faut utiliser des caractèristiques robustes et adéquates et éviter leur redondance pour faciliter la séparation des familles. Un autre aspect de la R.F est la décision, qui revient a attribuer a la forme d'entrée un nom correspondant a celui d'une famille de l'apprentissage. Dans ce cas on cherche parmi les familles celle qui maximise une fonction de correspondance ou de ressemblance.

Pour limiter le nombre de comparaisons on définit pour chaque famille quelques prototypes qui serviront d'unique réference pour les comparaisons. Ces comparaisons sont operées sur les caractèristiques, d'où l'importance de la caractèrisation dans la decision, et bien evidemment sur le resultat de la reconnaissance.

Dans le cadre de ce mémoire nous nous attachons a décrire d'une manière détaillée le 1er niveau de reconnaissance a savoir la détéction des concavités et le 2ème niveau qui est le calcul des primitives secondaires.

Dans le ***chapitre I*** afin de mieux situer notre travail nous présenterons une description générale d'un système de Reconnaissance de Forme.

Dans le ***chapitre II*** nous parlerons sur la reconnaissance optique de l'écriture, nous présenterons une description générale d'un **AOCR** ***(Automatic Optical Character Recognition)*** et nous décrirons les différents modules qui le compose.

Le ***chapitre III*** éxpose l'algorithme de détéction de concavités, c'est a dire nous montrerons comment les boucles et les concavités sont détéctées ,et comment détécter les primitives secondaires ,ce chapitre décrit aussi l'algorithmed'apprentissage et celui de la reconnaissance,on parlera aussi du fomat **TIFF** et de la structure du dictionnaire établi.

Dans le ***chapitre IV*** nous présenterons la performance de notre système **OCR** en faisant une analyse statistique indiquant les taux et les temps de reconnaissance .

CHAPITRE I

DESCRIPTION GENERALE D'UN SYSTEME DE RECONNAISSANCE DES FORMES

CHAPITRE I

DESCRIPTION GENERALE D'UN SYSTEME DE RECONNAISSANCE DE FORMES

I.1 - INTRODUCTION :

La démarche classique suivie en **R.F** consiste a opèrer suivant la (Figure I.1). Il faut noter que ce schema n'est pas purement linéaire ; des interactions peuvent apparaître entre les différents niveaux pour un éventuel retour en arriére.

Nous allons donner la fonction de chaque module de la figure I.1.

I.2 - DESCRIPTION DES MODULES :

1 - Le monde physique :

Ce processus (figure I.1) part du monde physique qui est un espace analogique de dimension infinie appellé espace des formes. Les objets dans cet espace ont une multitude de propriétés qu'il faut simplifier pour pauvoir passer au monde discret.

2 - Acquisition:

C'est un processus dont le rôle est de transformer l'image analogique,obtenue par un capteur physique, en une image numérique.Parmi ces capteurs on cite le scanner et la caméra.

Le codage est une partie essentiel suivant la partie d'acquisition. C'est l'opération de conversion numerique du monde physique continu vers un monde numerique discret.

Ce dernier est appelé espace de représentation qui a une dimension finie mais importante,choisie volontairement grande pour pouvoir disposer d'un maximum

d'informations sur la forme et a pouvoir selectionner des sous ensembles pour de multiples usages.

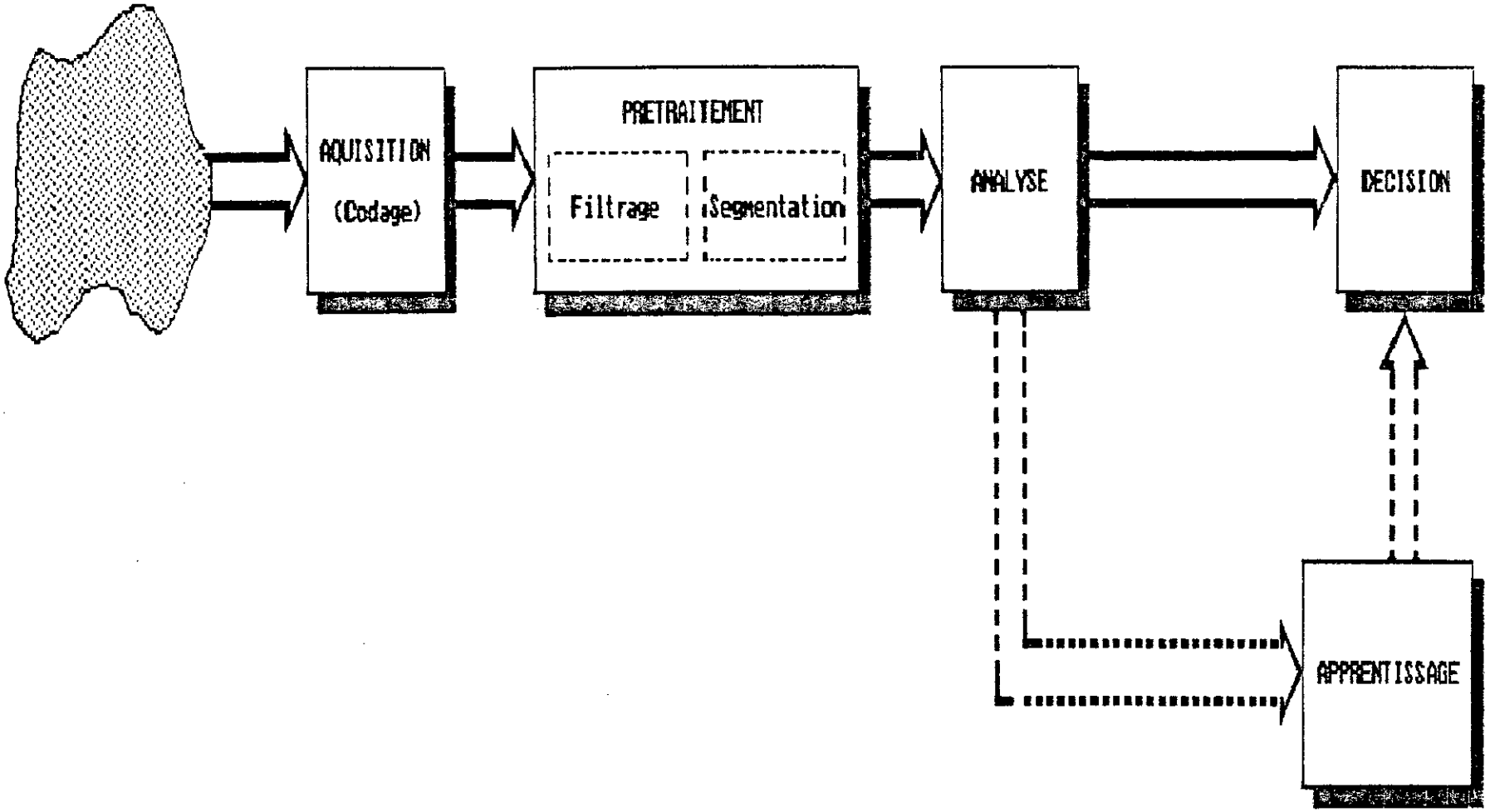


Figure I.1. schéma général d'1 système de reconnaissance de forme.

3- Pretraitement :

Cette étape consiste a réduire la dimension de l'espace de représentation et ceci en selectionnant les informations necessaires a l'application.

Cette selection passe souvent par l'élimination du bruit dû aux conditions d'acquisition(opération de filtrage), par la correction des erreurs (utilisation d'opérateurs morphologiques : dilatation et erosion) ainsi que par la réduction des données.

4 - Analyse :

Lors de cette étape, les techniques de R.F calculent un certain nombre de paramètres qui sont de nature géometrique, topologique ou statistique et servant comme données représentant la forme; ils sont limités en nombre. Ainsi l'espace des paramètres obtenu est de dimension très petite par rapport a celui de représentation.

5 - Apprentissage:

Il tente de définir des modéles de reférence ou de caracteriser des classes de decision ; il permet ainsi de dicter au système l'algorithme de décision le plus adéquat vis a vis des régles de modélisation choisies. L'espace ainsi obtenu s'appelle l'espace des noms, sa dimensio correspond au nombre de modèles éxistants.

6 - La Décision :

La décision est l'étape de reconnaissance proprement dite. Son rôle est d'identifier la forme a reconnaître a partir de l'apprenissage réalisé.Les critères utilisés pour la comparaison dans la décision sont les mêmes que ceux utilisés pour l'apprentissage. La réponse de la décision peut être :

1 - Le nom de la forme en cas de bonne reconnaissance.
2 - Plusieurs noms en cas d'ambiguité.
3- Ou bien le rejet de la forme en cas d'incompatibilité des descriptions, avec les formes de reference.

I.3 - CONCLUSION :

En général, un système de R.F est évalué par deux critères qui sont: la vitesse et le taux de reconnaissance.

La vitesse de reconnaissance est étroitement liée à la méthode utilisée, au langage de programmation ainsi qu'au matériel utilisés.

Par contre le taux de reconnaissance est lié à la dimension du dictionnaire (nombre de de prototypes acquis par apprentissage).

CHAPITRE II

RECONNAISSANCE OPTIQUE DE L'ECRITURE (O.C.R)

CHAPITRE II

LA RECONNAISSANCE OPTIQUE DE L'ECRITURE (OCR)

II.1 - INTRODUCTION :

Le but de la reconnaissance de l'écriture est de transformer un texte écrit en une représentation comprehenssible par une machine et facilement reproductible par un traitement de texte. Cette tâche n'est pas triviale car les mots possédent une infinité de représentations dûe aux différentes polices de caractères qui éxistent pour l'imprimé avec de nombreux styles.

Suivant le type d'écriture (manuscrite ou imprimée) un système doit reconnaître les opérations à effectuer et les résultats peuvent varier notablement.

En particulier il est obligatoire dans certains cas, d'effectuer une segmentation en vue d'isoler les caractères, chose qui n'est pas facile.

II.2 - HISTORIQUE :

La reconnaissance de l'écriture est mieux connue sous le nom d'**OCR** ***(Optical Character Recognition)*** du fait de l'emploi de procédés d'acquisition optiques. Son origine remonte aux années 1900 au cours desquelles on inventa le scanner à balayage pour la télévision et les machines à lire. ***"Tuyrin"*** développa alors la première application d'aide aux handicapés visuels, mais il a fallu attendre jusqu'à 1940 pour voir se réaliser la première version informatique de cette application ; les premières applications étaient essentiellement orientées vers les chiffres imprimés et quelques caractères latins. [1]

II.3 - APPLICATIONS :

L'*ORC* connait plusieurs applications pratiques dans plusieurs domaines d'activités parmi lesquelles on peut citer :

1 - Les Banques : pour l'authentification des chèques (correspondance entre montant et libellé d'une part et entre l'identité du signataire et sa signature d'une autre part).

2 - La Poste : par la lecture des adresses et le tri automatique du courrier.

3- Les Télécommunications : pour l'échange de fichiers informatisés a distance.

4- La Police et la Sécurité : pour la reconnaissance des empreintes; l'authentification du manuscrit et l'identification du scripteur.

5 - Les Affaires et l'Industrie : pour la gestion des stocks et la reconnaissance de documents techniques.

6 - L'Administration : pour la lecture automatique des documents administratifs.

II.4 - LES DIFFERENTS ASPECTS DE L'OCR :

Il existe plusieurs systèmes *OCR* selon le type de données traitées et bien sûr de l'application visée; voici quelques aspects de l'*OCR* :

- Reconnaissance de l'imprimé ou du manuscrit: L'approche n'est pas la même selon qu'il s'agit de l'imprimé ou du manuscrit. Dans l'imprimé les caractères sont bien allignés et souvent bien séparés verticalement (cas du Latin) ce qui simplifie la phase de lecture. Dans le cas du manuscrit les caractères sont souvent liés ce qui nécessite l'emploi de techniques de délimitation très spécifiques.

- Reconnaissance monofonte, multifonte, omnifonte: La question se pose pour un texte imprimé. Un système est dit monofonte s'il ne traite qu'une fonte a la fois ; il est dit multifonte s'il est capable de reconnaître un mélange de quelques fontes préalablement apprises; enfin un système est dit omnifonte s'il est capable de reconnaître toute fonte sans l'avoir apprise, ce qui relève actuellement du domaine de la recherche.

- Reconnaissance en ligne(on-line) ou hors ligne (off-line) : La première, dynamique se fait pendant l'écriture, elle permet de corriger ou de modifier l'écriture de manière directe et instantanée. Tandis que la deuxième démarre après la fin de l'acquisition du document entier, elle permet d'analyser un grand nombre de caractères au prix d'un prétraitement coûteux.

II.5 - ACQUISITION :

La première étape d'un système *OCR* consiste à digitaliser l'écriture et la présenter au système sous une forme lisible. Cette tâche n'est pas simple à causede la diversité des formats et de la qualité de l'écriture et du papier, le capteur utilisé est soit du mode statique ou du mode dynamique.

- Le mode statique : Il utilise essentiellement des scanners. Le scanner balaye le texte ligne par ligne, chacune est digitalisée en une serie de points. La resolution du scanner est exprimée en nombre de points par pouce ***(DPI "Dot Per Inch")***, les valeurs les plus utilisées ou courantes sont entre 200 et 400 *DPI*..

- Le mode dynamique : Il utilise la tablette graphique, qui envoie au contact du stylo des coordonnées de points.La précision est de l'ordre de 0.1mm, a condition que l'appui sur le stylo soit regulier de manière à ne pas produire des discontinuités dans les traits. Ce mode est très précieux en *OCR* car il donne des renseignements très profitables à la reconnaissance comme par exemple le sens de l'écriture.

voir figure I.1.

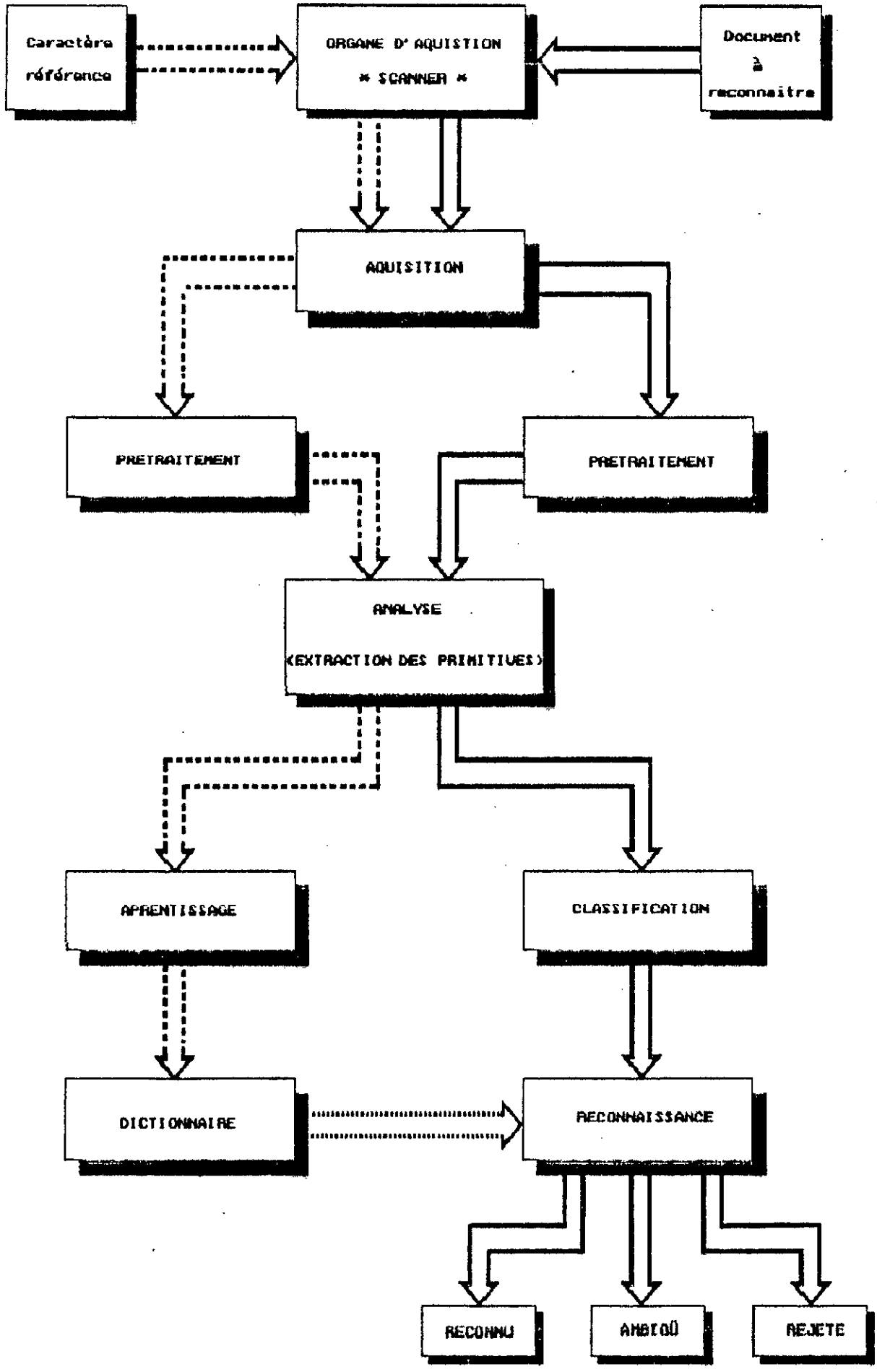


figure II.1. Description d'un système O.C.R.

II.6 - PRETRAITEMENT :

Le rôle du prétraitement est de préparer les données reçues par le mode d'acquisition, a la phase d'analyse qui procède à l'extraction des paramètres et des primitives. Donc il faut, pour que la phase d'analyse soit fiable, que les données soient: dénuées du bruit, corrigées de leurs erreurs, normalisées si c'est nécessaire et même réduite si leur nombre est trop grand.

- ***Supression du bruit*** : Ce bruit est dû en général aux capteurs du fait de la quantification et aussi des conditions de l'expérience (éclairage, état du document...). En général ce bruit est impulsionnel et se manifeste, dans le cas des images, par l'apparition de points isolés sur l'image ce qui rend sa supression possible par des techniques de filtrage.

- ***Homogénéisation des données*** : Elle consiste à éliminer de la forme l'information redondante, superflue et inutile pour notre application.

On connait ici plusieurs techniques; celles qui améliorent la qualité des données en accentuant les détails significatifs et celles qui selectionnent directement l'information pertinente.

- ***Normalisation des données*** : Si un système de reconnaissance est conçue pour une taille bien déterminée alors on utilise la normalisation pour pouvoir reconnaitre les formes les plus petites ou les plus grandes. La normalisation consiste a dilater ou a contracter les données sans, bien entendu, perdre l'information necessaire.

Les techniques de prètraitement comme le filtrage (élimination de bruit) ainsi que l'érosion et la dilatation (correction des erreurs dans l'image) ont été largement traitées dans les travaux précédents. Pour cela nous avons jugé inutile de reprendre cette partie[4].

II.7- L'ANALYSE :

Elle consiste à extraire les caractéristiques et primitives essentielles et utiles pour l'application et les exprimer sous une forme numérique. Une primitive peut être définie comme une forme élementaire constituée par une agglomération de pixels a laquelle il est possible de donner une interprètation, par exemple une boucle, une concavité, convexité, intersection de deux lignes, un changement de direction, une extrèmité.

Ces problèmes rencontrés ne sont pas seulement reliés à l'extraction des primitives, qui sont plus ou moins difficiles, mais aussi au choix de ces primitives qui doit être fait soigneusement et depend beaucoup de l'application, c'est à dire qu'il ne faut pas prendre des primitives qui ne servent pas à séparer entre deux objets différents ou qui ne figurent pas dans les formes à étudier. Par exemple si on veut reconnaître l'ensemble des formes {carré, rectangle,triangle, losange} on ne doit pas prendre comme primitives les courbes.

- ***Extraction de paramètres*** : On prend l'objet et on le divise en parties sur lesquelles on va effectuer des mesures qui caractériseront la forme. Il y a deux grandes approches d'analyse qui son t:

1 - *L'analyse globale* : Elle est fondée sur l'étude globale de propriétés de l'objet, sans distinction de comparaison ou de structure; on trouve des mesures :

a) numèriques : qui correspondent à des calculs quantitatifs, comme le diamètre, l'aire, la perimètre, l'amplitude, les coefficients de Fourier etc...

b) logiques : qui correspondent à des calculs qualitatifs où l'objet sera représenté par un vecteur logique tel que chaque élément du vecteur indique la présence ou l'absence d'une propriété.

2 - *L'analyse structurelle* : L'approche précédente devient inutile pour des objets riches en informations structurelles. Dans cette approche on n'est pas sensé seulement extraire les primitives et prendre les mesures; mais aussi établir des relations entre les primitives. Les mesures qu'on vient de citer peuvent être faites sur des paramètres géometriques (morphologiques) ou statistiques frequentielles.

a) Les mesures topologiques : sont très utilisées ; citons par exemple:la surface de l'objet, son perimètre, le nombre de concavités leurs directions ...

b) Les mesures d'orientation : Celles appliquées aux caractères possedant des allongements; ces mesures sont réalisées en calculant des moments d'inertie du 2ème ordre.

c) Les mesures de formes : Citons par exemple :

$$\text{compacité} = S/P^2 \qquad \text{(II.1)}$$

avec :

S : Surface de l'objet.

P : Périmètre de l'objet

$$\text{Allongement} = \text{Longueur/Largeur} \qquad \text{(II.2)}$$

Longueur, Largeur de l'image.

d) Les mesures statistiques : Elles consistent à analyser la distribution statistique des données. Ici on trouve plusieurs mesures statistique comme par exemple la moyenne,la variance,les moments etc....

e) Les mesures rationnelles : Elle consistent à calculer des rapports entre regions. On trouve les paramètres de taille (une entité est plus grande ou plus petite q'une autre) ; les paramètres de position (on dit que la 1ère entité se trouve à gauche ou à droite de la 2ème).

II.7 - L'APPRENTISSAGE :

Il consiste à établir des caractères prototypes qui serviront de reference lors de la décision.

L'étape d'apprentissage doit suivre automatiquement une classification des caractères.

Classification automatique :

En général dans les systémes de R.F, on associe à chaque forme un vecteur dit vecteur d'attributs, ses composantes sont les caractéristiques de la forme ou les primitives choisies. Ce vecteur représente alors un point dans l'espace de représentation des formes. Classifier les formes revient à classifier leurs vecteurs d'attributs en comparant les composantes de ces vecteurs une à une. La classification doit vèrifier les propriétés suivantes :

1 - *Compacité* : c'est à dire que les formes appartenant à une même classe doivent être plus proches les unes des autres que celles appartenant à des classe differentes.

2 - *Independance* : C'est à dire que les classes doivent être bien séparées, indépendantes et sans recouvrement entre elles. On se base sur la notion de proximité pour attribuer un vecteur de mesure à une classe.

Cette notion de proximité est souvent exprimée par la notion de distance entre les éléments de la même classe, c'est ce qu'on va utiliser dans notre travail.

II.8 - LA DECISION :

La decision est l'étape de reconnaissance proprement dite. Son rôle est d'identifier la forme test à partir de l'apprentissage réalisé.

Les critères utilisés pour la décision sont les mêmes que ceux utilisés pour l'apprentissage. En effet, il est évident que si les critères sont différents on ne peut arriver avec certitude à un résultat cohérent.

Certaines techniques de décision utilisées sont fondées sur la notion de proximité et nécessitent un calcul de distance ou de probabilité de ressemblance avec les modèles définis.

La réponse de décision peut être, selon le cas, le nom de la forme en cas de bonne reconnaissance, plusieurs noms en cas d'ambiguité, ou bien le rejet de la forme en cas d'incompatibilité de description avec les formes de réference.

Notion de distance :

Une distance nous permet de juger si deux objets sont proches l'un de l'autre ou non . C'est une notion trés générale, en R.F, elle peut être évaluée entre deux formes de la même classe ou bien entre une forme et une classe.

Définition:

Soit E un ensemble quelconque de points, E est dit espace métrique réel s'il existe une fonction appelée distance et notée :

$$d : E \times E \rightarrow R ,$$

vérifiant les propriétés suivantes :

1 - Séparabilité :

$$\forall (a, b) \in E^2, a \neq b \Rightarrow d(a,b) > 0 \qquad (II.3)$$

2 - Réflexibilité :

$$\forall a \in E, d(a,a)=0. \qquad (II.4)$$

3 - Symétrie:

$$\forall (a, b) \in E^2, a \neq b \Rightarrow d(a,b) = d(b,a). \qquad (II.5)$$

4 - Inégalité triangulaire :

$$\forall (a, b,c) \in E \times E \times E, d(a,c) <= d(a,b)+d(b,c). \qquad (II.6)$$

Distance entre vecteurs:

Soit E un ensemble de vecteurs, soient :

$$X = \{ xi \}, \qquad \text{(II.7)}$$

$$Y = \{ yi \}, i = 1..n, \qquad \text{(II.8)}$$

avec $(X,Y) \in E^2$.

On définit plusieurs distances comme suit :

- *Distance de Hamming :*

$$d1(X,Y) = \sum_{i=1}^{N} | xi - yi | . \qquad \text{(II.10)}$$

- *Distance Euclidienne:*

$$d2(X,Y) = \sqrt{(\sum_{i=1}^{N} (xi - yi)^2)}. \qquad \text{(II.11)}$$

- *Distance maximum :*

$$d_{\infty}(X,Y) = \text{Max} | (xi - yi) | . \qquad \text{(II.12)}$$

- *Distance dn :*

$$dn(X,Y) = (\sum_{i=1}^{N} | xi - yi |^{n})^{1/n}. \qquad \text{(II.13)}$$

Distance d'un point a une classe :

Très utilisée en R.F, elle permet d'attribuer un élement x a une classe C_k et cela en se basant sur un critère de proximité comme suit :

$$x \in C_k \Leftrightarrow C_k = \text{Argmin}_{Ci}\, d(x, C_i). \qquad \text{(II.14)}$$

.où : Argmin(f(ci)) est la fonction qui donne la classe Ci minimisant la fonction f.

La distance entre un point et une classe:

$$d_1(x,C_i) = \inf [d(x,y) ; y(C]. \qquad \text{(II.15)}$$

$$d2(C_1,C_2) = \inf [d(x,y) ; x \in C_1; y \in C_2]. \qquad \text{(II.16)}$$

Distance binaire:

Certaines caracteristiques ne sont pas mesurables donc on doit leur attribuer un code binaire par exemple :
1 si elles existent et 0 sinon.

Soit *X1,X2* deux vecteurs, tels que:

$$X1 = \{x_i1\}. \qquad \text{(II.17)}$$

$$X2 = \{x_i2\}. \qquad \text{(II.18)}$$

avec : $i=1..n$; $xi1$, $xi2 \in \{0, 1\}$

On definit une variable a comme suit:

$$a = \sum_{i=1}^{N} (x_i \times y_i). \qquad \text{(II.19)}$$

alors: plus a est élèvée, plus x_i1 ressemble à x_i2 .

On définit une autres fonctionc b comme:

$$b = (1 - x_i1)\,(1 - x_i2). \qquad (II.19)$$

tel que ***b*** : **Fonction de dissemblance.**

II.9- Conclusion :

Nous avons fait dans ce chapitre un bilan de ce que nous pensons être l'essentiel en reconnaissance de l'écriture. Nous pensons que le domaine de l'écrit trés ouvert même qu'il éxiste aujourd'hui des systéme commercialisés .

Des problèmes demeurent encore, et n'ont pas trouvé de solutions intéressantes ont été données notament pour le manuscrit.

Pour l'imprimé des solutions intéressantes ont été données notament pour la reconnaissance du multifonte, mais l'omnifonte reste infranchissable et rejoint de ce fait le manuscrit.

Il est trés inportant d'élargir le champs de la reconnaissance en faisant intervenir le contexte pour mieux résourdre les indécisions.

CHAPITRE III

TRAVAIL REALISE

CHAPITRE III

TRAVAIL REALISE
(LOGICIEL)

III-1-INTRODUCTION :

Notre travail consiste à réaliser un système de reconnaissance optique de caractères (OCR) arabes, multitailles, multifontes, et dans toutes les positions possibles de l'écriture du caractère (isolé, lié au début, au milieu, ou bien a la fin du mot) et cela avec une méthode structurelle qui se base sur les caractéristiques morphologiques du caractère arabe.

Cette toute nouvelle approche procède en deux niveaux :

- *Le premier niveau :* utilise les paramètres non-métriques pour grouper l'ensemble des caractères en classes: c'est la classification.

- *Le second niveau* : est purement heuristique, il permet de séparer les caractères de la même classe:c'est l'interprétation.

III - 2 - CARACTERISTIQUES DES CARACTERES ARABES :

L'alphabet arabe comprend 28 lettres plus La Hamza qui a un rôle trés important dans la phonétique et le Lemalif qui est en réalité un caractère composé du Lem et du Alif.

Contrairement aux écritures latines ou bien aux caractères latins l'arabe n'a pas de voyelles, mais plutôt des signes de ponctuations qui peuvent être placés au dessus ou bien au dessous du caractère. Les difficultés rencontrés dans la reconnaissance des caractères arabes sont les suivantes : [3]

a - Il existe un grand nombre de styles et de fontes de l'écriture arabe où l'on trouve un même caractère sous des formes trés différentes comme :

ي – ي – ي س – س – س ح – ح – ح

b - L'écriture arabe est connectée et non pas isolée comme l'écriture imprimée latine, donc on trouve un même caractère dans plusieurs positions par rapport aux mots, soit isolé , lié au début, lié au milieu , ou bien lié a la fin du mot :

ب – بـ – ـبـ – ـب حـ – ـحـ – ـح

c - Il existe des caractères qui se recouvrent, donc on rencontre des problèmes dans la séparation de ces caractères comme :

محمد – الحج – الجزائر – تمت – بحث

d - Il éxiste des caractères qui sont ponctués soit par un, deux ou trois points, ou bien par une Hamza (ء).

e - Les caractères arabes n'ont pas la même hauteur on trouve(ـسـ ، ـبـ) comme on trouve (ـلـ ، ـكـ alors il sera difficile de détécter les bruits.

III - 3 DESCRIPTION DE LA METHODE

Comme on l'a noté , cette méthode procède en deux niveaux:

III-3-1 -PREMIER NIVEAU

Ce niveau est la classification des caractères , qui prend comme critères les concavités (dans tous les sens), les boucles et leurs nombres qui sont les caractéristiques principales,elles sont morphologiques (non métriques).

Pour les concavités, on peut trouver un aspect métrique juste pour éliminer les fausses boucles ou bien les points qui appartiennent a la même concavité.

Le choix de ces caractéristiques permet au système de travailler avec des caractères multitailles, sans avoir recours a une normalisation éventuelle (qui peut causer une déformation du caractère et une perte de temps).

Dans ce premier niveau on aura le nombre de concavités (dans les différents sens) et le nombre de boucles qui éxistent dans un caractère, ainsi on peut classer notre caractère.Cette classe est calculée de la façon suivante :

$$Classe=H+4\times L+16\times R+64\times U+256\times B \qquad (III - 1)$$

Avec : H : nombre de boucles (Hole).
L : nombre de concavités vers la gauche (Left).
R : nombre de concavités vers la droite(Right).
U : nombre de concavités vers le haut (Up).
B : nombre de concavités vers le bas (Below).[2]

Exemple:

Caractère	H	L	R	U	B	Classe
ر د ز	0	1	0	0	0	4
هـ	2	0	0	0	0	2
س ش	0	0	0	3	0	192
2 5 ج ح خ	0	1	1	0	0	20

NOTE :

Pour les caractères arabes, les concavités vers le bas sont en général introuvables, si elles éxistent elles sont faibles en surface pour cela on n'a pas pris en compte ce type de concavités.Mais il est trés interéssant de les garder pour des systèmes OCR bilingues (ARABE - LATIN).[3]

III-3-2 DEUXIEME NIVEAU :

C'est a ce niveau que se fait l'interprétation et l'identification. Le premier niveau nous permet la classification des caractères, chaque classe pouvant contenir un ou plusieurs caractères. Pour reconnaître un caractère il faut le comparer avec tous les caractères de sa classe et ceci en se basant sur les caractéristiques secondaires qui sont purement heuristiques.

Les primitives secondaires utilisées dans notre système de reconnaissance sont les suivantes :

1) - La forme du caractère :un caractère peut être carré, allongé , ou bien debout selon le rapport (Largeur/Hauteur) du caractère.

- Un caractère carré comme : ر – ذ – د
- Un caractère allongé comme : ب – ف – ت
- Un caractère debout comme : أ – ل – ك – ع

On définit une variable forme comme suit :

- si le caractère est allongé:

Forme=1

- si le caractère est carré:

Forme=2

- si le caractère est debout:

Forme=3

Voir figure (III-1).

2) - Le taux de remplissage des quatres coins du plus petit rectangle qui contient le caractère (rectangle détécté aprés cadrage du caractère): Cette caractéristique nous donne une idée sur la répartition du caractère aux quatre coins (haut-droit, haut-gauche, bas-droit, bas-gauche).

On définit une variable CorVar qui est évaluée de la façon suivante :

Si le coin(i)est rempli avec(i=1..4)
alors

$$CorVar=CorVar+2^{(i-1)}.$$

Voir figure (III-2).

3) - L'éxistance des points (y compris la Hamza) : Cette variable peut prendre soit la valeur (1), pour les caractères ponctués c'est à dire pour l'éxistance des points, et la valeur (0) pour le cas contraire, c'est à dire l'abscence des points.

On a appelé cette variable **ExPoint** .

Voir figure (III-3).

4) - La position des points (y compris la Hamza) : on a défini une variable appelée PosPoint qui prend les valeurs suivantes:

a - La valeur 1 pour la position des points en haut du caractère.

PosPoint= 1

b- La valeur 2 pour la position des points au millieu du caractère.

PosPoint=2

c- La valeur 3 pour la position des points en bas du caractère.

PosPoint=3

Voir figure (III-4)

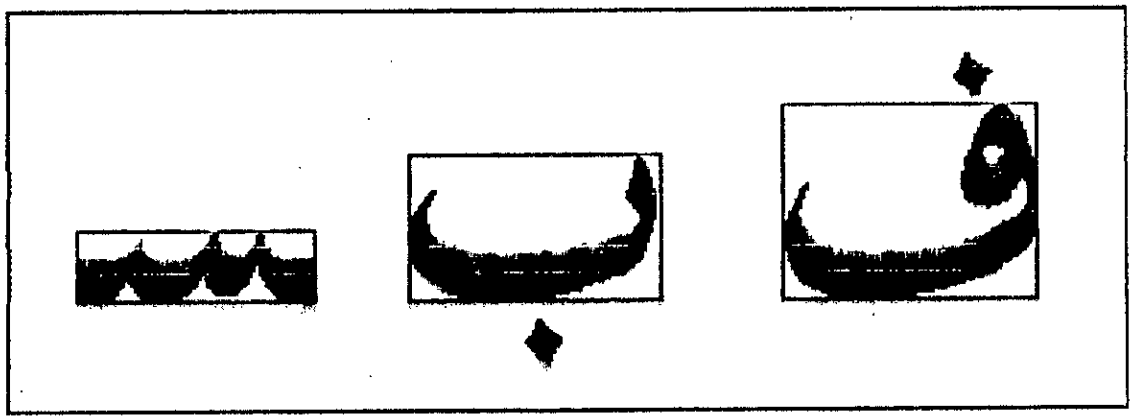


Figure (III-1) a - Caractères allongés

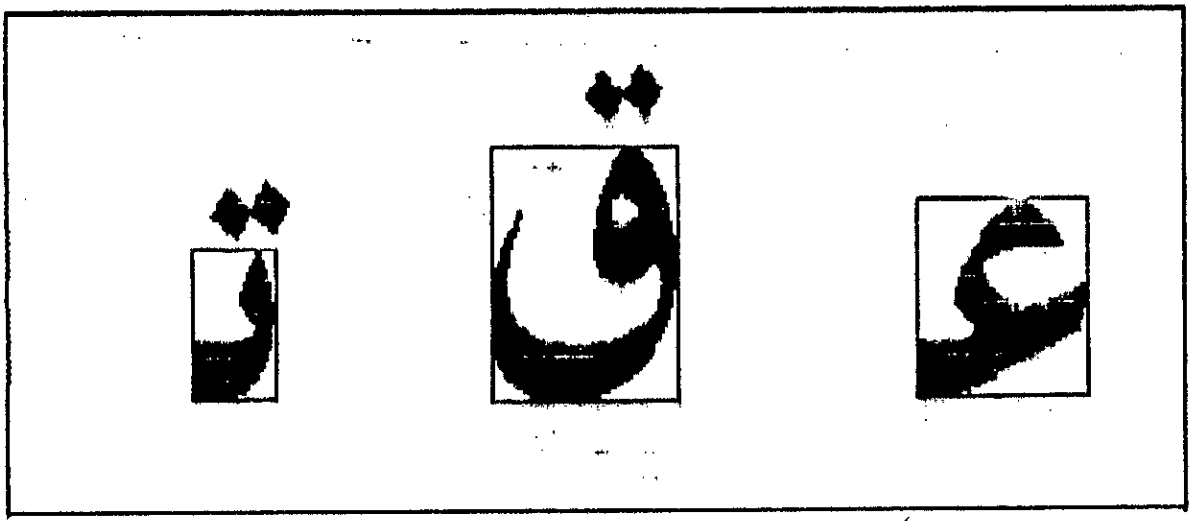


Figure (III-1) b - Caractères carrés

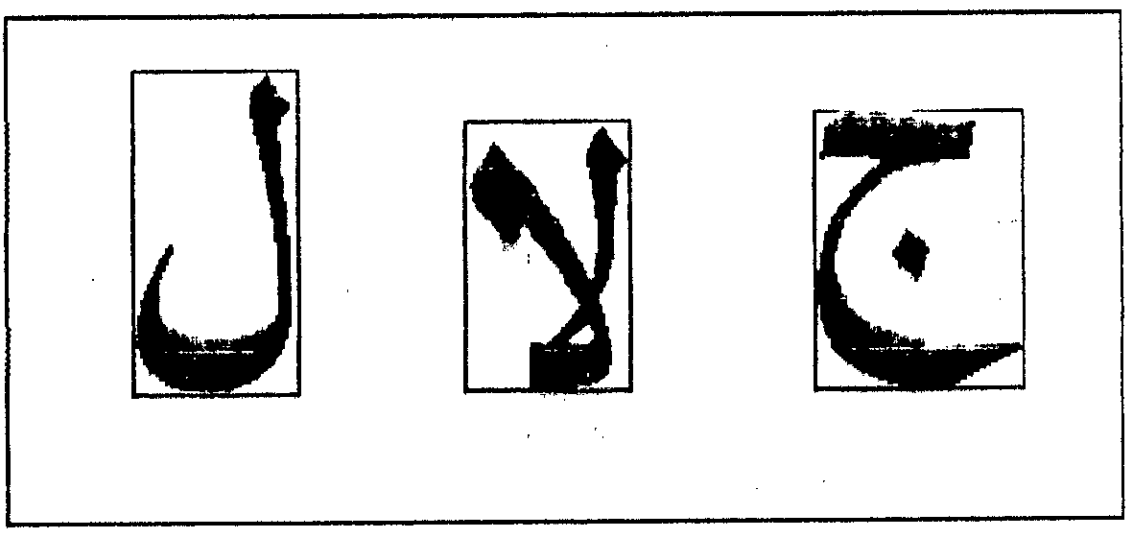


Figure (III-1) c - Caractères debouts

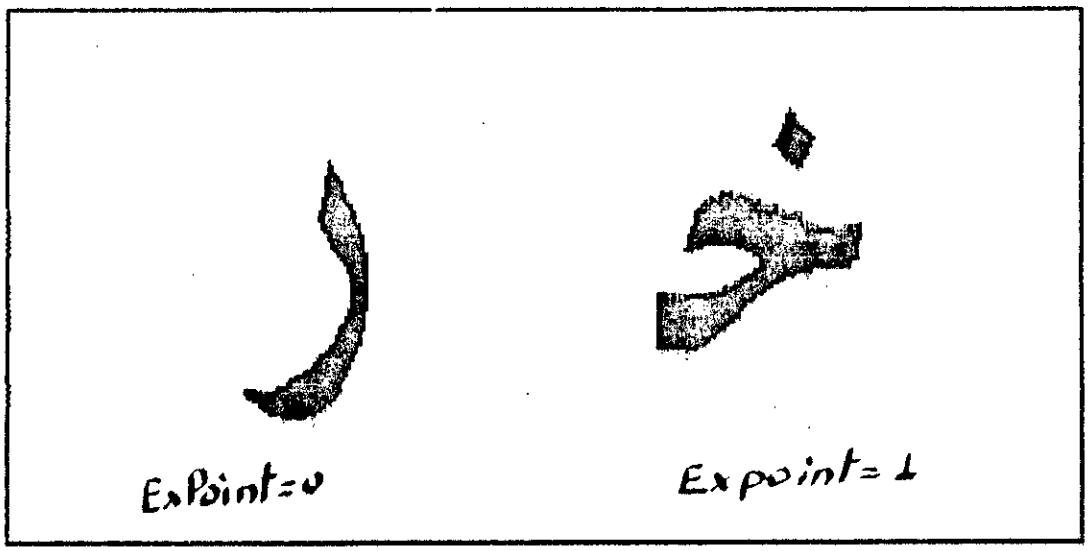


Figure (III-3) - ExPoint : 3ème caractéristique

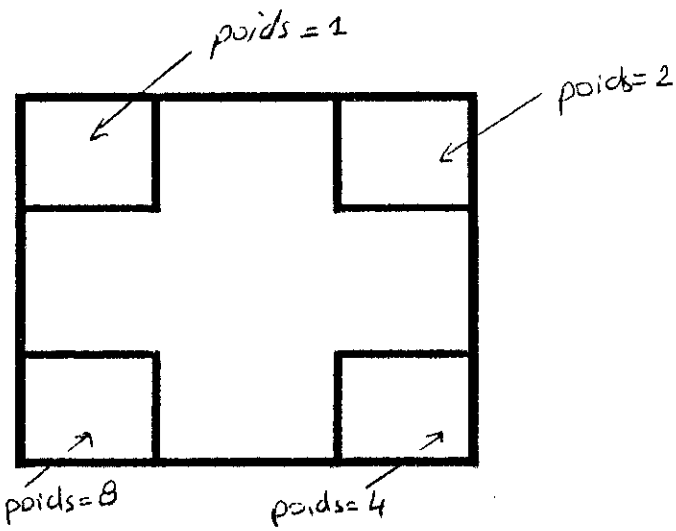


Figure (III-2) - CorVar : 2ème caractéristique

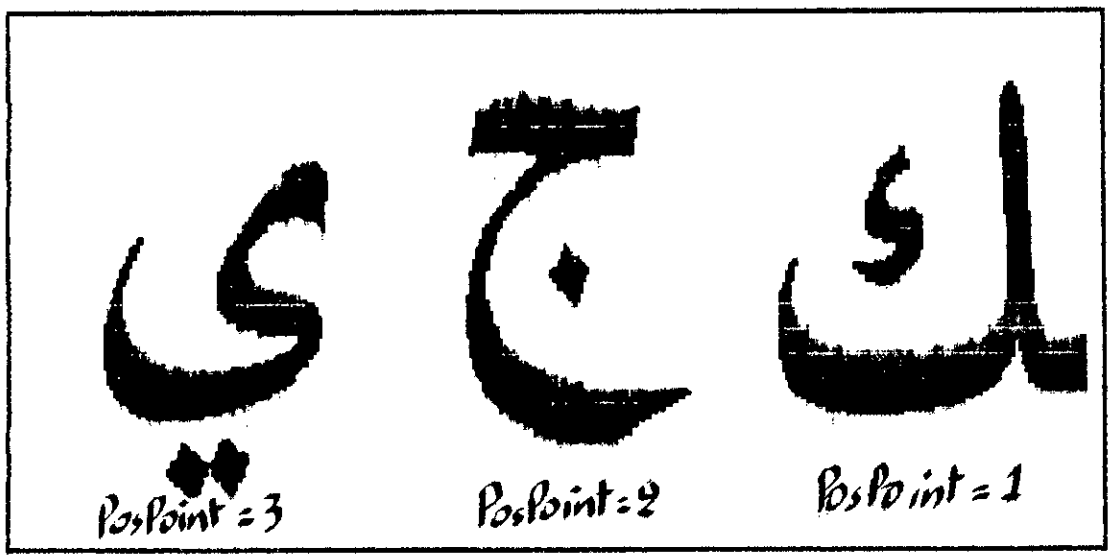


Figure (III-4) PosPoint : 4ème caractéristique

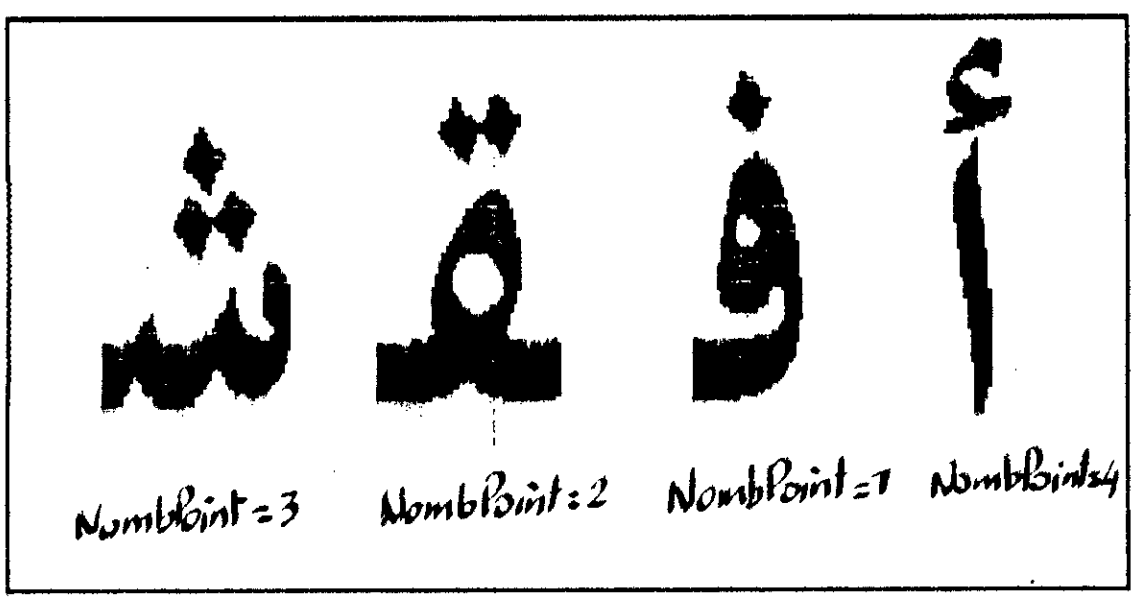


Figure (III-5) NombPoint : 5ème caractéristique

5) - Le nombre de points : On définit une variable appelée NombPoint qui prend les valeurs suivantes :

a- La valeur 1 pour un point.

NombPoint=1

b- La valeur 2 pour deux points.

NombPoint=2

c- La valeur 3 pour trois points.

NombPoint=3

d- La valeur 4 pour la Hamza.

NombPoint=4

Voir figure (III-5)

Note : si la 3ème caractéristique secondaire est nulle (ExPoint = 0) alors les 4ème et 5ème caractéristiques sont obligatoirement nulles c'est a dire (PosPoint = 0, NombPoint = 0).

Avec ces caractéristiques on définit un vecteur d'attributs qui caractérise chaque prototype de la même classe. Voir le schèma suivant :

C1	C2	C3	C4	C5	C6	C7	C8

Avec : Ci le ième élément du vecteur d'attribut.

III - 4 L'EXTRACTION DES PRIMITIVES

Le procédé de reconnaissance (idem pour celui de l'apprentissage) se fait suivant deux niveaux:

1) - La classification.
2) - L'identification .

Pour la classification on utilise les primitives principales(ou primaires) qui sont les concavités dans les différents sens et les boucles .

Pour l'identification on utilise les primitives secondaires qu'on citera aprés.

III-4-1 PRIMITIVES PRINCIPALES:

Ce sont les primitives qui nous permettent de regrouper les caractères en familles ou plutôt en classes, donc leurs choix est important . On remarque facilemement que les caractères arabes sont formés d'un agencement de concavités et de boucles (sauf quelques cas particuliers voir figure (III-6) ; donc on éstime que ce choix est trés convenable.

Une concavité est une partie d'un caractère qui est courbée vers l'intérieur (ressemble a une vallée),elle peut être dirigée dans plusieurs directions (soit vers la droite, la gauche, le bas ou le haut), figure(III-7).

DETECTION DES PRIMITIVES PRINCIPALES :

Comme on l'a déja noté, la difficulté du choix des primitives est de même grandeur que leur extraction,donc il faut utiliser une méthode trés efficace pour pouvoir trouver la classe du caractère.

PRINCIPE DE LA METHODE

Aprés avoir cadré le caractère (trouver le plus petit rectangle qui le contient), on fait le balayage de toute la surface, contenant le caractère, horizontalement et verticalement.Pendant ce balayage on mémorise les coordonnées des lignes (respectivement des colonnes) qui présentent quatre transitions au moins ; (une transition que ce soit 0 - 1 ou 1 - 0) parceque l'existence de 4 transitions ou plus signifie qu'on a croisé le caractère 2 fois ou plus, donc il y a une possibilité d'éxistance d'une concavité ou d'une boucle voir figure (III-8).

Si le caractère est bien cadré alors le nombre de transitions de chaque ligne (respectivement colonne) est pair ,l'indice des transitions (0 - 1) est impair et l'indice des transitions (1 - 0) pair. On prend chaque ligne (respectivement colonne) mémorisée suite aux balayages et on cherche le centre des segments [T2n, T2n+1], voir figure (III-9).Tel que Ti représente la ième transition.

A la fin de cette étape on obtient un ensemble de points (appelés points éssentiels) qui seront a la base de la détection des concavités et des boucles.

Le principe est le suivant :

A partir de chaque point on essaye de tirer une droite dans les huit sens : si on croise le caractère dans un sens, on peut conclure qu'il n'existe pas de concavité dans ce sens, et ainsi de suite.

Avec les huit sens (Haut, Haut-droit, Haut-gauche, Droit, Gauche, Bas, Bas-droit, Bas - gauche), on construit une fonction booléenne liée a ce point et on décide a quel type de concavité appartient ce point .

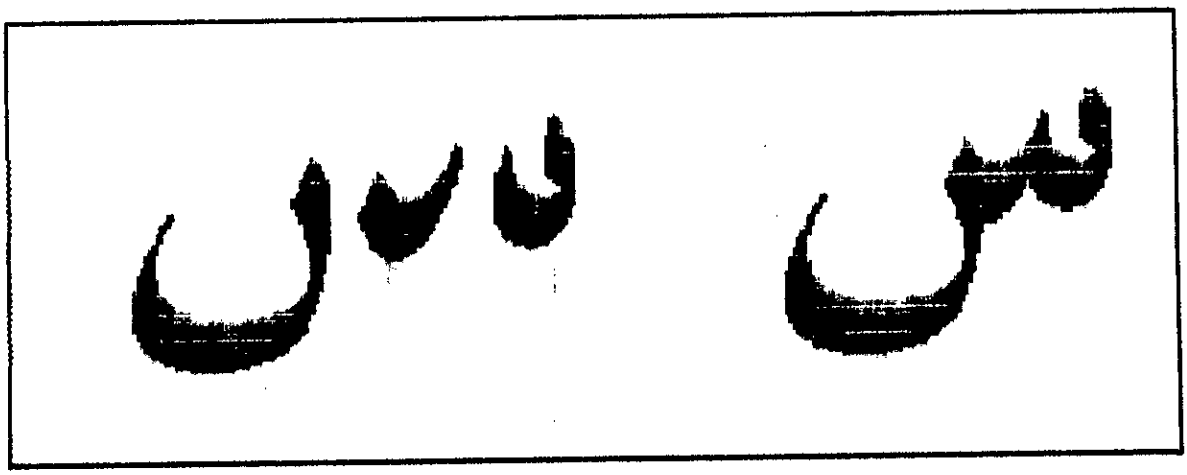


Figure (III-6)a

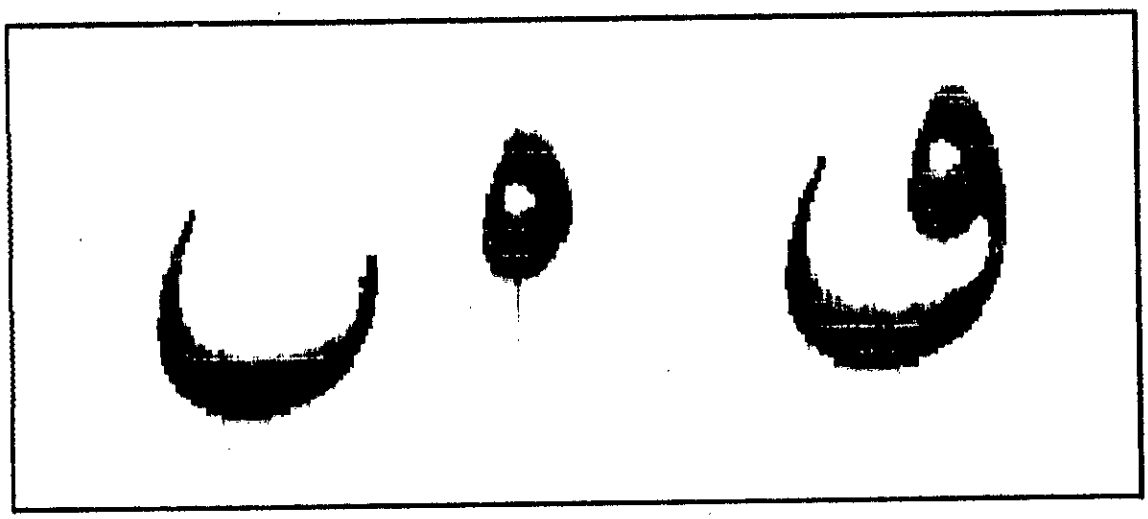


Figure (III-6)b

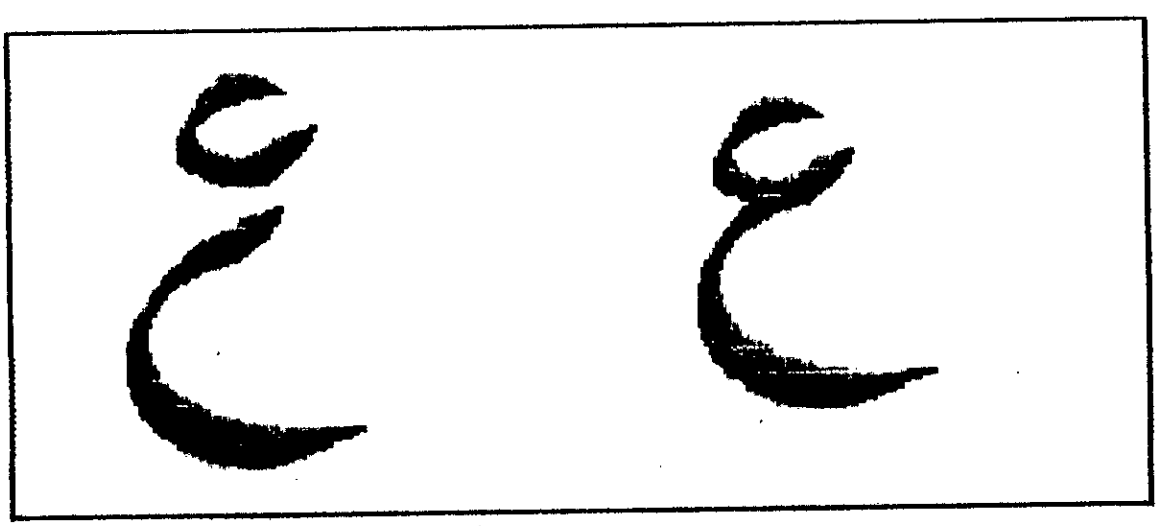


Figure (III-6)c

Caractères arabes : agencement de concavités et de boucles.

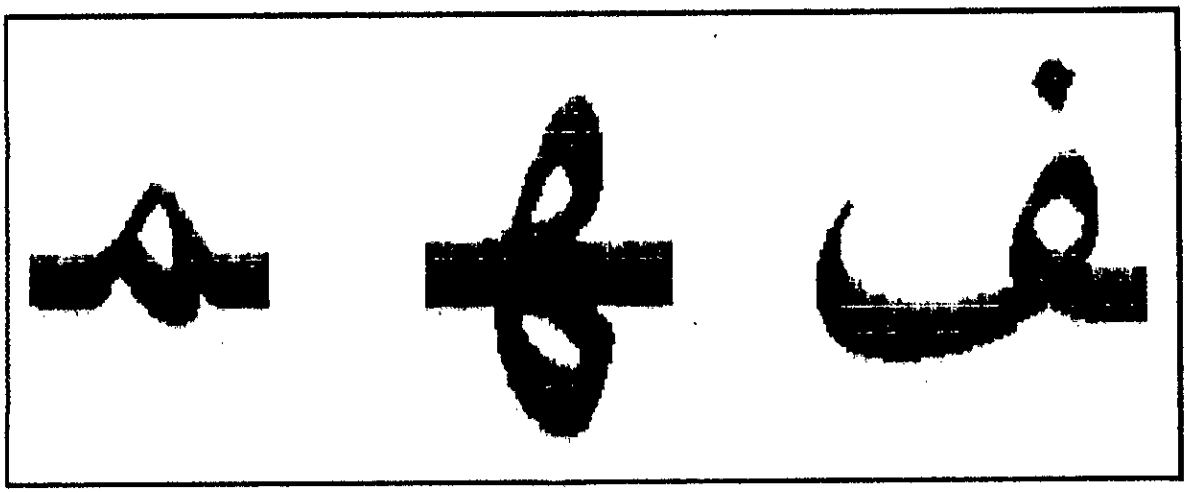


Figure (III-7) Boucles

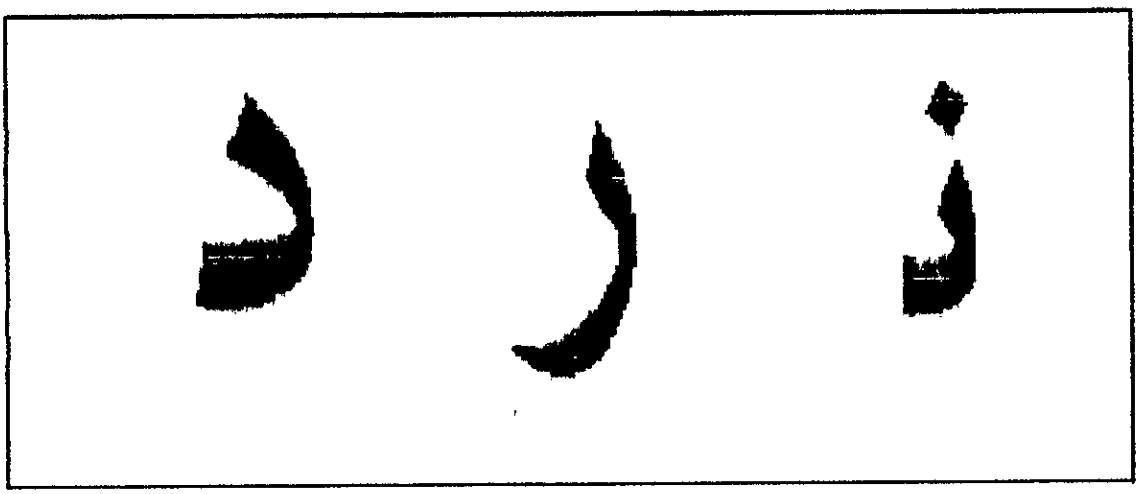


Figure (III-7) concavités vers la gauche

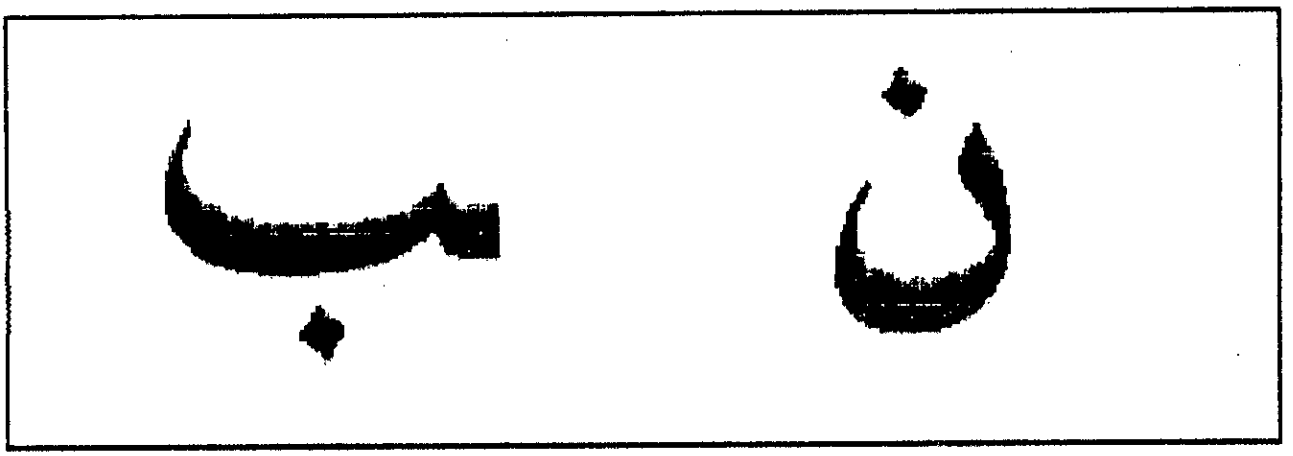


figure II-7 concavité vers le haut

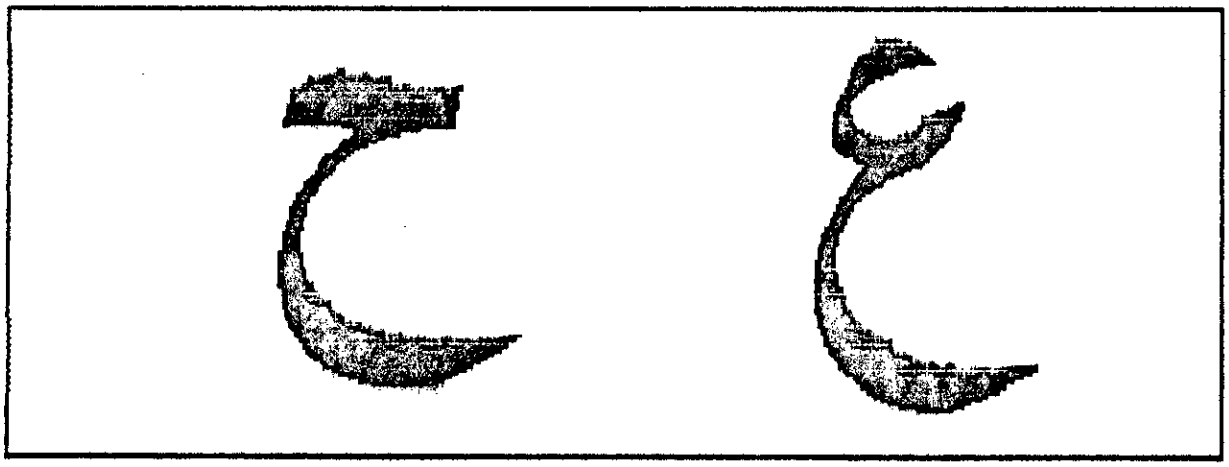


Figure (III-7) concavité vers la droite et vers la gauche.

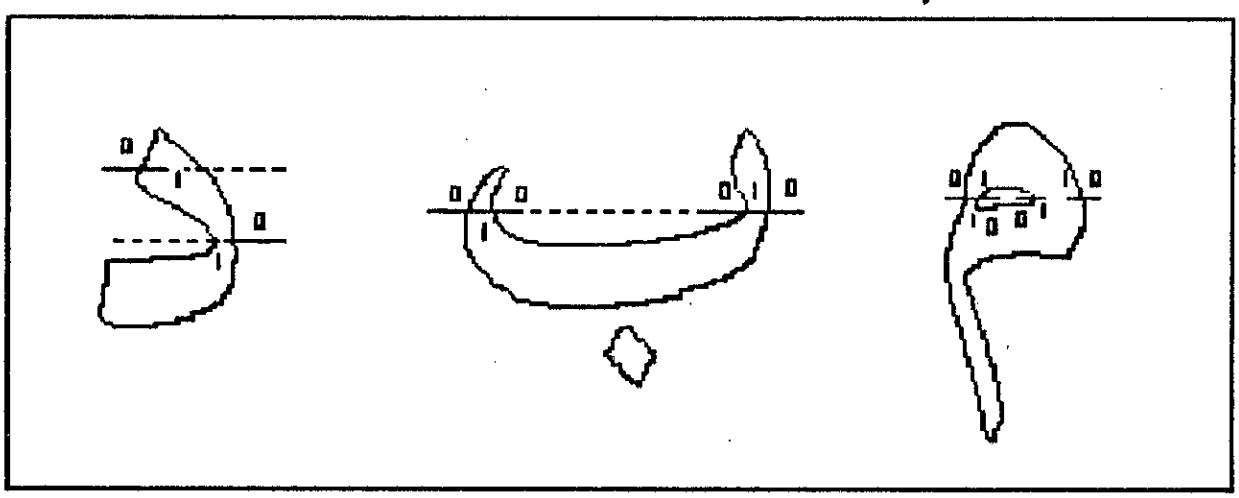


Figure (III-8). Transitions.

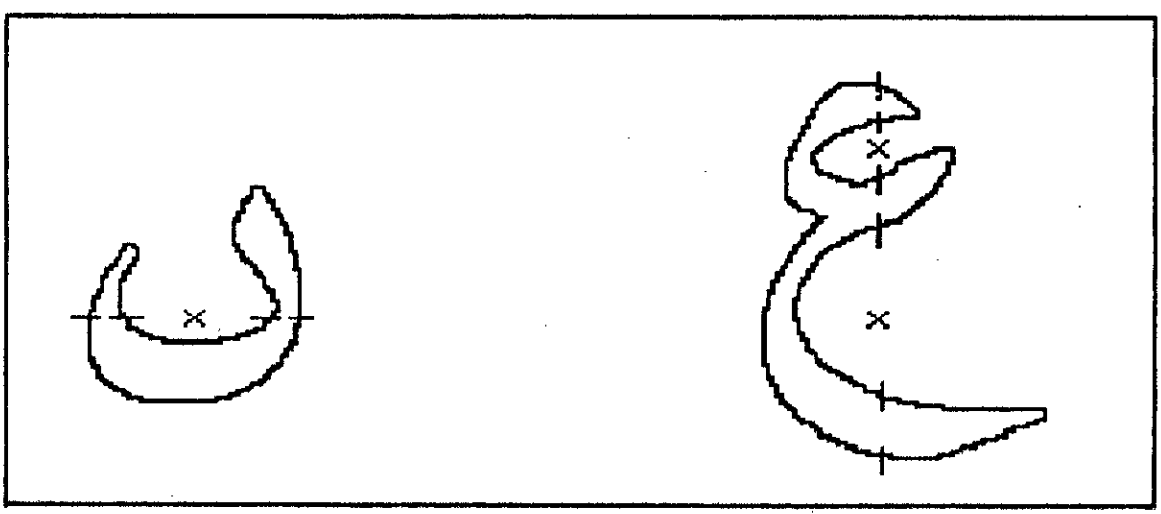


Figure (III-9) points essentiels.

Par exemple :

1- Si on croise le caractère dans tous les sens alors le point consideré appartient a une boucle .voir figure (III-10).

2- si on croise le caractère dans les sens : bas, bas - droit, bas - gauche alors on est en présence d'une concavité vers le haut voir figure (III-11).

Cette étape va nous donner le nombre de concavités dans les 4 sens et celui des boucles, mais il y a des points qui peuvent appartenir a une même concavité ou a une même boucle. Pour cela on a prévu une troisième étape qui permet de nous donner le nombre exact des concavités et des boucles.

Dans cette étape on essaye de faire une liaison [6] entre chaque deux points essentiels, s'ils sont liés (abscence de tout point allumé _ appartenant au caractère _ entre ces deux points) alors ils appartiennent a la même concavité, et si l'und'eux représente une boucle et l'autre représente une concavité quelconque alors la boucle est liée avec la concavité donc il y a une ouverture entre eux ,on conclut que la boucle détéctée est une fausse boucle.(voir figure (III-12).

Pour cela on relie d'abord les boucles entre elles, ensuite les concavités du même type puis les concavités de types différents entre elles et enfin les boucles avec chaque type de concavité.

A la fin de cette étape on aura le nombre réel de boucles et celui de chaque type de concavité, voir figure (III-13).

Cette méthode a donné de bons résultats et s'est avérée efficace, néanmoins elle donne parfois un résultat érroné, et cela est dû non pas a la méthode mais a la forme du caractère; on trouve soit :

1- De fausses boucles qui sont difficiles a éliminer.
2- Des concavités qui ne sont pas bien séparées .
3- La détéction de trés faibles concavités.

On a remarqué que cet algorithme est sensible au bruit, pour cela on a prévu une procédure de bouchage de trous dans le caractère et l'élimination de points isolés, ce sont les opérations d'érrosion et de dilatation qu'on prévoit en général dans l'étape de prétraitement.[6].

II-4-2 PRIMTIVES SECONDAIRES :

L'extraction des caractéristiques secondaires est trés importante parcequ'elles servent a séparer les caractères de la même classe.

On a choisi cinq caractéristiques secondaires (citées précédemment), nous expliquerons la méthode d'extraction de chaque caractéristique dans ce qui suit :

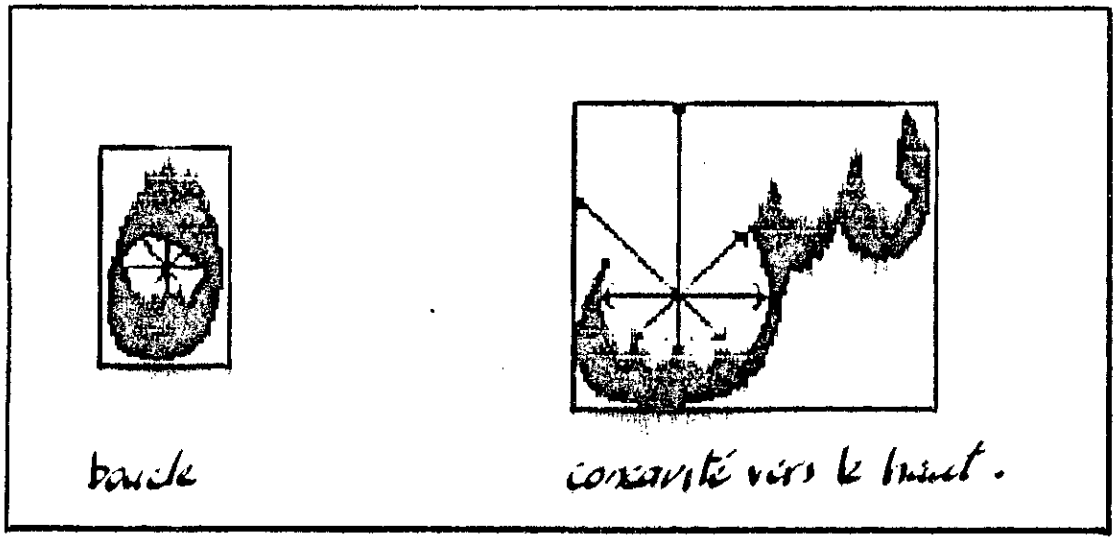


Figure (III-11) : Détection de concavités

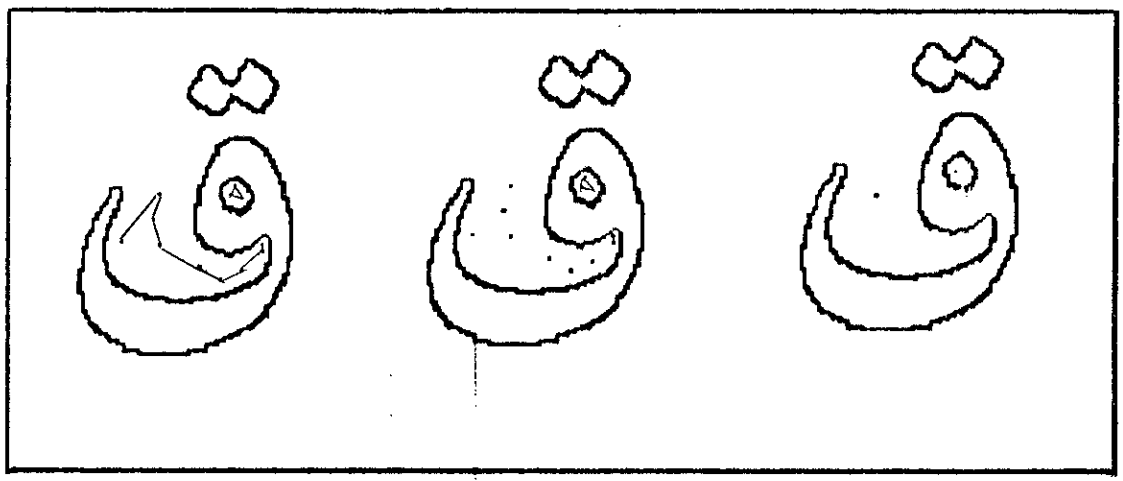


Figure (III-12) Liaison entre points essentiels.

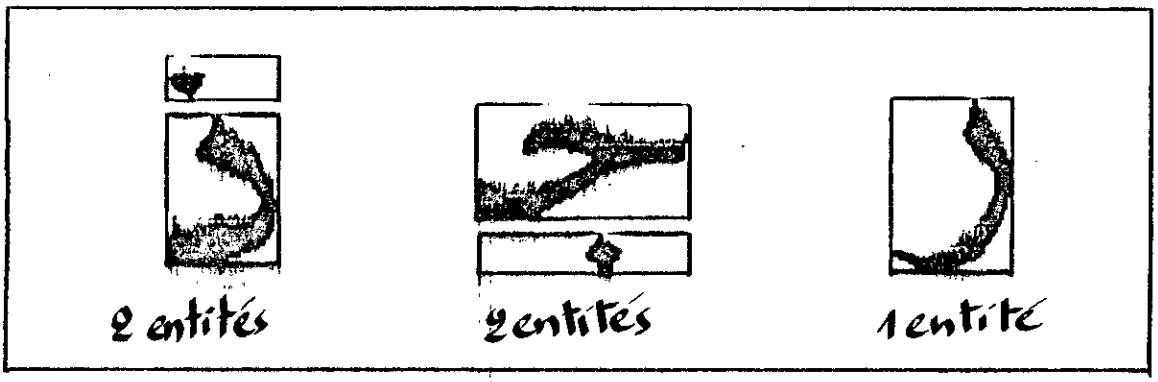
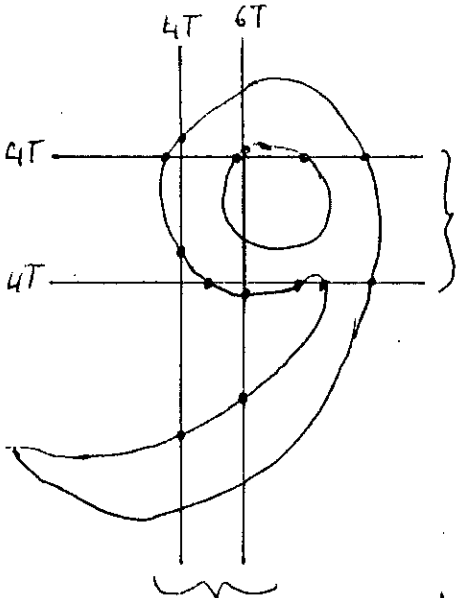
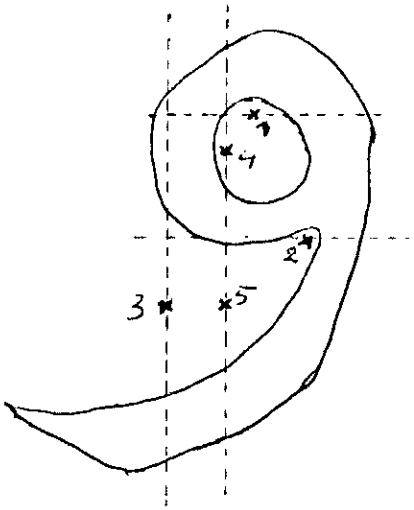


Figure IV-14. Détection d'entités.



T: transition (0-1 ou 1-0)

(1)

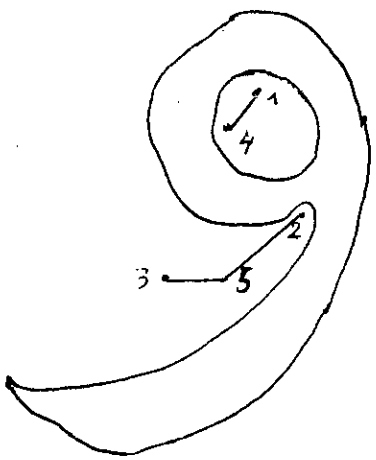


(2)

5 points essentiels.

1, 4 : boucle (eboucles)

2, 3, 5 : Concavité Gauche
(3 concavité gauches)



Le point (1) est relié avec le point (4)
⇒ (1) et (4) appartiennent à la même boucle
Le points (2), (3), (5) sont liés entre eux
⇒ ils appartiennent à la même concavité.
les points (1) et (4) ne sont pas reliés
avec les points (2), (3), (5) donc la boucle
qu'ils la représente est une vraie boucle.

Exemple Complet de détection de Concavités

II-4-2-1 -Extraction de la 1ére caractéristique :

La forme du caractère :

pour extraire cette caractéristique on mesure la largeur (W) et la hauteur (H) du caractère ensuite on fait le rapport suivant :

$$Rapport1 = W/H$$

- Si Rapport1 > Seuil2 , alors on dit que le caractère a une forme horizontale ou bien il est allongé,on attribue a la variable Forme la valeur (1) et on aura :

$$Forme = 1$$

- Si Seuil1 < Rapport1 < Seuil2 , alors on dit que le caractère a une forme carrée, et on aura:

$$Forme = 2$$

- Si Rapport1 < Seuil1 , alors on dit que le caractère a une forme verticale ou bien il est debout et on aura :

$$Forme = 3$$

Les seuils sont estimés aprés plusieurs expériences a :

$$seuil1 = 0.9$$
$$seuil2 = 1.1$$

II-4-2-2 -Extraction de la 2ème caractéristique:

Taux de remplissage des quatre coins:

La variable à calculer est appelée **CorVar**, on a W et H la largeur et la hauteur respectives du caractère,on considère quatre carrés de dimensions :

a- (W/4)×(W/4) si le caractère est debout.
b- (H/4)×(H/4) si le caractère est allongé ou bien carré.

Ces carrés sont pris dans chaque coin du rectangle qui contient le caractère (ce rectangle est trouvé aprés cadrage du caractère).

On calcule la surface de chaque carré notée S.

On calcule la surface pleine de chaque carré notée Sp.

On fait le rapport suivant:

$$Rapport2 = Sp / S$$

Ce rapport (**Rapport2**) représente le taux de remplissage du carré considéré.

Si Rapport2 >= seuil3 , on dit que le coin considéré est plein, et on attribue a une variable booléenne appelée Taux la valeur(1), on aura :

Taux=1.

Sinon il est vide et on aura

Taux=0.

Aprés plusieurs tests on a pris :

Seuil3 = 0.5.

On fait ceci pour les quatre coins ,alors on aura les quatre Taux suivants :
Taux (haut-gauche), Taux (haut-droit), Taux (bas-gauche), Taux(bas-droit).

On calcule ensuite la variable appelée **CorVar** de la façon suivante :
CorVar = Taux (Haut - Gauche)+2×Taux (haut - droite) +4×Taux(bas-droit)
+8 × Taux (bas-gauche).

Cette variable CorVar sera comprise entre (0 et 15), c'est a dire :

0 ≤ CorVar ≤ 15.

Voir figures(III - 13).

III-4-2-3 Extraction de la troisième caractéristique:

L'éxistance des points ou la Hamza :

Cette caractéristique comme les autres caractéristiques est trés importante dans l'identification des caractères, mais dans certains cas elle est trés difficile a extraire .On procède en deux étapes, si dans la 1ère étape on détecte les points ou la Hamza, alors on ne fait pas la 2ème étape sinon on la fait .

Dans la 1ère partie on fait un balayage horizontal de gauche a droite et de haut en bas sur toute la surface du rectangle qui contient le caractére.

Dans la 1ére ligne de balayage on doit rencontrer au moins un point allumé du caractère(un point qui appartient au caractère),on continue le balayage jusqua la fin du caractère.Si on rencontre une ligne où aucun pixel n'est allumé alors on est en présence d'un vide, en continuant le balayage on rencontre le caractère; on peut alors conclure que le caractère testé est composé de deux entités.

On peut rencontrer un vide a nouveau en continuant le balayage donc le nombre d'entités devient trois;mais en général , un caractère arabe ne peut contenir plus de deux entités.

On a :

nombre d'entités = nombre de vide +1

Si aprés balayage de tout le caractère on ne rencontre aucun vide (ou aucune ligne éteinte), on ne peut rien conclure sur le nombre d'entités du caractère car il se peut que le caractère recouvre les points (c'est a dire que le point se trouve a l'intérieur du caractère ou bien le corps du caractère et les points ne sont pas bien séparés comme dans le cas du DJIM (ج)ou le KEF (ك) isolé.

Si le nombre d'entités détecté aprés la première étape est égal a (1) alors on ne peut rien conclure quant a l'existance des points .On passe alors a la deuxième étape .Dans cette étape on détecte l'existance ou la non existance des points a l'intérieur du caractère.

Pour cela on balaye toute la surface du caractère par un carré, si la périphérie du carré est éteinte et il existe au moins un pixel allumé dans ce dernier alors on peut dire qu'il éxiste une entité séparée du corps du caractère d'où le nombre d'entités est égal a deux et donc il existe des points dans ce caractère (ou bien une Hamza), sinon le caractère n'est pas ponctué.

On définit une variable appelée **ExPoint** comme suit :
Si le nombre d'entités est deux :

ExPoint = 1.

Si le nombre d'entités est un :

ExPoint = 0.

Voir figure (III - 14)

II-4-2-4 -Extraction de la quatrième caractéristique :

POSITION DES POINTS (OU LA HAMZA) :

La position des points est facile a détécter par rapport a la caractéristique précédente qui est l'éxistance des points. Si aprés la 1ère étape de détéction des entités on trouve deux entités, on calcule la surface de chacune d'elles, il est clair que l'entité la plus petite en surface représente les points ou la Hamza et l'autre entité représente le corps du caractère.

On définit une varible PosPoint comme suit :

Si les points sont en haut :

PosPoint = 1

Si les points sont au milieu :

PosPoint = 2

Si les points sont en bas :

PosPoint = 3

Si la plus petite entité (en surface) se trouve au dessus de l'autre entité alors les points ou la Hamza se trouve en haut du corps du caractère et on aura :

PosPoint = 1

Dans le cas contraire on aura :

Pospoint = 3.

C'est a dire que les points sont au bas du corps du caractère.

Si la deuxième entité est détéctée aprés la 2ème étape de détéction d'entités alors on tire a partir du milieu de cette entité une droite vers le haut et une autre vers le bas du corps du caractère ,si on croise le caractère dans le sens haut alors les points sont en bas du caractère et on aura :

PosPoint = 3.

Si on croise le caractère dans le sens bas alors les points sont en haut du caractère et on aura :

PosPoint = 1.

Si on croise le caractère dans les deux sens alors les points sont au milieu du caractère et on aura :

PosPoint = 2.

Si le nombre d'entités est égal a 1 aprés les deux étapes de détéction d'entités alors les points n'éxistent pas et on aura :

PosPoint = 0.

voir fig III.14

III-4-2-5 -Extraction de la cinquième caractéristique :

NOMBRE DE POINTS :

On fait ce travail si :

ExPoint= 1.

La variable NombPoint prend les valeurs déja citées(1 , 2 , 3 ou 4).

Cas de deux points :

Les deux points sont faciles a détécter , aprés plusieurs essais on a opté pour les critères suivants :

On définit le rapport suivant:

Rapport3 = Wp / Hp .

Avec Wp,Hp la Largeur et la Hauteur respectives des points.

On définit aussi la variable MaxInt comme suit :

MaxInt :représente le nombre maximal de transitions verticales (0 - 1, ou 1 - 0) de l'entité qui représente les points.

Si (Rapport3 >= 1.1) et (MaxInt = 4)

Alors

NombPoint = 2.

Cas de trois points :

On calcule les surfaces haute et basse de l'entité qui représente les points, notées respectivement : **SurfH, SurfB.**

On calcule aussi la surface qu'occupe les points notée **SurfPl** *(surface pleine)*, on définit aussi la surface Surf qui représente la surface du rectangle (aprés cadrage des points) qui contient les points.

On définit les rapports suivants :

Rapport4 = SurfB / SurfH.

Ce rapport nous donne une idée sur la distribution des points au haut et au bas du rectangle qui les contient.

Rapport5 = SurfPl / Surf .

Ce rapport nous donne le taux de remplissage du rectangle déja défini.

Les trois points ont une distribution de surfaces plus importante en bas qu'en haut du rectangle qui les contient , et le taux de remplissage de ce rectangle est faible, on aura :

Si (Rapport4 >= 1.7) et (Rapport5 <= 0.6)

Alors

NombPoint = 3.

Cas de la Hamza :

La Hamza a une forme carrée presque comme le point,mais ce qui la caractérise c'est le nombre de transitions verticales (0 - 1, 1 - 0) trés élevé ,donc on calcule ce nombre noté NombTr.

Si (0.9 <= Rapport3 <= 1.1) et (NombTr <= 4)

Alors

NombPoint = 4 . (code de la Hamza)

Cas du point :

Le nombre de transitions verticales maximum est égale a deux (2),c'est ce qui différencie le point de la Hamza.

Si (0.9 <= Rapport3 <= 1.1) et (NombTr = 2)

Alors

NombPoint = 1 .

La variable **Rapport3** a été définie précédemment.

Tous les seuils ont été fixés aprés plusieurs essais. voir fig III.15.

III-5- ACQUISITION :

Elle constitue la première étape dans notre système de reconnaissance de formes .L'acquisition des caractères utilisés a été faite par un scanner de type HP-Scann-JET IIP (HPC 1790A), avec une résolution optique de 300 Dpi et des résolutions horizontale et verticale de 78 Dpi ,donc la reconnaissance est de type OffLine.

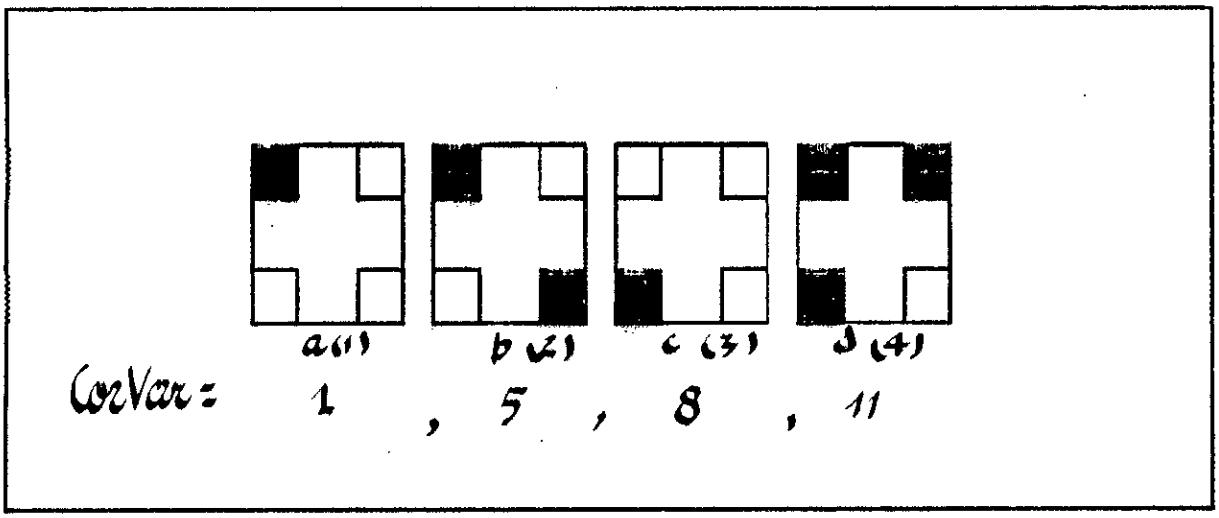


Figure (III-13) Le taux de remplissage.

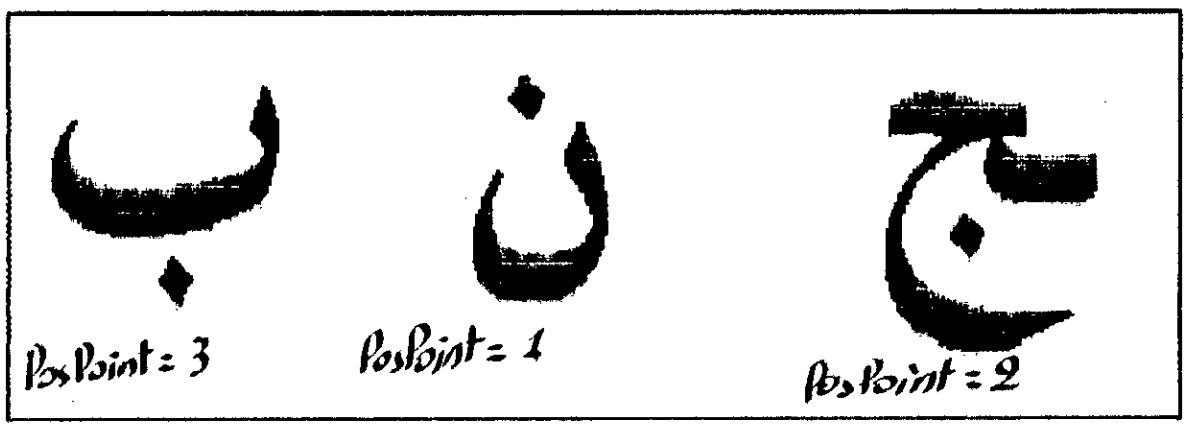


Figure (III-14) Pospoint : Position des points

Figure (III-15) Nombre de points

Les conditions que doit vérifier l'image ou le caractère sont :

a- La résolution doit être supérieure a 100 Dpi.
b- Les règles de connexion des caractères arabes doivent être respectées.
c-Les voyelles courtes (Tachkil : Fatha, Dhamma, Kesra, Soukoune, Chedda) ne sont pas acceptées.

On a opté a ce que le scanner nous donne une image sous forme d'un fichier TIFF malgré qu'il éxiste plusieurs formats d'image comme : BMP, GIF, PIC ect... , et cela pour les avantages qu'il présente malgré sa complexité .

III-5-1 FICHIER TIFF

III-5-1-1 EXPLOITATION DU FICHIER TIFF :[2]

Tout type de fichier informatique a des règles qui gouvernent sa structure, pour pouvoir exploiter ce type de fichier il faut connaitre ces règles .

Le plus simple des fichiers **TIFF** *(Tagged Image File Format)* est constitué de 3 principales parties comme il est précisé dans la figure (III - 15).

a- Première partie : *L'ENTETE DU FICHIER :*

Composée de huit octets ,ils contiennent 3 informations :

* Les deux premiers octets (0 et 1) contiennent le code ASCII du caractère (I) pour INTEL ou du (M) pour **MOTOROLA.**

* Les deux octets suivants (2 et 3) contiennent le numéro dela version (42 pour une compatibilité eventuelle).

* Les derniers quatre octets (4, 5, 6, 7) contiennent un pointeur vers la 2ème partie c'est a dire le 1er **IFD.**

b- La deuxième partie : *L'IFD(Image File Directory):*

C'est le répertoire du fichier,il contient des informations sur l'image, comme:les dimensions, les couleurs, compression ect... .Au début d'un IFD on trouve le nombre de Tags ou champs que contient cet IFD, écrit sur 2 octets; juste aprés ces deux octets commencent les Tags.

Un Tag est un groupe de 12 octets qui représente une information, ces octets sont divisés en 4 parties :

* Les octets (0 , 1) désignent le type de l'information représentée par le Tag (soit la largeur de l'image ou bien sa hauteur ,nombre de bits par pixel,format de compression,échelle de gris ect ...).

Exemple :

Si on trouve la valeur (256) alors ce champs contient la largeur de l'image.
Si on trouve la valeur (258) alors ce champs représente lenombre de bits par pixel.

* Les octets (2 ,3) contiennent l'information sur le type de la donnée que représente le Tag (type du point de vue informatique).

On peut trouver
1 : la donnée est de type Byte (8 bits).
2 : la donnée est en code ASCII.
3 : la donnée est de type Short (16 bits non signés).
4 : la donnée est de type Long (32 bits non signés).
5 : la donnée est de type Rationnel.

* Les octets (4, 5, 6, 7) contiennent le nombre de données, s'il s'agit par exemple de la longeur de l'image ce nombre est égal a 1, si la donnée représente une chaîne de caractères alors le nombre de données sera la longueur de cette chaîne.

* Les derniers octets (8 , 9, 10, 11):contiennent la donnée elle même. Si la donnée ne tient pas sur 4 octets alors ces 4 octets contiennent un pointeur vers la donnée (bien évidemment en dehors de l'IFD).

Un IFD se termine par 4 octets qui contiennent un pointeur vers un deuxième IFD d'une autre image dans le même fichier ; c'est l'avantage du format TIFF : un seul fichier peut contenir plusieurs images.

Il faut noter qu'un IFD peut contenir jusqu'a 45 Tags, ce nombre dépend de la richesse de l'image en informations, l'ordre des Tags n'est pas important.

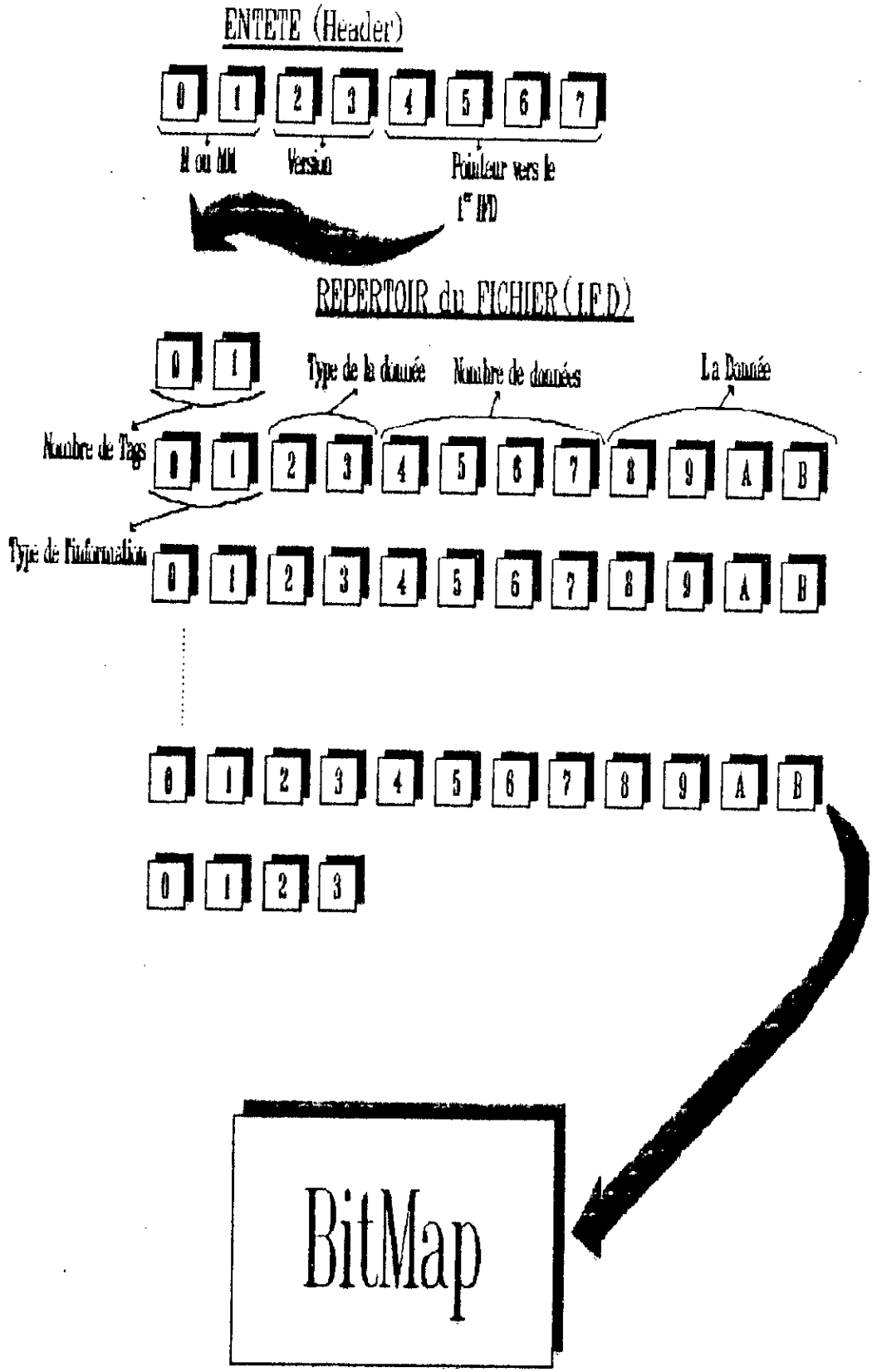
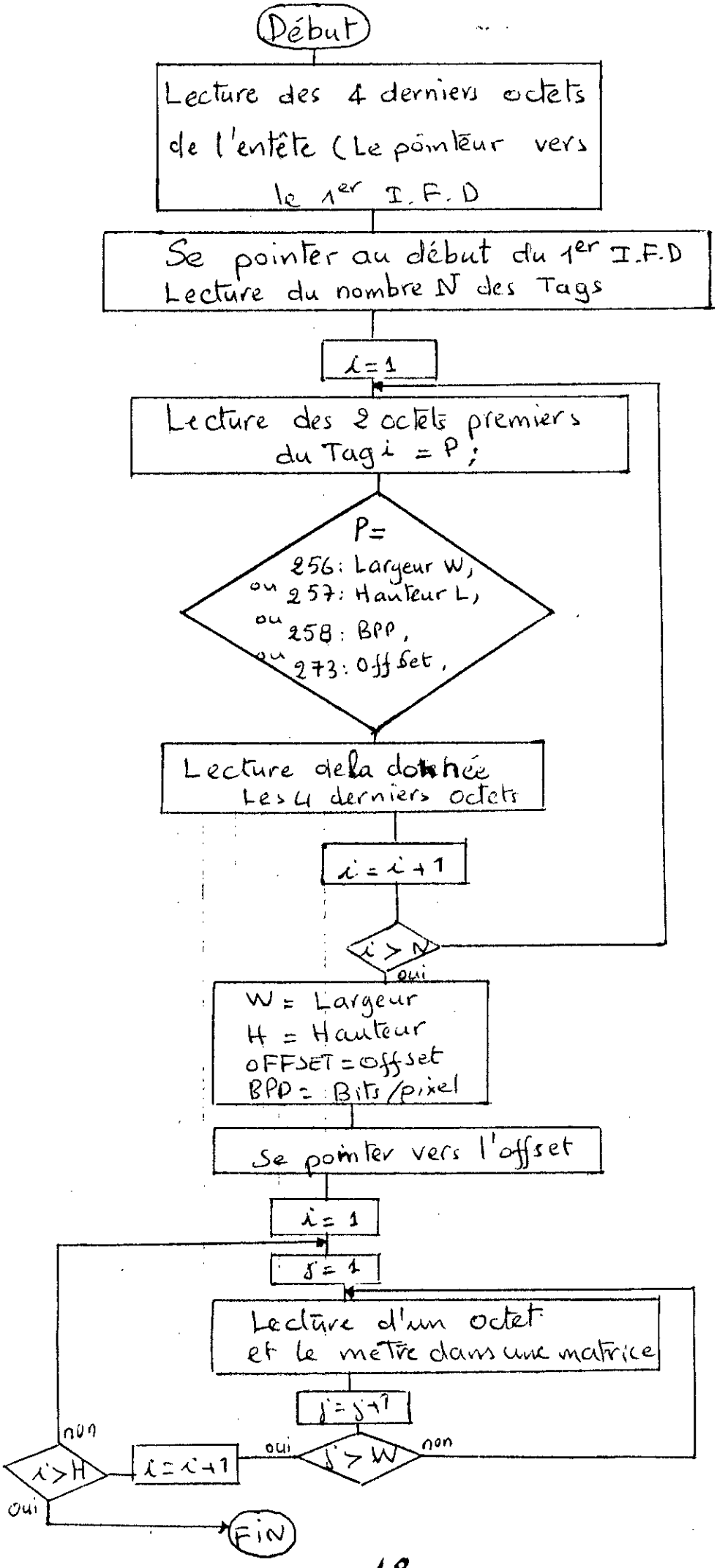


fig III-16- Constitution d'1 fichier TIFF.

- Organigramme de lecture d'un fichier (TIFF).



c- Troisième partie : *Le BitMap :*

C'est l'image elle même, toutes les informations concernant l'image se trouvent dans la partie précédente.

Il reste a noter que chaque octet de cette partie peut représenter (M) pixels avec :

$$M = 8 / Bpp.$$

où Bpp est le nombre de bits par pixel.

Si Bpp = 4 alors M = 2 ,c'est a dire 16 couleurs.
Si Bpp = 8 alors M = 1 ,c'est a dire 256 couleurs.
Si Bpp = 1 alors M = 8 ,c'est a dire 2 couleurs(image binaire).

Les informations les plus importantes pour nous,pour exploiter un fichier TIFF sont les suivantes :

* Les dimensions (Largeur, Hauteur)de l'image.
* Le nombre de bits par pixel.
* L'emplacement du BitMap dans le fichier.

III-6 APPRENTISSAGE :

L'apprentissage qu'on a utilisé est supervisé. On introduit le caractère, qui servira par la suite comme prototype, on extrait ses caractéristiques pricipales et secondaires, on lui donne un nom et on le range dans un dictionnaire.

III-6-1 EXTRACTION DES PRIMITIVES :

L'étape d'extraction des primitives primaires et secondaires fait suivant l'algorithme suivant:

1- On fait la lecture du fichier qui contient le caractère, on aura une matrice IM[W,H] où W est la largeur de l'image, H sa hauteur.

2- On fait le cadrage du caractère.

3- On extait les caractéristiques principales du caractère et on calcule sa classe.

4- On fait un balayage horizontal du caractère.

Si on détécte un vide (qui est réprésenté par une ligne complètement éteinte)

Alors : Existance de deux entités ; **ExPoint = 1.**

Calcul des surfaces des deux entités **SufB** et **SufH**.

Si SurfH > SurfB

Alors points en bas ; **PosPoint = 3.**

Sinon points en haut ; **PosPoint = 1.**

Identification des points ; 1 point : **NombPoint = 1.**
2 points : **NombPoint = 2.**
3 points : **NombPoint = 3.**
Hamza : **NombPoint = 4.**

Sinon : On fait balayage de toute la surface du caractère avec un carré.

Si on trouve une position du carré où son périmètre est complètement éteint et il y a au moins un pixel allumé dedans.

Alors : **ExPoint=1.**

Si a partir des points vers le haut on ne rencontre pas le caractère .

Alors : **PosPoint = 1** *(haut).*

Si a partir des points vers le bas on ne rencontre pas le caractère .

Alors : **PosPoint = 3** *(bas).*
Sinon : PosPoint = 2 (milieu).

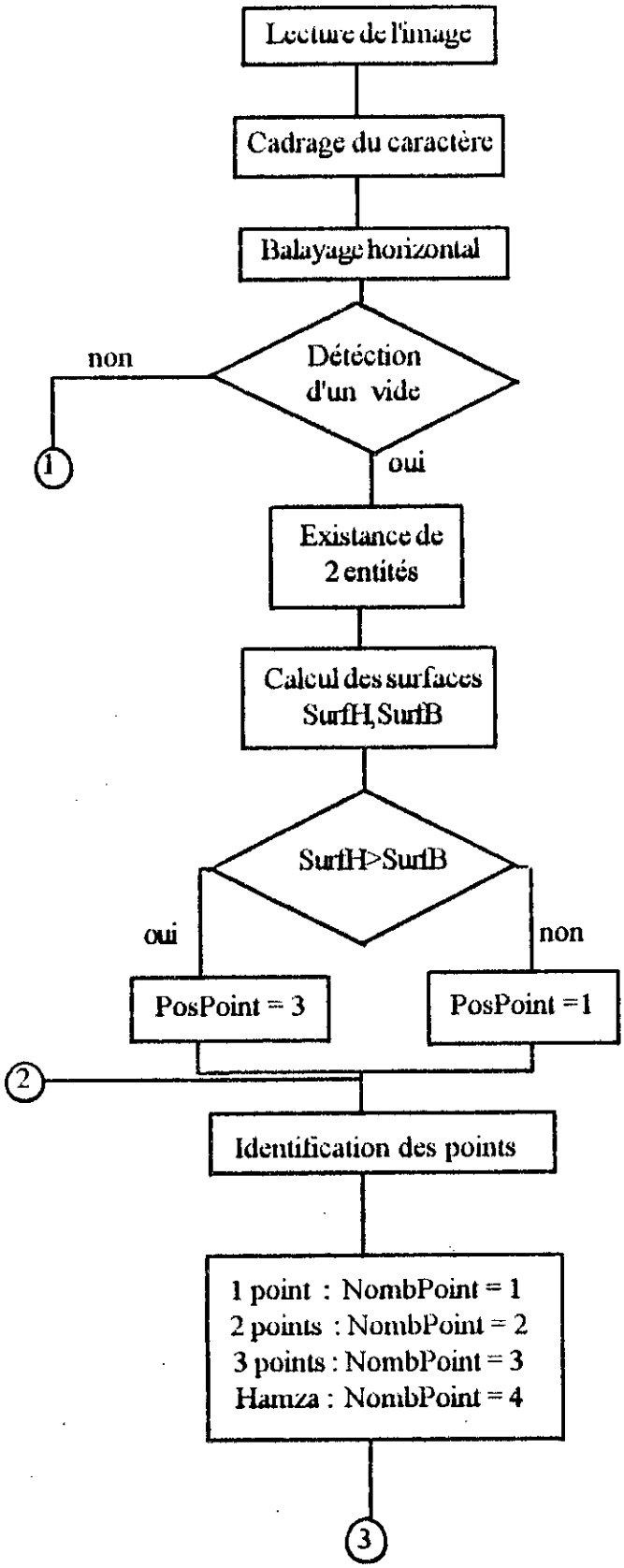
Identification des points ; 1 point : **NombPoint = 1.**
2 points : **NombPoint = 2.**
3 points : **NombPoint = 3.**
Hamza : **NombPoint = 4.**

Sinon : Le caractère ne contient ni points ni Hamza:

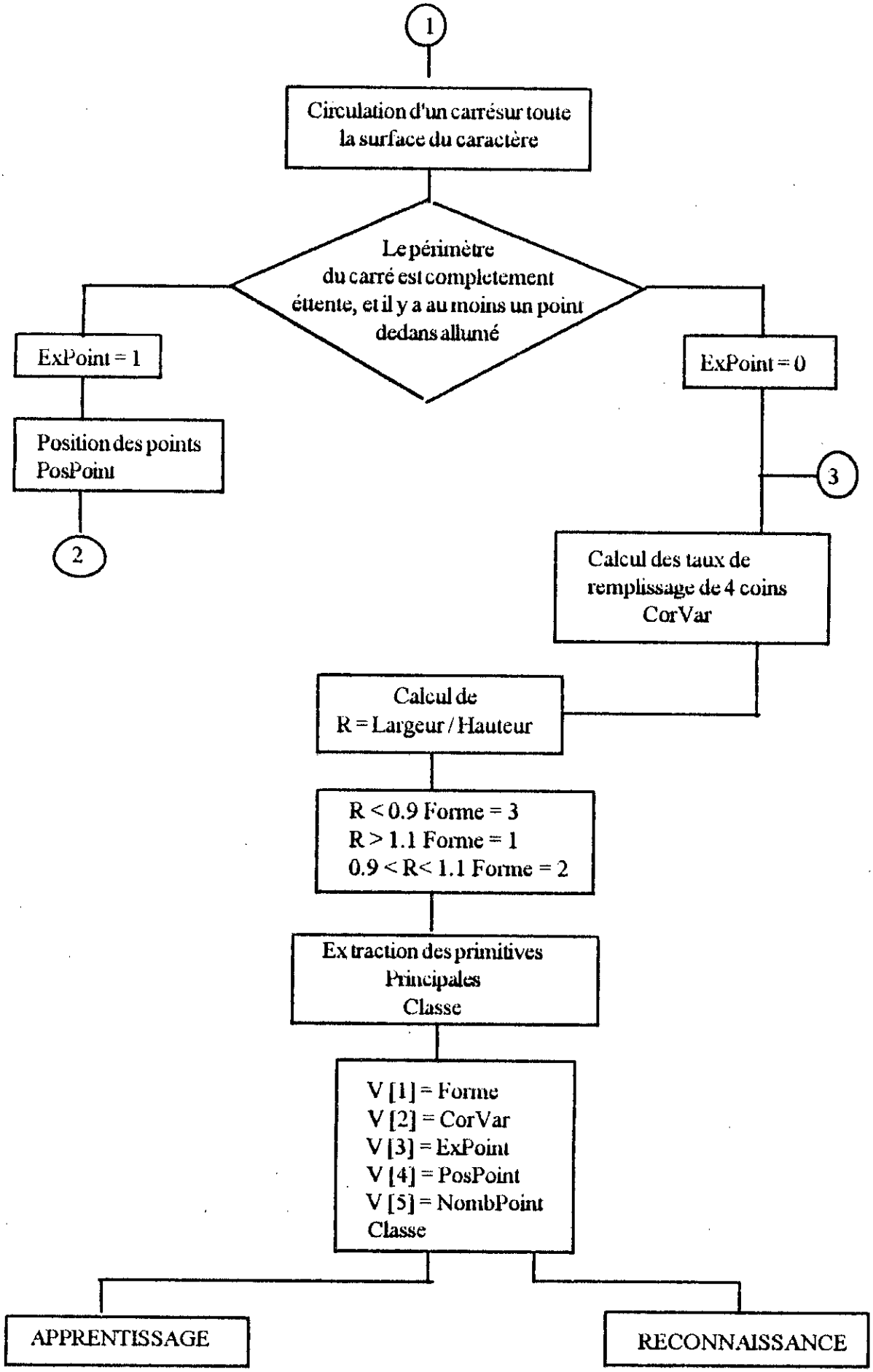
ExPoint = 0.

5- Calcul des taux de remplissage des quatre coins.

CorVar =Coin (haut-droit)+2×Coin (haut-gauche)+8×Coin (bas-droit)
+8×Coin (bas-gauche).



ORGANIGRAMME DE DETECTION DE PRIMTIVES



ORGANIGRAMME DE DETECTION DE PRIMTIVES (Suite)

6- Calcul du rapport **Rapport1=W/H**.

Si Rapport1 > 1.1 alors :Forme = 1 (Caractère allongé).

Si 0.9 < Rapport1 < 1.1 alors : Forme = 2 (Caractère carré).

Si Rapport1 < 0.9 alors :Forme = 3 (Caractère debout).

7- On donne un nom au caractère,on appelle cette variable : **NomCar**.

On obtient a la fin un vecteur V[i] de caractéristiques :

V[1]=Forme.
V[2]=CorVar.
V[3]=ExPoint.
V[4]=PosPoint.
V[5]=NombPoint.
V[6]=NomCar.

Cet algorithme est bien détaillé dans l'organigramme qui suit:

III-6-2 STRUCTURE DU DICTIONNAIRE :

Le dictionnaire est un fichier résidant dans le disque, il contient les caractères prototypes de référence;il est composé de deux parties:

1 / L'ENTETE:

Elle est formée de groupes de 4 octets tel que :

- Le 1er octet contiendra le numéro de classe .

- Le 2ème octet contiendra le nombre de prototypes appelé Npr que contient cette classe.

- Le 3ème et le 4ème octets contiennent respectivement les poids faible et fort du pointeur vers le début de la classe considéréedans le fichier et cela a partir du début du dictionnaire.

2 / LES CLASSES :

Cette partie contient les classes tel que chaque prototype est décrit sur 8 octets, ces octets contiennent, les éléments du vecteur de caractéristiques V :

* Le 1er octet contient la variable Forme.

Forme = {1,2,3.}

* Le 2ème octet contient la variable CorVar.

CorVar={0,1,..,14,15}.

* Le 3ème octet contient la variable ExPoint.

ExPoint={0,1}.

* Le 4ème octet contient la variable PosPoint.

PosPoint={0,1,2,3}.

*Le 5ème octet contient la variable NombPoint.

NombPoint={0,1,2,3,4}.

* Le 6ème octet contient la varible NomCar c'est a dire le nom du caractère.

* Les 7ème et 8ème octets sont réservés.

A partir de la fin de la 1ère partie ***(Entête)*** commence la 2ème partie qui contient les classes, tel qu'il soit réservé pour chaque classe un espace déterminé, le nombre de classes lui aussi est déterminé.

Notre dictionnaire peut contenir au maximum 80 classes, et chaque classe peut contenir au maximum 60 prototypes ; alors on aura :

L'Entête, de taille : $4 \times 80 = 320$ octets.

Le Corps, de taille:8 * 60 * 80 = 38400 octets.

Donc la taille du dictionnaire sera : 38720 octets.

Grâce a cette structure on peut accéder a n'importe quelle classe dans le dictionnaire sans le parcourir entièrement.

III - 7 LA RECONNAISSANCE :

La reconnaissance est la dernière étape dans la chaine d'un système OCR elle représente le bloc décisif .Aprés cette étape nous aurons une de ces trois décisions :

* Le caractère est reconu.

* Le caractère est rejeté.

* Le caractère est ambigu(indécision).

Mais l'étape de reconnaissance est toujours précédée de l'étape d'apprentissage qui nous permet d'avoir un dictionnaire de référence contenant les caractères prototypes.

La reconnaissance est basée sur la notion de distance, qui est calculée entre le caractère inconnu a reconnaître et les caractères prototypes du dictionnaire ayant la même classe que le caractère inconnu .Le caractère inconnu est reconnu comme étant le caractère prototype qui a la distance minimale,c'est a dire qui lui ressemble le plus.

III - 7 - 1 DISTANCES UTILISEES:

Nous n'avons utilisé dans notre système que deux distances **D1** et **D2**.

La distance **D1** est calculée entre les Formes et **D2** est calculée entre les **CorVars** (*les taux de remplissage des quatre coins)*;

C'est a dire qu'on a calculé les distances entre les deux premières caractéristiques secondaires. Les trois dernières caractéristiques secondaires ExPoint, PosPoint, NombPoint) doivent être identiques .

1 /Distance D1 :

La première caractéristique secondaire est la forme du caractère, on peut dire qu'un caractère qui s'écrit allongé peut être trouvé sous la forme carrée mais il est loin d'être trouvé debout. L'inverse est aussi vrai, alors on peut dire que la forme carrée est proche de la forme allongée même chose pour la forme debout mais la forme debout est loin de la forme allongée.

On definit D1 comme suit:

$$D1 = abs (Vx[1] - Vp[i]).$$

avec :

Vx [i] : élément d'ordre (i) du vecteur de caractéristique du caractère x a reconnaître.
Vp [i] : élément d'ordre (i) du vecteur de caractéristiques du caractère prototype p.

Pour qu'on puisse dire que le caractère x ressemble dans la forme au caractère prototype p il faut que :

$$D1 \leq 1 .$$

2 /La distance D2 :La deuxième caractéristique est CorVar.

Dans ce cas on a affaire a une distance binaire. Soit la figure (III - 16).

Si on prend le caractère (2), il peut être trouvé sous la forme des caractères (1) ou (4) (c'est a dire un coin en moins ou en plus respectivement), mais ne peut pas être trouvé sous la forme du caractère (3). De la on calcule la distance D2 comme suit :

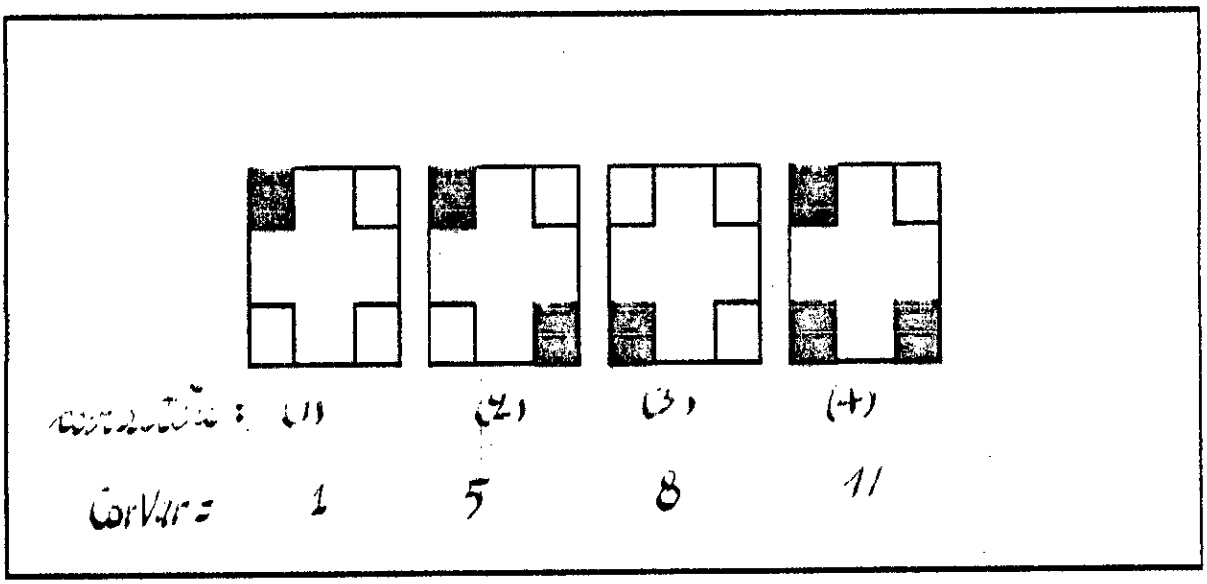


Figure (III-17)

On a :

$$A = (Vx[2]) \text{ XOR } (Vp[2]) .$$

A : Nombre binaire.
D2 = Nombre de (1) que contient A.

Exemple :

D2[(1),(2)]:distance entre le caractère (1) et le caractère (2) dela figure (III - 16)

D2[(1),(2)]= F[(0001) XOR (O1O1)] = 1.

D2[(2),(4)]= F[(0101) XOR (1101)] = 1.

D2[(2),(3)]= F[(0101) XOR (0010)] = 3.

Tel que F est une fonction donnant le nombre de (1) que contient son argument écrit en binnaire.

Pour que le caractère x ressemble au caractère prototype il faut que:

$$D2 \leq 1.$$

III-7-2 ALGORITHME DE RECONNAISSANCE:

Une fois les caractéristiques principales et secondaires du caractère a reconnaître sont extraites ,on commence l'identification suivant l'algorithme de reconnaissance suivant :

1- Les caractéristiques principales nous donnent la classe du caractère, on se place dans le dictionnaire au début de cette classe et on charge tous les vecteurs de caractéristiques de chaque prototypes, on aura :

CAi[j] : jème caractéristique du ième prototype.

Avec j={1, 2, 3, 4, 5};
i=1..Npr ;
Npr : le nombre de prototypes de la classe.

On cherche les prototypes qui ont les mêmes caractéristiques que le caractère inconnu et cela en calculant la distance Dis qui est définie comme suit :

$$Dis[i] = \sum_{j=1}^{5} | CAx[j]-CAi[j] | ;$$

avec : i = 1 .. Npr.
CAx[j]:jème caractéristique du caractère a reconnaitre.
CAi[j]:déja définie.

les prototypes qui vérifient :

$$Dis[i] = 0$$

sont mis dans une liste appelée Liste1, qui contient k proptotypes.

Si k = 1 le caractère est identifié.
Si k > 1

Si tous les prototypes décrivent le même caractère c'est a dire qu'ils ont le même nom.

Alors le caractŠre est identifié.

Sinon le caractère est ambigü.

Si k = 0 ; on met dans une deuxième liste appelée Liste2 les prototypes qui verifient:

$$D = 0.$$

avec :

$$D[i] = \sum_{j=3}^{5} | CAx[j] - CAi[j] | .$$

i=1..k1.
k1 représente le nombre de prototypes de Liste2.

Si k1=0, le caracractère est rejeté.

Si k1=1, on calcule D1 et D2

Si (D1≤1)et(D2≤1) alors le caractère est reconu.

Sinon il est rejeté.

Si k1>1 ,on met dans une liste appelée Liste3 tous les caractères qui vérifient:

$$(D1 \leq 1) \text{ et } (D2 \leq 1) ;$$

On obtient k2 caractères.

SI k2=1 le caractère est reconu .
SI K2=0 le caractère est rejeté .
Si k2>1 on calcule la fréquence d'occurence de chaque caractère [3] ;
Si :il y a un seul prototype qui a une fréquence d'occurence la plus élevée ;
alors : le caractère est reconu.

Sinon il y a ambiguité.

III - 8 - CONCLUSION :

Il faut noter les problèmes rencontrés surtout pour déterminer les seuils , car les formes du caractère diffèrent d'un style a un autre et d'une taille a une autre et quelque fois lorsque le caractère recouvre les points il est difficile de séparer ces entités l'une de l'autre. Les résultats obtenus sont satisfaisants,malgré ces problèmes.

CHAPITRE IV

RESULTATS OBTENUS

CHAPITRE IV

RESULTATS OBTENUS

IV - 1 IMPLEMENTATION:

Notre travail a été écrit en TURBO PASCAL version7.0 sur un PC compatible IBM (microprocesseur 80486 ; 66 MHz d'horloge ; 4 MEGA Octets de RAM).

Pour l'aquisition des caractères on a utilisé un scanner **HP ScanJet IIp Scanner**

IV - 2 DICTIONNAIRE:

On ne peut juger la validité de notre méthode qu'a travers les résultats expérimentaux obtenus sur des données réelles.Il est donc nécessaire de disposer d'un dictionnaire constitué de fontes et de tailles de caractères différents.Une partie des caractères utilisés (59 caractères) a été scannée par **HP ScanJet IIp Scanner** et une autre partie de ces caractères (208 caractères) a été générée par le logiciel **"PaintBrush" de Windows.**

Nous avons considéré deux types de fontes et deux tailles diférentes.
Tous les caractères testés sont différents soit parcequ'ils appartiennent a des fontes différentes soit parcequ'ils sont de tailles différentes.

Le dictionnaire utilisé contient 155 caractères (les caractères appris). voir annexe.

IV - 3 RECONNAISSANCE

Nous avons évalué les statistiques séparement sur deux ensembles :

Les caractères générés et les caractères scannés .

1- Pour les caractères générés nous obtenons pour 208 caractères:

Un taux de reconnaissance global de 100% dont :

a- 94.71 % de reconnaissance pure .
b- 00.96 % d'indécision.
c- 04.32 % de substitution.

2- Pour les caractères scannés nous obtenons pour 59 caractères:

Un taux de reconnaissance global de 100% dont:

a- 84.74 % de reconnaissance pure.
b- 05.08 % d'indécision.
c- 10.16 % de substitution.

Le taux de reconnaissance peut augmenter grâce à la flexibilite du dictionnaire et cela en faisant l'apprentissage d'autres caractères, cette fléxibilite est le résultat de l'utilisation de pointeurs au sein du dictionnaire.

Le temps moyen de reconnaissance d'un caractère est de 30 centième de seconde.Ce temps peut être amélioré si la taille des caractères utilisés est plus faible.

Le temps moyen de reconnaissance d'un caractère de l'ensemble des caractères générés de la taille la plus petite est de 14 centième de seconde ,on voit bien que ce temps est plus faible que le temps moyen global.

L'opérateur d'extraction de concavités qu'on a utilisé détécte les concavités vers le bas mais nous ne les avons pas pris en compte parcequ'elles sont faibles pour les caractères arabes ,si on prend en considération cette concavité(vers le bas) on peut appliquer cet opérateur pour les caractères latins .

On a fait un petit test sur les caractères latins et on a eu des résultats satisfaisants,sur un dictionnaire de 29 lettres (majuscules et minuscules) on a eu deux cas d'ambiguité qui sont le (V et Y), (C et G) donc un taux de reconnaissance de 86.20 %, et le temps moyen de reconnaissance d'un caractère est 25 centième de seconde.

Conclusion Générale

CONCLUSION GENERALE:

Cette étude vise a noter la faisabilité d'un système de reconnaissance de caractères arabes isolés multitaille, multifontes. Pour éffectuer cette reconnaissance on ne dispose d'aucune information sur la position du caractère a l'intérieur du mot.

Nous n'avons pas utilisé un algorithme de squelétisation a cause des inconvenients qu'il présente par exemple il est couteux en temps et ne conserve pas toujours la forme du caractère. Comparé aux techniques de squelétisation notre algorithme de détéction de concavités a le mérite d'être rapide parcequ'il travaille directement au niveau du caracère.

Les résultats obtenus sur 267 caractères testés nous ont permis d'estimer que cet objectif a été atteint. Tout le long de notre tavail nous nous sommes attachés a développer et a exploiter une seule approche d'extraction des primitives:celle qui consiste a détécter les concavités et les boucles.

Par ailleurs nous avons consacré une grande partie de notre travail a l'expérimentation et aux tests de performance de notre opérateur d'extraction de concavités et de boucles et de notre système global de reconnaissance de formes.

En effet en R.F seule l'expérience permet de guider l'imagination vers des solutions possibles d'amélioration et vers des problèmes non résolus. Nous pensons avoir montré la fiabilité de l'opérateur déja cité et la robustesse des primitives secondaires (à savoir Forme, CorVar, ExPoint, PosPoint, NombPoint) qui sont des conditions nécessaires pour la reconnaissance du 2ième niveau (interprétation).

La méthode de reconnaissance qu'on a utilisé est purement topologique.Elle est nécessairement indépendante de la taille des caractères et de toute notion métrique.

BIBLIOGRAPHIE :

[1]: **ABDEL BELAID & YOLAND BELAID,**
" RECONNAISSANCE DES FORMES ,Méthodes et applications " .
Inter Edition 1992.

[2]:**ALFRED POOR,**
"PC MAGAZINE " .Looking at the TIFF spécification from the incide.

[3]:**BASSAM KURDY,**
" MULTIFONT RECOGNITION SYSTEME FOR ARABIC CHARACTER ".

[4] **L.SAADAOUI,**
"TECHNIQUES DE FILTRAGE DES IMAGES". Thèse de Magistère E.N.P

[5]:**MATOUGUI,**
"REALISATION D'UN SYSTEME DE RECONNAISSANCE OPTIQUE DES CARACTERES ARABES " Thèse de magistère .C.D.T.A.

[6]:**MICHEL LUCAS,**
"LA REALISATION DES LOGICIELS GRAPHIQUES INTERACTIFS "
Publié en .1979.

[7]:**KOK-LARY HENG,**
"ANALYSE D'IMAGE DE LIGNES:DES PIXELS AU PRIMITIVES ".
Thèse de doctorat de l'université de PARIS 6
Specialité Informatique -1986.

[8]:**TOUMAZET,**
"TRAITEMENT DE L'IMAGE SUR MICROORDINATEUR".

Caractères Prototypes utilisés.

ـأ	أ	ـئ	ـئ
ـا	ـا	ـإ	إ
ـب	بـ	بـ	بـ
ـت	تـ	تـ	تـ
ـث	ثـ	ثـ	ثـ

جـ	ـجـ	ـج	ج
حـ	ـحـ	ـح	ح
خـ	ـخـ	ـخ	خ
دـ	ـدـ	ـد	د
رـ	ـرـ	ـر	ر

ط ط ط ط

ظ ظ ظ ظ

ك ك ك ك

ل ل ل ل

م م م م

نـ ـنـ ـن ن

صـ ـصـ ـص ص

ضـ ـضـ ـض ض

عـ ـعـ ـع ع

غـ ـغـ ـغ غ

ف	فـ	ـفـ	ـف
ق	قـ	ـقـ	ـق
س	سـ	ـسـ	ـس
ش	شـ	ـشـ	ـش
ه	هـ	ـهـ	ـه

و	ؤ	ـؤ	ؤ
ي	ي	ـيـ	ي
ئ	ئ	ـئـ	ئ
ى	ى		
لا	لأ	ـلأ	لأ
ة	ة		

4 3 2 1

8 7 6 5

0