REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE D'ALGER

D.E.R. DE GENIE ELECTRIQUE ET INFORMATIQUE

DEPARTEMENT D'ELECTRONIQUE

MEMOIRE

En vue de l'obtention du diplôme

d'Ingénieur d'Etat en Electronique

THEME

Mise au point d'un système OCR

Réalisé par :
M. H. KAHOUL
M. M. DAMOU

Proposé par :
Mme L. HAMAMI

Promotion 1997

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE D'ALGER

D.E.R. DE GENIE ELECTRIQUE ET INFORMATIQUE

DEPARTEMENT D'ELECTRONIQUE

MEMOIRE

En vue de l'obtention du diplôme

d'Ingénieur d'Etat en Electronique

THEME

Mise au point d'un système OCR

Réalisé par :
M. H. KAHOUL
M. M. DAMOU

Proposé par :
Mme L. HAMAMI

Promotion 1997

Dédicaces

J'e dédie ce modeste travail

- A ma grand mère
- A mes très chers parents pour leur soutien durant toutes mes années d'études
- A toute ma famille
- A tous mes amis et particulièrement Mahiou, Samir, Khaled, Salim, Farid

Houssam

J'e dédie ce modeste travail

- A la mémoire de mon père
- A ma très cher mère
- A ma soeur et mes frères
- A tous les miens qui partagent mes peines et mes veux .

Mohamed

Remerciements

Nous remercions Mme L.Hamami qui nous a confié ce travail , et qui nous a guidé et encouragé tout le long de notre travail.

Nous tenons à remercier M.S.DILMI , M.B.ZEKRINI et M.R.BOUBERTACKH pour le soutien qu'ils nous ont apporté afin de mener à bien ce projet.

Enfin , que toutes les personnes ayant contribué , de près où de loin , à la réalisation de ce projet trouvent ici , l'expression de notre profonde gratitude

ملخص

العمل الذي قمنا به يهد ف إلى إنجا ز نظام تعرف ضوئي للحروف، كامل ، قـادر علـى القيـام بعمليات المعالجة و التعرف للحروف العربية المتعددة الأنماط و الحجوم و الصادرة من نص ممسـوح ضوئيا . بالنسبة للمعالجة نقترح نوعيـن مـن المرشـحات : مرشـح معـدل و آخـر وسـيط . فيمـا يخـص استخراج الحروف من الكلمات فالطريقة المتبعة هي طريقة مقترحة ، تعتمد على بنيـة الكتابـة العربيـة. أما عن الطريقة المتبعة للتعرف فهي طريقة تركيبية ، بنيوية تعتمد على استخراج خاصيـات أساسية (ثقوب و تقعرات) بغية الترتيب و خاصيات ثانوية من أجل التعرف على الحرف و إزالة الإلتباس .

Résumé :

Notre travail consiste à mettre au point un système **OCR** , complet , capable de réalisé les opérations de prétraitement (filtrage et segmentation) et de reconnaissance pour les caractères arabes, imprimés, multitailles et multifontes issus d'un texte scanné. Pour le prétraitement, on propose deux types de filtrage : le moyen et le médian ; pour la segmentation, une méthode heuristique établie après une étude de la morphologie de l'écriture arabe est utilisée. La méthode de reconnaissance pour sa part est une méthode structurelle qui repose sur l'extraction de primitives primaires (trous et concavités) pour la classification et de primitives secondaires pour l'identification et l'élimination des ambiguïtés.

Abtract :

This work presents a full **OCR** system , multifont, multisize for the arabic printed characters extracted from a scanned text. For this , our application propose in the processing phase two kinds of filters : Average and Median. The printing word is directelly segmented into characters on the basis of heuristic criterious. In the step of recognition , the method used is a structural one based upon the extraction of primary caracteristics (holes and curves) to find class that the unknown caracter can belong. Other secondary caracteristics are used to identify the caracter and to reduce confusions.

Sommaire

INTRODUCTION GENERALE

Introduction générale

La reconnaissance des formes s'intéresse à la conception et à la réalisation de systèmes (matériels ou logiciels) capables de percevoir et , dans une certain mesure , d'interpréter des signaux captés dans le monde physique . Chacun de nous fait usage constant de ses sens , tant dans sa vie quotidienne (pour se déplacer , lire son journal , discuter avec un interlocuteur , etc..) que dans ses activités professionnelles (par exemple , pour déceler un léger défaut sur un produit manufacturé , interpréter un spectrogramme de signal sonore ou lire une image de scanner). Dans toutes ces tâches , l'être humain fait appel à la fois à ses organes sensoriels et à une quantité de reconnaissance , indispensables pour interpréter correctement ce qu'il perçoit .Par là même , la reconnaissance des formes se rapproche de l'intelligence artificielle , les deux disciplines partageant souvent les mêmes modèles et s'enrichissant mutuellement .

Les chercheurs et les ingénieurs ont tenté depuis de long années , et avant même la naissance de l'informatique , de concevoir des machines dotées de telles capacités sensorielles . L'intérêt de ces travaux est double : ils couvrent des champs d'application évidemment multiples et contribuent par ailleurs à parfaire notre connaissance des processus cognitifs humains . Depuis l'analyse de scènes polyédriques artificielles simples ou la reconnaissance de quelque mots prononcés par un locuteur unique , au cours des années 50-60 , à la reconnaissance des formes à effectuer des progrès considérables qui permettent désormais la réalisation de prototypes avancés et de produit commercialisables dans des domaines d'application variés , aidées en cela par l'évolution de la technologie .

Dans le cadre de ce modeste mémoire nous proposons une mise au point d'un système OCR (*Optical Characters Recognition*) , multitaille et multifontes destiné plus particulièrement au caractères arabes .

Notre travail s'articule autour des chapitres suivants:

Premier chapitre :

Afin de mieux situer notre travail nous présenterons une description générale des systèmes de reconnaissance de formes.

Deuxième chapitre:

Traite les étapes d'acquisition et de prétraitement en y trouve les organigrammes détaillés de la segmentation ainsi qu'un bref rappel sur les techniques de filtrage et de détection de contours .

Troisième chapitre:

Les étapes d'apprentissage et décision sont regroupées dans ce chapitre à cause de l'étroite collaboration des techniques associées. Nous y discutons sur l'extraction des primitives principales et secondaires ainsi que la structure du dictionnaire.

Quatrième chapitre:

On présentera dans ce chapitre le logiciel que nous avons réalisé. Celui-ci englobe toutes les notions évoquées dans ce mémoire.

Enfin , une conclusion générale terminera ce mémoire.

CHAPITRE I

INTRODUCTION AUX SYSTEMES DE RECONNAISSANCE

1.1 présentation

La reconnaissance des formes (RF) est la première étape d'un long processus de compréhension de notre univers dans le cadre général de la communication Homme-machine. Etant placée en amont de ce processus , la RF se doit de résoudre les problèmes liés au *codage* des formes , à leur pramétrisation et à leur discrimination . Cette tache est souvent difficile pour plusieurs raisons. D'abord , les formes appartiennent à un monde physique dont la transcription numérique est très complexe à cause de l'absence de *capteurs* adaptés à toutes les situations . Ensuite , la nature des formes et leur apparence varient d'un échantillon à un autre (même au sein de la même famille) , ce qui multiplie les dimensions de l'espace de représentation et rallonge les temps de décision.

Il est pourtant essentiel que l'information transmise à l'ordinateur soit la plus adéquate possible vis-à-vis de la forme à traiter et ceci n'est pas a priori une chose simple : il s'agit en effet de passer d'un objet dans un monde physique continu à une forme numérique dans un monde discret . Ce passage , même s'il se fait par la voie de capteurs spécialisés , impose le plus souvent des simplifications de la représentation donc un appauvrissement de l'information et une certaine "mise à plat" des propriétés de la forme .

Une des tâches importantes de la RF est donc de bien caractériser la forme à partir de sa seule transcription numérique , c'est-à-dire de lui trouver une description qui la singularise mais aussi qui la rapproche de ses formes voisines et qui l'éloigne de ses "fausses" semblables. Pour arriver à réaliser finement cette distinction , la RF élargit la caractérisation à l'ensemble des échantillons d'une même famille de formes et juge ainsi globalement de la qualité de la description. Cette tache , qui s'appelle communément *apprentissage* , constitue des groupements de formes ayant les mêmes caractéristiques. Ainsi , plus les caractéristiques sont robustes et différentes d'un ensemble de formes à l'autre , plus il est facile de séparer les formes en familles homogènes et plus il est automatique d'associer une forme à une famille . En revanche , plus les propriétés sont redondantes contradictoires et incomplètes , plus les répartitions en familles sont approximatives et plus il est difficile de décider . On voit là le compromis difficile que doit faire la RF entre la redondance et l'altération des données des capteurs d'une part , et la fiabilité de sa modélisation d'autre part . Ce compromis ne peut être réalisé sans l'aide des techniques de traitement souples et robustes à la fois , d'ou les nombreuses investigations dans ce domaine depuis plus d'un demi-siécle .

Un autre aspect important de la RF est la décision . Décider revient à émettre un avis sur la forme entrée et souvent à lui attribuer une "étiquette" ou un nom correspondant à celui d'une des familles de l'apprentissage. Dans ce cas , on cherche parmi les familles celle qui "maximise" une fonction de

correspondance , de ressemblance , de similitude ou de corrélation entre sa description et celle de la forme entrée . Afin de limiter le nombre de comparaisons et simplifier l'étude de la ressemblance , on définit pour chaque famille quelques représentants ou *prototypes* qui serviront d'uniques références pour les comparaisons. Les comparaisons sont opérées sur les caractéristiques , d'ou , encore une fois , l'importance de la caractérisation dans la mise en oeuvre des règles de décision.

Enfin , l'interprétation est une tache difficile et une même forme ne peut être reconnue de manière "sure" qu'après l'analyse de son contexte . C'est le cas par exemple en lecture optique de formules , ou un même trait horizontal est interprété comme un signe de soustraction ou un trait de fraction selon qu'il sépare horizontalement ou verticalement deux expressions arithmétiques. En vision , une même colonne de briques est reconnue comme une cheminée ou comme un poteau selon qu'elle se trouve placée sur le toit de la maison ou devant sa façade.

1.2 Processus de reconnaissance

La démarche classique suivie en RF consiste à opérer selon le schéma général de la figure1.1. Ce schéma n'est pas purement linéaire . Des interactions peuvent apparaître entre les différents niveaux , pour d'éventuels retours en arrière.

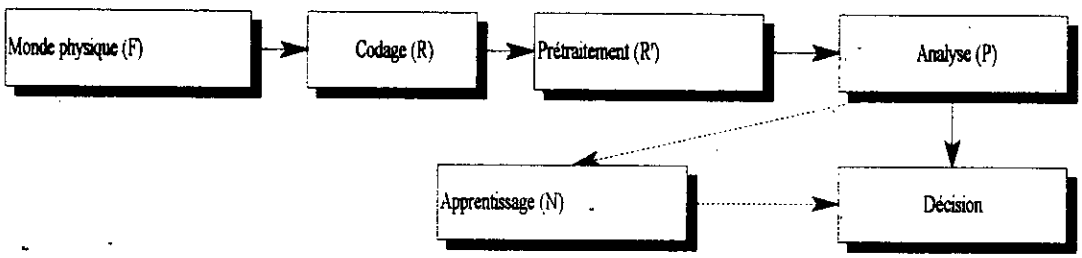


Figure 1.1 : *Schéma général d'un système de reconnaissance des formes*

Nous allons rappeler le rôle de chaque étape dans l'élaboration du processus complet de la reconnaissance .

Le monde physique : La chaîne part du monde physique qui est un espace analogique de dimension infinie appelé *espace des formes (F)* . Les objets , dans cet espace , sont décrits de différentes façons avec une multitude de propriétés dont il serait difficile de tenir compte de chacune à la reconnaissance des formes . La loi de passage au monde discret nécessite forcément une sélection et par conséquent une certaine simplification .

Le codage : C'est une opération de conversion numérique du monde physique continu vers un monde numérique discret . Ce dernier , appelé aussi *espace de représentation (R)* , a une dimension encore trop importante même si elle finie . La dimension r de cet espace est choisie volontairement grande de manière à pouvoir disposer d'un maximum d'informations sur la forme et à pouvoir y sélectionner des sous-ensembles pour des multiples usages.

Le prétraitement : L'étape de prétraitement consiste à sélectionner dans l'espace de représentation l'information nécessaire à l'application . Cette sélection passe souvent par l'élimination du bruit du aux conditions d'acquisition , par la normalisation des données , ainsi que par la suppression de la redondance . Le nouvel espace de représentation (R') a une dimension r' très inférieure à r mais demeure un espace de grande dimension et contient des informations encore assez primitives .

L'analyse : Lors de l'étape d'analyse , les techniques de RF calculent un certain nombre de caractéristiques ou de *paramètres* . Ces paramètres correspondent à des mesures de nature géométrique , topologique ou statistique et servent comme seules données de représentant la forme . Ils sont généralement limités en nombre . Ainsi , l'espace obtenu est l'espace des paramètres (P) de dimension p très petite par rapport à r'.

L'apprentissage : L'apprentissage ou modélisation est une étape clé dans la chaîne de la reconnaissance . Son rôle est d'éclairer la décision à l'aide de connaissance à priori sur les formes . A partir de critères spécifiques aux formes , l'apprentissage tente de définir des modèles de référence ou de caractériser des "classes" de décision . Il permet ainsi de dicter au système l'algorithme de décision le plus adéquat vis-à-vis des règles de modélisation choisies . L'apprentissage peut être paramétrique (la modélisation et les fonctions de décision concomitantes sont définies par un ensemble de paramètres) , ou non paramétrique (la décision ne nécessite pas la spécification du type de paramètres et traduit , par un exemple , par une minimisation du risque d'erreur) .
L'espace (N), ainsi obtenu , s'appelle *l'espace des noms* puisqu'il contient les noms des modèles ou

classes qu'il à formés . Sa dimension *n* correspond globalement au nombre de modèles ou classes existantes.

La décision : La décision ou classement est l'étape de reconnaissance proprement dite. Son rôle est d'identifier la forme test à partir de l'apprentissage réalisé. La méthode de décision est souvent "exhibée" par l'apprentissage , ce qui veut dire que les critères utilisés pour la comparaison sont les mêmes que ceux utilisés pour l'apprentissage . En effet , il est évident que le choix de critères différents pour la décision, n'assurant pas l'uniformité de la description , ne peut pas conduire avec certitude à des résultats cohérents .

Parmi les techniques utilisées , certaines sont fondées sur la notion de proximité et nécessitent de calculer une distance ou une probabilité de ressemblance avec les modèles définis. D'autres sont fondées sur l'analyse de la structure de la forme et essaient plutôt de vérifier une certaine cohérence dans les relations entre les différentes sous-structures.
La réponse de la décision peut être , selon le cas , le nom de la forme en cas de bonne connaissance , plusieurs noms en cas d'ambiguïté , ou bien le rejet de la forme en cas d'incompatibilité de description avec les formes de référence . Dans les deux premiers cas , la réponse peut être accompagnée d'un taux ou score de confiance.

1.3 Reconnaissance de l'écriture

Le but de la reconnaissance de l'écriture est de transformer un texte écrit en une représentation compréhensible par une machine et facilement reproductible par un traitement de texte. Cette technique n'est pas facile vu que les mots possèdent une infinité de représentations dûes aux nombreuse polices de caractères avec de nombreux styles (italique , gras,.....).

1.3.1 Bref historique

L a 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 machine à lire [Ste70 ,Man86]. Tuyrin développa alors la première application d'aide aux handicapés visuels,

mais il a fallu ensuite attendre jusqu'en 1940 pour voir se réaliser la première version informatique de cette application . Depuis , une gamme importante de logiciels et de matériels existe avec des performances assez satisfaisantes. Les premières applications étaient essentiellement orientées vers les chiffres imprimés et quelques lettres d' anglais (latin). La limitation du vocabulaire a permis d'atteindre rapidement des performances élevées (de 96% à 99%). Ensuite , les applications ont été étendus progressivement vers la reconnaissance des caractères imprimés isolés (90% à 97%) .

1.3.2 L'OCR (reconnaissance optique de caractères)

Elle consiste à récupérer automatiquement des textes imprimés ou manuscrits en évitant la phase pénible de la saisie par clavier.

La transformation d'un texte imprimé en code ASCII se déroule en deux phases. La première phase se résume à numériser la page entière et la deuxième phase consiste , à partir d'un fichier graphique (BMP,PCX ,TIFF...) , à distinguer des formes que l'on identifie comme des lettres de l'alphabet .

1.3.3 Les problèmes de l'OCR

Lors de la reconnaissance , on distingue quatre difficultés majeures : distinguer le texte des autres éléments graphiques , reconstituer l'ordre du texte , reconnaître les caractères spécifiques qui varient suivant les langues , et enfin la plus importante étape est d'identifier les caractères quelque soient leur taille , leur épaisseur et la police utilisée. Suivant le type de document , on peut avoir d'autres problèmes tels que le jaunissement du papier et surtout distinguer les textes des photographies dans les magazines. La reconnaissance des caractères constitue la phase la plus délicate.

1.3.4 Les différents aspects de L'OCR

Reconnaissance monofonte , multifonte , et omnifonte

Pour un texte imprimé, un système est dit monofonte s'il ne traite qu'une seule fonte à la fois. Dans ce cas , l'apprentissage est facile et simple à réaliser et le taux de reconnaissance et très

élevé.Un système est dit multifonte s'il est capable de reconnaître un mélange de quelques fontes, parmi un ensemble , préalablement apprises. L'apprentissage doit gérer les ambiguïtés dues à la ressemblance des caractères des différentes fontes . Enfin , un système omnifonte est un système qui permet de reconnaître toute fonte sans l'avoir absolument apprise [Bel92].

Reconnaissance On-line (en ligne ou en temps réel)

La reconnaissance on-line est une reconnaissance dynamique que se déroule pendant l'écriture. Dans ce cas , on utilise la tablette graphique pour l'acquisition du document .

Reconnaissance Off-line (hors ligne ou en différé)

La reconnaissance démarre après l'acquisition du document entré. Ce mode permet l'acquisition d'un nombre important de documents et nécessite un traitement complexe pour retrouver l'ordre de l'écriture.

1.3.5 Les techniques de l'OCR

a. Méthode de corrélation ou de masquage

Le principe de cette méthode est fondé sur l'examen d'un certain nombre de pixels dans l'image du caractère et l'étude de leur appartenance. Une forme est reconnue si l'ensemble de ses pixels sont présents. Cependant à cause de la rigidité de ces méthodes (conçues à partir d'emplacements fixes de pixels) chaque masque de la taille du motif recherché est déplacé sur toute la région où le motif est supposé être et l'on cherche le maximum de la corrélation . Cette méthode est la plus ancienne et n'est plus employée aujourd'hui que pour des produits spécifiques (banque ; poste) et présente l'avantage d'être très rapide , relativement fiable et surtout très facile à mettre en oeuvre. Mais elle correspond à une série de caractères bien particuliers dans une police prédéfinie avec une taille fixe.

b. Méthodes structurelles

Elle sont fondées sur la structure propre du caractère. La structure est exprimée en terme de composants primitifs correspondant à les formes élémentaires telles qu'un rebroussement , un changement d'orientation , un accroissement ou un décroissement de pente. Ces composantes sont appelées primitives. Parmi ces méthodes, on peut citer :

Méthodes de tests : Elles consistent à appliquer sur chaque caractère entré des tests de plus en plus fins sur l'absence ou la présence de ces primitives.

Le processus le plus répandu ou habituel consiste à diviser à chaque test l'ensemble des choix en deux jusqu'à n'obtenir qu'une seule forme correspondant au caractère entré.

Comparaison des chaînes : Les caractères sont représentés par des chaînes de primitives. La comparaison du caractère test avec un modèle de référence consiste à mesurer la ressemblance entre les deux chaînes en calculant les distances.

Méthode syntaxique : Chaque caractère est représenté par une phase dans un langage où le vocabulaire est constitué de primitives. Les caractères d'une même famille sont représentés par une grammaire. La reconnaissance consiste à déterminer si la phase de description du caractère peut être générée par la même grammaire. On définit la distance d'une phrase à une grammaire comme étant la distance minimale de modification à faire subir à la phase pour qu'il soit acceptée par la grammaire.

c. Méthode des experts

Chaque expert a la responsabilité de son caractère. L'avantage de cette technique est la rapidité. Dés qu'un caractère est identifié , on passe à l'autre. Le seul problème de cette méthode est la complexité de programmation d'un expert.

Un système expert est un logiciel issu des recherches en intelligence artificielle (IA) destiné à remplacer l'être humain dans toute tâche d'expertise sur un domaine ou un problème spécifique en tentant de reproduire le raisonnement d'un ou plusieurs experts humains [Ben94] .

d.Méthode heuristique

Les méthode heuristiques sont des procédures basées sur l'intuition et l'expérience de l'automaticien. Les performances et la structure de ces systèmes en dépendront. Cette approche présente les avantages suivants :

- Elle peut simplifier la résolution de problèmes complexes.
- Elle est souple et peut s'adapter à divers types de problèmes.

Les règles et procédures utilisées dans cette technique sont spécifiques aux problèmes à résoudre et ne peuvent donc pas être généralisées à d'autres.

1.4 Champs d'applications

L es applications auxquelles cette discipline est classiquement attachée sont les suivantes:

La reconnaissance de la parole est liée à plusieurs disciplines comme le décodage accoustico-phonétique , la reconnaissance de mots , la reconnaissance de phrases , la reconnaissance du locuteur et la compréhension du dialogue orale homme-machine. Parmi les applications les plus courantes , nous pouvons citer la commande vocale , la dictée automatique , la traduction en temps réel de langues étrangères et la rééducation de malentendants.

La reconnaissance de l'écriture est attachée à la reconnaissance des caractères manuscrits et imprimés , la reconnaissance de mots et de phrases , la reconnaissance du scripteur et la reconnaissance de documents. parmi les applications de la reconnaissance de l'écriture , signalons le tri automatique du courrier par lecture et reconnaissance des adresses , l'authentification de chèques bancaires , la saisie et l'archivage de documents.

la vision comprend le traitement , l'analyse et l'interprétation des images . Les applications sont plus nombreuses que dans les cas précédents. On peut citer la sûreté avec la reconnaissance des empreintes digitales , la médecine avec l'analyse des images de radiographies ou d'échographies , l'armée avec la surveillance et le guidage de cibles , etc.

1.5 Conclusion

En général , un système de reconnaissance de forme 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'aux matériels utilisés.
Par contre le taux de reconnaissance est lié à la dimension du dictionnaire (nombre de prototypes acquis par apprentissage).

CHAPITRE II
ACQUISITION ET PRETRAITEMENT

2.1 Présentation

Nous allons suivre la progression des étapes de traitement donnée par le schéma de la figure 1.1. La première étape d'un système OCR consiste à digitaliser l'écriture et à la présenter au système sous une forme lisible avec un minimum de bruit ou de déformations.

On traitera dans ce chapitre le codage (acquisition) et les techniques de prétraitement qui nous permettent de réduire le bruit.

2.2 Le Codage

L'opération du codage , dite aussi acquisition , consiste à transformer un ensemble de données *analogiques* en un ensemble de données *numériques* de manière à pouvoir les traiter par ordinateur. Cette transformation doit se faire de la manière la plus fidèle possible , c'est-à-dire sans perte d'informations pertinentes et en conservant les propriétés essentielles de l'objet physique.

2.2.1 Dispositif d'acquisition

Scanner

Le scanner est le moyen le plus utilisé en OCR car il permet l'acquisition d'un nombre important de documents et d'avoir directement à sa sortie une image binaire.

C'est un organe périphérique de l'ordinateur permettant la saisie d'informations existantes sur un support papier . Il reçoit la lumière réfléchie par l'image fortement éclairée par des tubes lumineux , ces rayonnements sont convertis en ondes électriques puis un convertisseur transforme l'information électrique en une information numérique.

La résolution d'un tel appareil est exprimée en nombre de points par pouce (ppp ou dpi en anglais), elle désigne sa capacité à digitaliser les traits fins. En OCR , la résolution varie de 200 ppp à 400. Une plus grande résolution n'apporte rien à la précision , au contraire , elle augmente le nombre de points à traiter et génère ainsi un bruit appelé grain du papier [Bel92].

a) Scanner à main

C'est un scanner qu'il faut déplacer à la main sur le document à numérisé. Sa forme est proche est proche de celle d'une souris

b) Scanner de table

b.1.Scanner à défilement

Le document est entraîne à travers une fente pour passer devant la barrette CCD (charge coupled device.).Ce type de scanner n'accepte que les feuilles valantes, il oblige donc à utiliser des photocopies de documents volumineux.

b.2. Scanner à plat

Le document repose sur une vitre et n'est pas déplacé par l'appareil. C'est la barrette CCD qui se déplace le long de la page.

2.2.2 formats des fichiers image

Les fichiers BMP

Ce fichier est devenu grâce aux efforts de Microsoft un format très utilisé pouvant accepter des images de 24 bits et même 32 bits et constitue donc une passerelle de choix entre l'environnement Windows et le monde MSDOS.

Un fichier BMP peut schématiquement se décomposer en quatre parties consécutives. La première appelée <<Bitmap File Header>> fournit des informations générales sur le fichier. La deuxième <<Bitmap Info Header>> contient essentiellement des informations sur l'image proprement dite (dimensions, format des couleurs ...). Si les couleurs ne sont pas codées sur 24 bits, la troisième partie fait correspondre à chaque code une valeur absolue sur 24 bits. Enfin la quatrième partie <<Bitmap>>rassemble le codage complet de chacun des pixels de l'image. La plupart des informations sont stockées sous forme d'un mot de 32 bits (4 octets) afin de mieux coller à l'architecture 32 bits des processeurs d'Intel 80386 et 80486 et d'utiliser au mieux leurs registres internes.

A noter enfin, que la taille en octets d'une ligne doit être en multiple de 4 et que le format BMP a été crée pour P.C. En effet, selon que l'on travaille avec processeur Intel ou Motorolale sens des octets n'est pas le même : poids faible suivi du poids fort dans le premier cas et poids fort suivi du poids faible pour le second. Le transfert d'un fichier BMP dans un autre monde que celui du P.C. (Macintosh) doit donc tenir compte de ce facteur. Il suffit d'inverser le sens des octets du Header

(entête) servant à décrire les paramètres de type INT (entier sur 2 octets) et LONG (entier signé sur 4 octets)[Sou92].

Structure du fichier BMP

a) L'entête du fichier (BITMAP FILE HEADER)

La longueur de l'entête du fichier BMP est toujours fixée à 14 octets . Ces octets permettent de donner des informations sur le fichier . Les deux premiers bytes permettent d'identifier par les deux lettres ASCII << BM >> . Les 4 suivants indiquent la taille totale du fichier , viennent ensuite 4 octets à zéro puis les 4 derniers contiennent la valeur de l'offset de la partie bitmap .

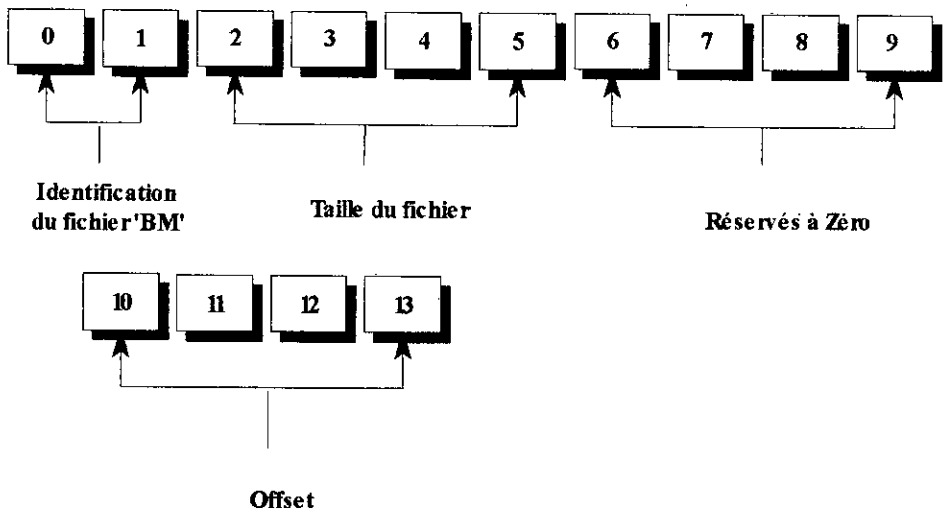


Figure 2.1 *Entête du fichier BMP*

b) Informations sur l'image : BITMAPINFO HEADER

Dans cette partie, le rôle de chaque byte est donné dans le tableau suivant :

Nombre de bytes	Nature de l'information lue
4	la taille occupée par le bitmap Info Header (pour un fichier BMP Windows. 3.1, cette taille vaut 40 octets).
4	la largeur de l'image exprimée en pixels
2	nombre de surface de l'image
2	nombre de bits par pixel (1, 4, 8ou 24).
4	méthode de compression (0 :non compression)
4	taille de l'image en octets.
4	Résolution de l'image en pixel par mètre.
4	Nombre de couleurs utilisées (0 :toutes les couleurs).
4	Nombre de couleurs importantes.

Les fichiers TIFF

Le fichier TIFF conçu en 1987 par Aldus et Microsoft est sans aucun doute le fichier image le plus répandu à l'heure actuelle sur P.C , et le plus privilégié dans les applications graphiques de l'univers MS DOS . TIFF s'apparente en réalité à la liste chaînée , chaque image contient un descripteur (IFD) et un pointeur sur le descripteur suivant s'il existe . Chacun de ces IFD est en fait un tableau contenant les éntités qualifiées de Tag d'où le nom de Tagged Image File format.

A l'inverse des autres types de fichiers , la structure du TIFF est flexible et dynamique. Il peut contenir plusieurs images oubien la même à des résolutions différentes et on ne risque pas de rencontrer dans l'entête du fichier l'un des descripteurs. Les images stockées peuvent être en couleur, en échelle de gris et d'une taille aussi grande que voulue. Il est possible de coder à peu prés toutes les caractéristiques que l'on souhaite dans un fichier TIFF mais rares sont les applications capables d'exploiter le format au sens large du terme.

Pour ce qui est des données , TIFF ressemble à une sorte de liste chaînée . Chaque Tag étant en fait un indicateur spécialisé régissant une caractéristique de l'image. C'est dans cette structure que réside l'universalité de ce format. Il est en effet possible de faire évoluer un tel format au fur et à mesure qu'apparaissent de nouvelles contraintes [Poor91] .

- Les deux premières octets (0 et 1) sont soit (49H et 49H) ou (4DH et 4DH) qui correspondent aux caractères de l' ASCII (I I) pour INTEL et (M M)pour MOTOROLA
- Les deux octets suivants (2 et 3) contiennent le numéro de la version (42 pour une compatibilité éventuelle).
- les derniers quatre octets (4,5,6,7) contiennent un pointeur vers la 2éme partie c'est-à-dire le premier IFD.

b. L'IFD(Image File Directory)

C'est le répertoire du fichier , il contient des informations sur l'image, comme les dimensions ,les couleurs , la compression ,ect . Au début d'un IFD on trouves le nombre de Tags ou champs que contient cet IFD , écrit sur deux octets , juste après 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 deux premiers octets correspondent au type d'informations représentées par le champ (largeur de l'image , nombre de bits par pixel, format de la compression , échelle de gris , ect).
Ils représente le code du champs . Il est à noter que certains champs peuvent être absents et que les champs peuvent être placés dans un ordre quelconque puisque chacun est repère par une valeur appropriée. Mais il est recommandé de les placer par ordre croissant.

- Les deux octets suivants indiquent le type (au sens informatique) des données:

1	: Octet
2	: Code ASCII 8 bits
3	: Short (entier non signé sur 16 bits)
4	: Long (entier non signé sur 32 bits)
5	: Rational(rapport de 2 longs.

- Les quatre octets suivent indiquent le nombre de données ,s'il s'agit par exemple de la longueur de l'image ce nombre est égal à 1, si la donnée représente une chaîne de caractères alors le nombre de données sera la longueur de cette chaîne .

- Les quatre suivants contiennent la donnée elle- même . Si cette donnée ne tient pas sur 4 octets alors les quatre derniers octets contiennent un pointeur vers la section supplémentaire ou cette information sera retrouvée. Cette section peut être placée entre le pointeur de la prochaine IFD et le répertoire suivant.

Calculer la dimension de la donnée à partir des deux informations :Nombre de données et le type de données .Si l'information représente quatre octets ou plus , la données représente en fait un pointeur.

c. BITMAP

C'est l'image même les information la concernant se trouvent dans la partie précédente .
Il est à noté que les dimensions de l'image ,la largeur et la hauteur sont codé respectivement par 256 et 257, le nombre de bits par pixel codé par 258 et l'offset par 273.

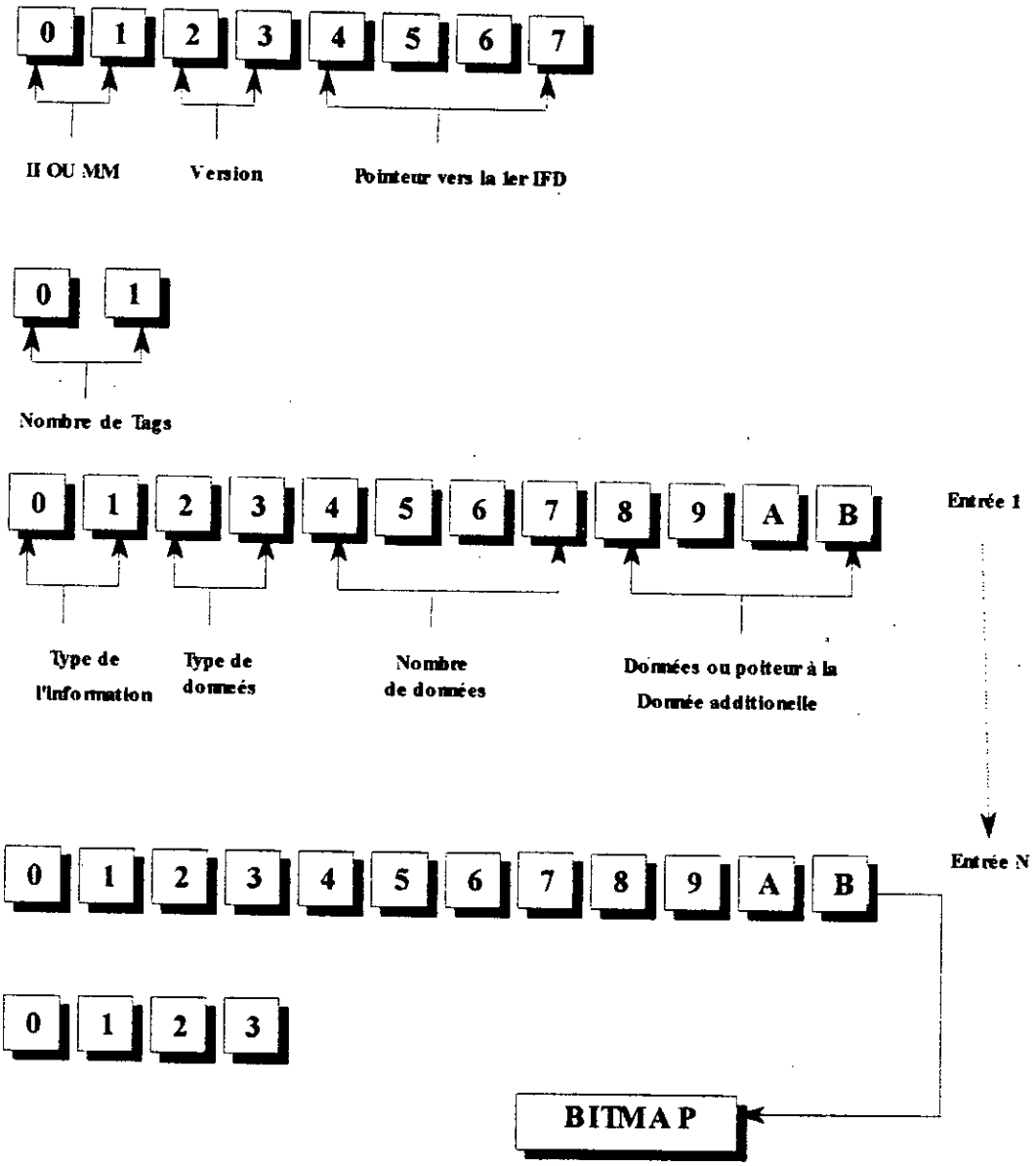


Figure 2.2 . *Constitution d'un fichier Tiff*

2.3 Prétraitement

Le rôle du prétraitement est de "préparer" les données reçues du capteur à la phase suivant d'analyse consacrée à l'extraction des paramètres . Cette phase n'est possible et surtout fiable que si les données du capteur sont dénuées de *bruit* , corrigées de leurs erreurs éventuelles , *homogénéisées, normalisées* et *réduite* à l'essentiel. Toute les techniques élaborées dans ce sens se

gardent de modifier les propriétés essentielles des formes , ce qui pourrait conduire , dans le cas contraire ,à de graves erreurs d'analyse et plus tard de reconnaissance.

2.3.1 Suppression du bruit

Le but de cette étape est de débarrasser les données image du *bruit* de l'acquisition et de ne garder que l'information significative de la forme représentée . Le bruit peut être du bien sure aux capteurs , mais aussi aux conditions de prise de vue (éclairage, positionnement,...).
parmi les méthodes de suppression du bruit citons le filtrage (moyen, médian) et la binarisation

Le filtre moyen (lissage)

Un pixel est remplacé par la valeur moyenne de ses pixels voisins. C'est un filtrage spatial passe bas qui supprime le bruit lorsque celui-ci est aléatoire et cohérent avec l'image , mais à l'inconvénient de réduire la résolution spatiale de l'image .Voici deux exemples de masques qui peuvent être utilisés pour réaliser le lissage :

$$\frac{1}{9}\begin{pmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{pmatrix} \quad et \quad \frac{1}{16}\begin{pmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{pmatrix}$$

Le filtre médian

Ce filtre n'est pas le résultat d'une combinaison linéaire de pixels. L'exemple le plus simple est le filtre de Tuckey .,un pixel est remplacé par la valeur médiane de ses pixels voisins.

Le seuillage (binarisation)

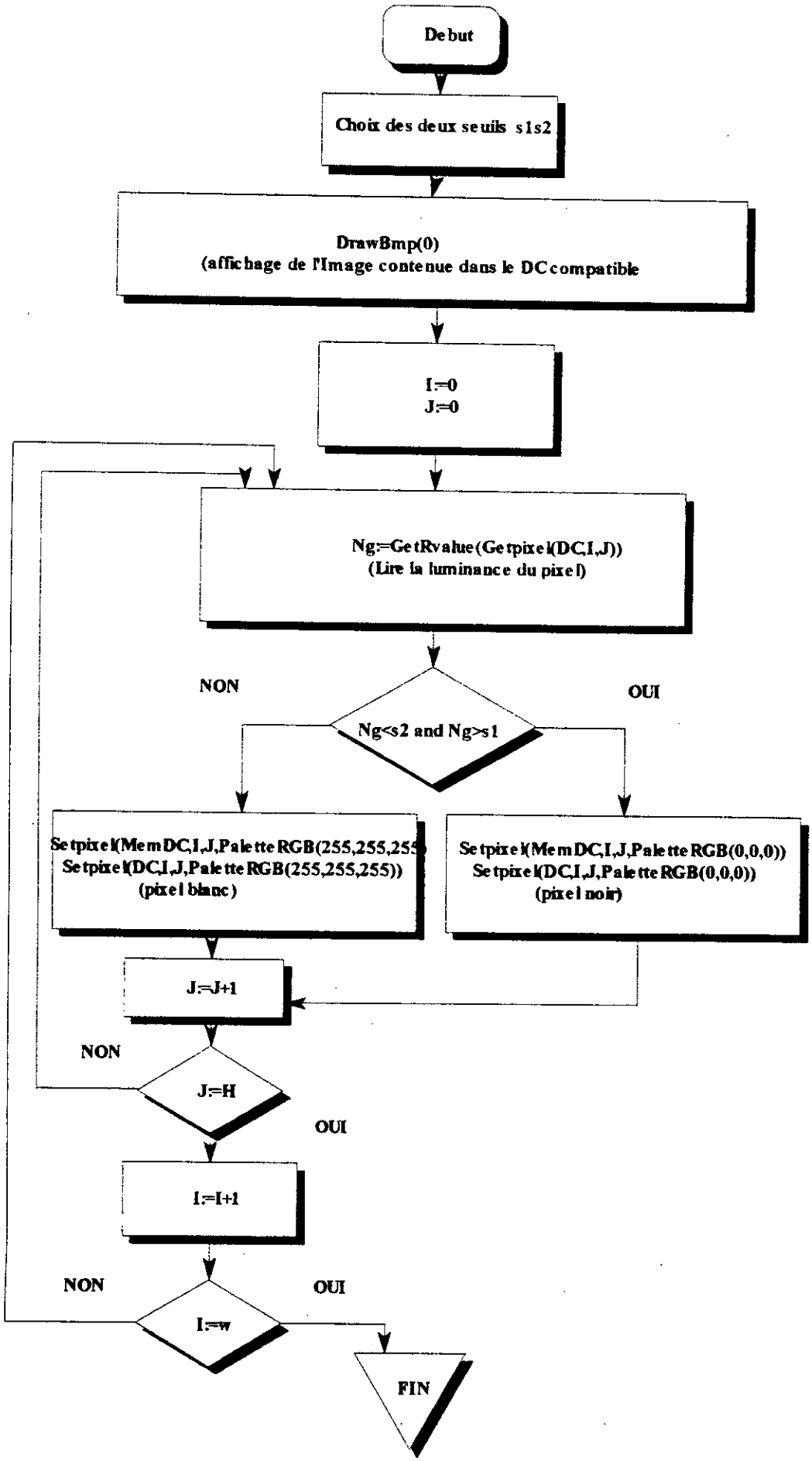
Cette fonction consiste à examiner les pixels et à ne retenir que ceux dont la valeur est comprise entre deux seuils de donnés.Ceci a pour but de sélectionner des plages de niveaux de gris correspondant à des régions d'intérêt dans la scène étudiée . Selon le cas , l'utilisateur peut affecter un code différent à chaque région ou obtenir une image binaire ou le noir représente toutes les régions et le blanc représente le fond . Dans ce dernier cas l'opération de seuillage s'appelle *binarisation* .

Voici un exemple de binarisation

$$\forall p \in E \Rightarrow F(f(p)) = \begin{cases} 0 \ si \ f(p) < S1 \ ou \ f(p) > S2 \\ 1 \ \text{sinon} \end{cases}$$

ou *S1* et S2 sont deux seuils de détermination d'une plage de niveaux de gris .

Dans la figure 2.3 on présente l'organigramme détaillé de la binarisation ,que l'on à utiliser dans l'application.


Debut
Choix des deux seuils s1s2
DrawBmp(0)
(affichage de l'Image contenue dans le DC compatible
I:=0
J:=0
Ng:=GetRvalue(Getpixel(DC,I,J))
(Lire la luminance du pixel)
NON
OUI
Ng<s2 and Ng>s1
Setpixel(MemDC,I,J,PaletteRGB(255,255,255)
Setpixel(DC,I,J,PaletteRGB(255,255,255))
(pixel blanc)
Setpixel(MemDC,I,J,PaletteRGB(0,0,0))
Setpixel(DC,I,J,PaletteRGB(0,0,0))
(pixel noir)
J:=J+1
NON
J:=H
OUI
I:=I+1
NON
OUI
I:=w
FIN

2.3.2 La segmentation

La segmentation est une des étapes les plus importantes dans la chaîne de la reconnaissance des caractères . Elle consiste à sélectionner l'information nécessaire à l'application . Toutes les opérations de reconnaissance s'appuient sur ses résultats d'où l'importance accordée a cette étape.

Les techniques de segmentation employées dans le domaine de la reconnaissance des caractères tentent de localiser les lettres (caractères) dans le mot en indiquant leurs limites gauche et droite. Dans le cas du texte arabe une telle opération présente quelques difficultés dues aux propriétés contextuelles propres à l'écriture arabe. Cette difficulté a souvent conduit les chercheurs à éviter la segmentation et à tenter la reconnaissance directement sur la totalité de la chaîne .

L'étape de segmentation peut être subdivisée en 3 sous -étapes :

- La segmentation horizontale : On procède ici à la localisation des lignes de texte.
- La segmentation verticale : Chaque ligne de texte obtenue dans la 1ère étape est traitée pour donner les différentes parties connexes la constituant .
- La segmentation en caractères : Cette étape vise l'obtention des caractères à partir des parties connexes préalablement trouvées.

a) Segmentation horizontale

Le principe de la segmentation horizontale consiste à détecter , en premier lieu , le premier pixel noir et cela en parcourant les lignes de l'image par un balayage horizontal . Une fois la ligne début marquée , on poursuit le balayage jusqu'à obtenir une ligne qui ne contient aucun pixel noir :C'est la ligne fin .

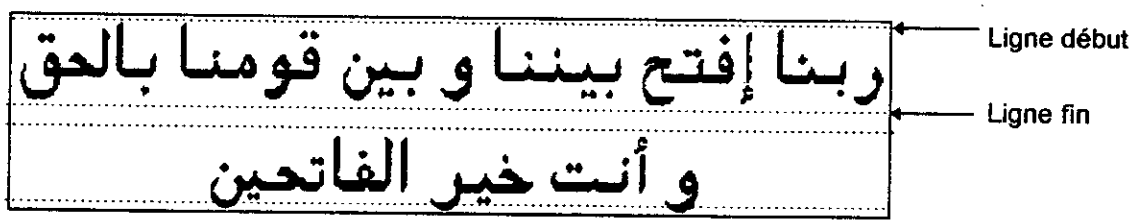
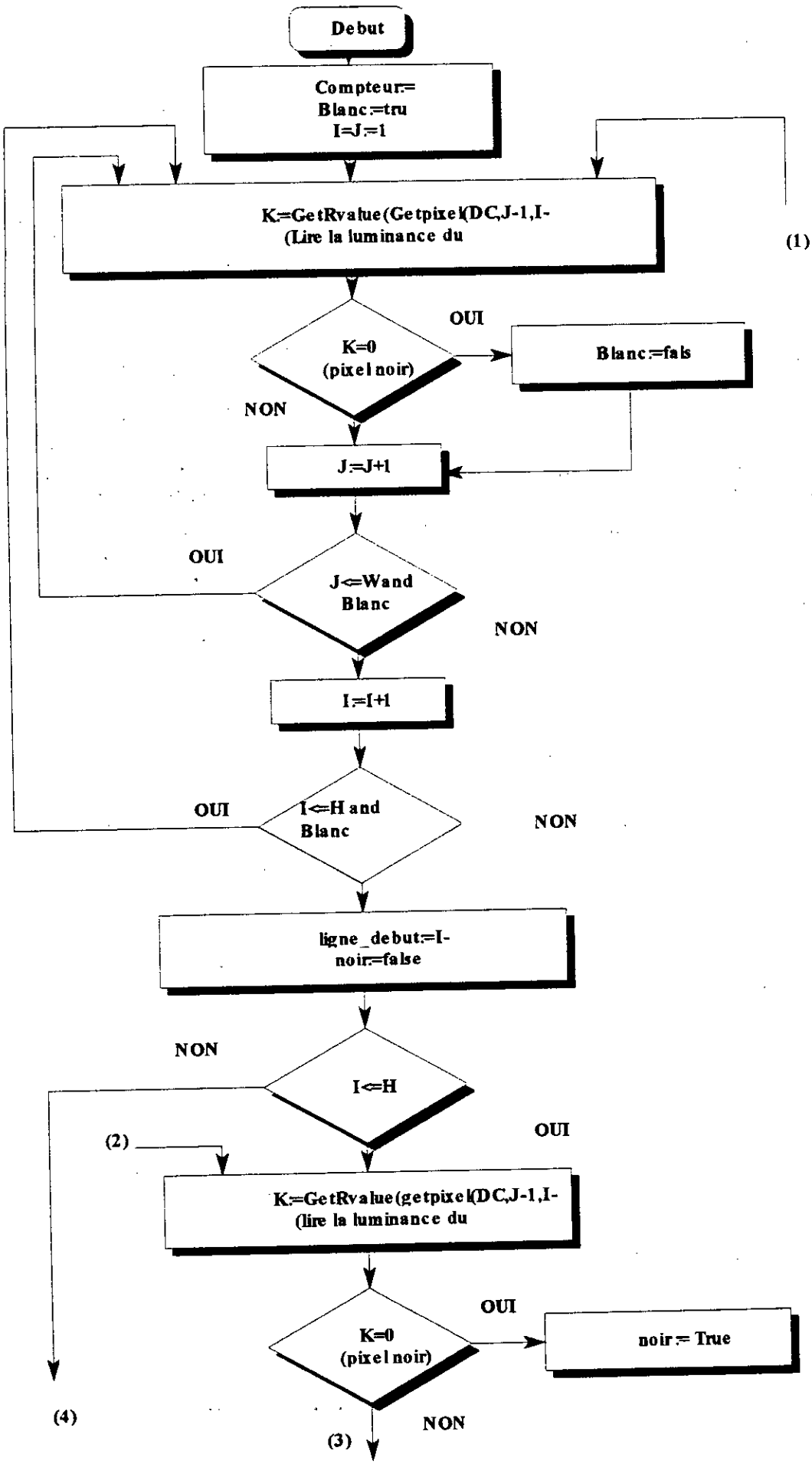


Figure 2.4. *Segmentation horizontale*

L'organigramme de la figure 2.5. représente les différentes étapes de la segmentation horizontale .


Debut
Compteur:=
Blanc:=tru
I=J:=1
K:=GetRvalue(Getpixel(DC,J-1,I-
(Lire la luminance du
(1)
OUI
K=0
(pixel noir)
Blanc:=fals
NON
J:=J+1
OUI
J<=Wand
Blanc
NON
I:=I+1
OUI
I<=H and
Blanc
NON
ligne_debut:=I-
noir:=false
NON
I<=H
OUI
(2)
K:=GetRvalue(getpixel(DC,J-1,I-
(lire la luminance du
OUI
K=0
(pixel noir)
noir := True
(4)
NON
(3)

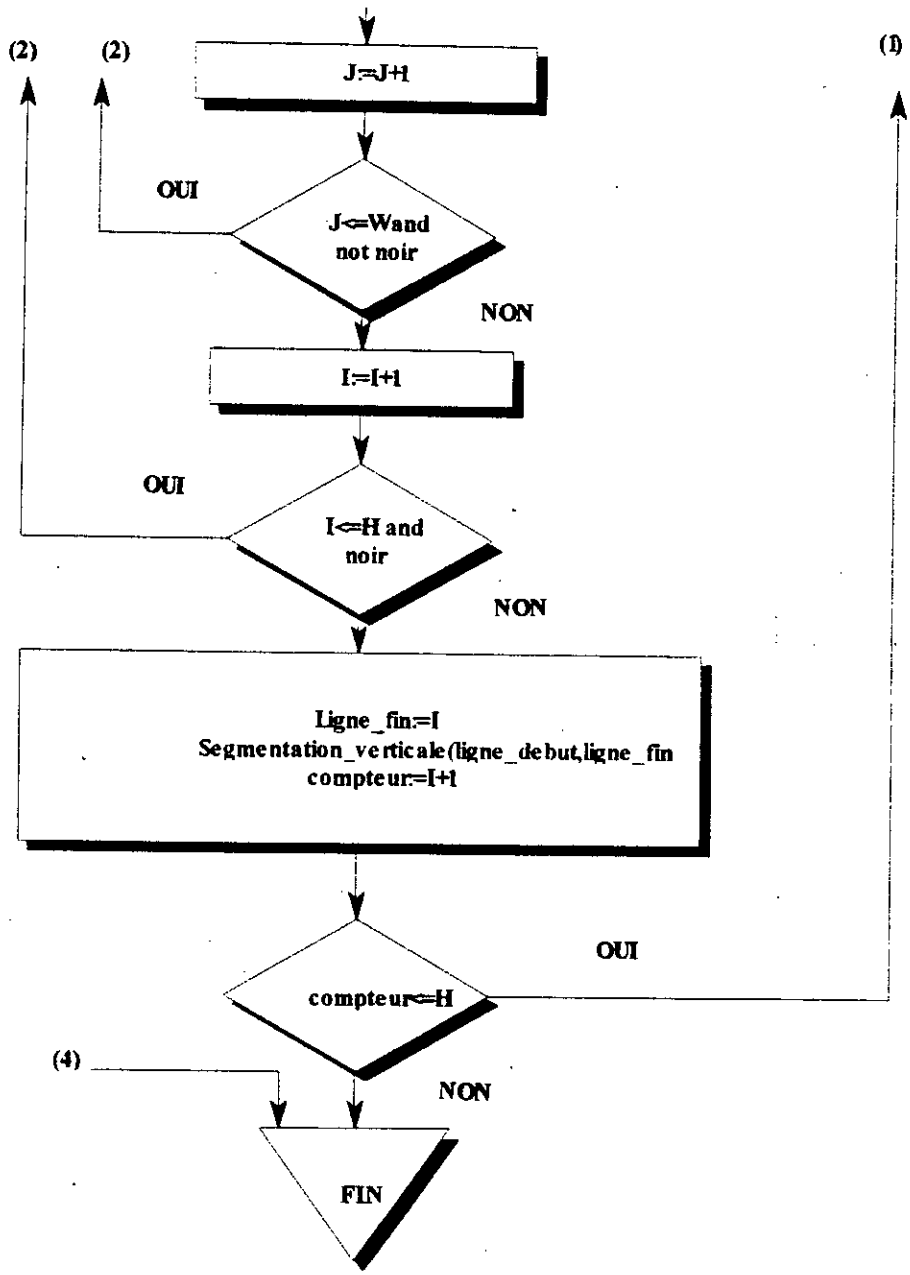


Figure 2.5. *Organigramme de le segmentation horizontale*

b) Segmentation verticale

Disposant maintenant des lignes , nous procédons à la segmentation verticale c'est à dire la séparation en parties connexes .

Notons que les mots ou sous - groupes de mots sont constitués de blocs continus de colonnes non vides séparés par des espaces vides . Il est à remarquer qu'à ce stade on ne peut séparer les caractères attachés d'un mot .

Tout comme la segmentation horizontale , la segmentation verticale est basée sur le repérage au préalable d'une colonne qui contient au moins un pixel noir (*colonne début*) et ce par le balayage de toutes les lignes . Dés l'obtention de cette colonne , on parcourt toutes les colonnes par un balayage de toutes leurs lignes jusqu'à obtention d'une colonne qui ne contient aucun pixel noir *(colonne fin)* .

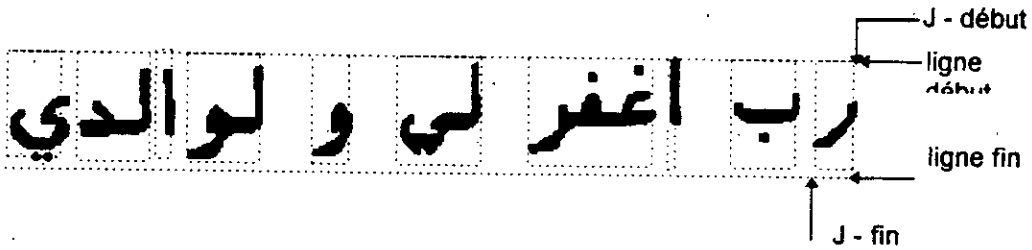
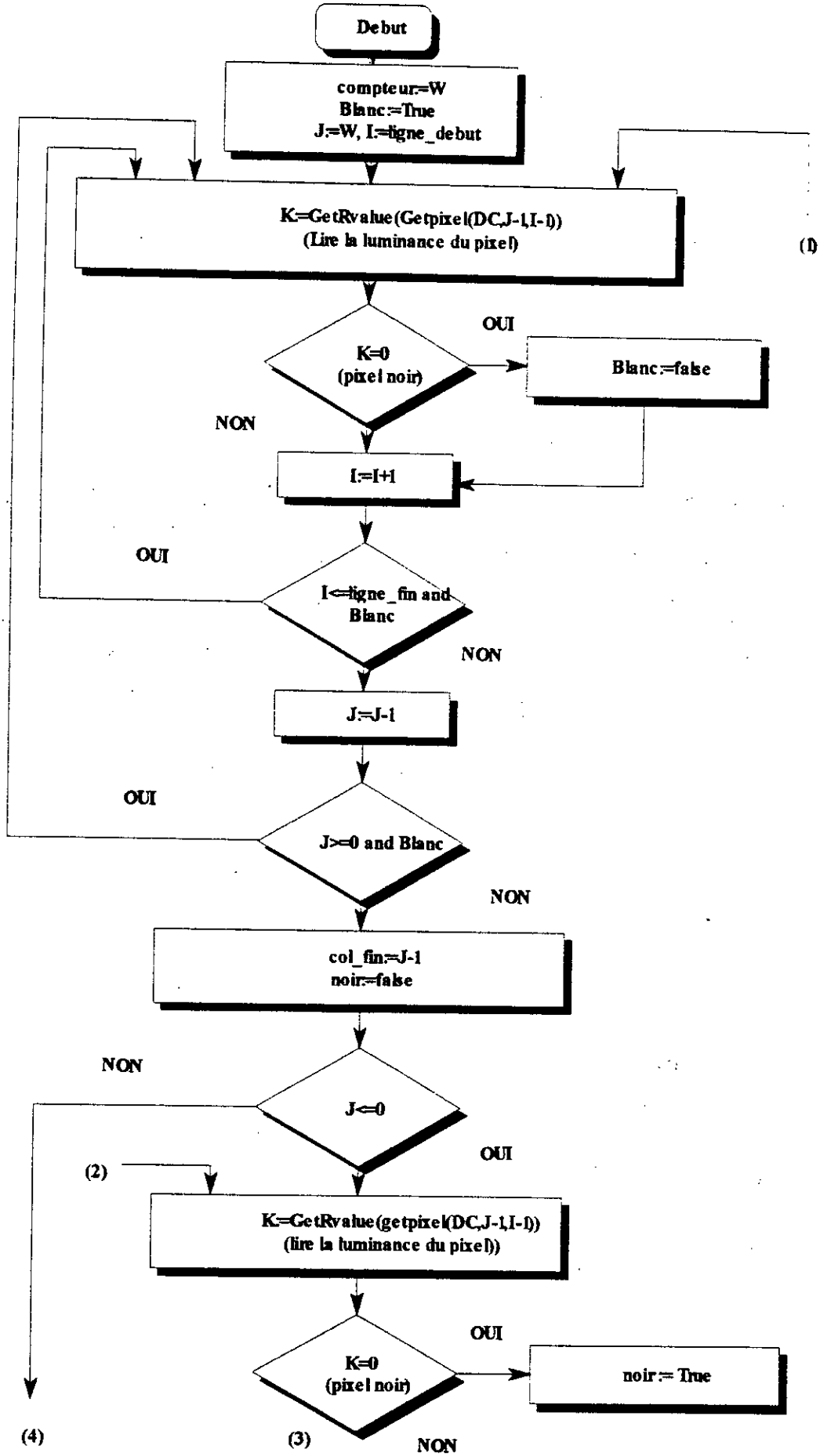


Figure 2.6. *Segmentation verticale*

L'organigramme de la figure 2.7. illustre en détaille la segmentation verticale .


Debut
compteur:=W
Blanc:=True
J:=W, I:=ligne_debut
K:=GetRvalue(Getpixel(DC,J-1,I-1))
(Lire la luminance du pixel)
(1)
K=0
(pixel noir)
OUI
Blanc:=false
NON
I:=I+1
OUI
I<=ligne_fin and
Blanc
NON
J:=J-1
OUI
J>=0 and Blanc
NON
col_fin:=J-1
noir:=false
NON
J<=0
OUI
(2)
K:=GetRvalue(getpixel(DC,J-1,I-1))
(lire la luminance du pixel))
K=0
(pixel noir)
OUI
noir:= True
(4)
(3)
NON

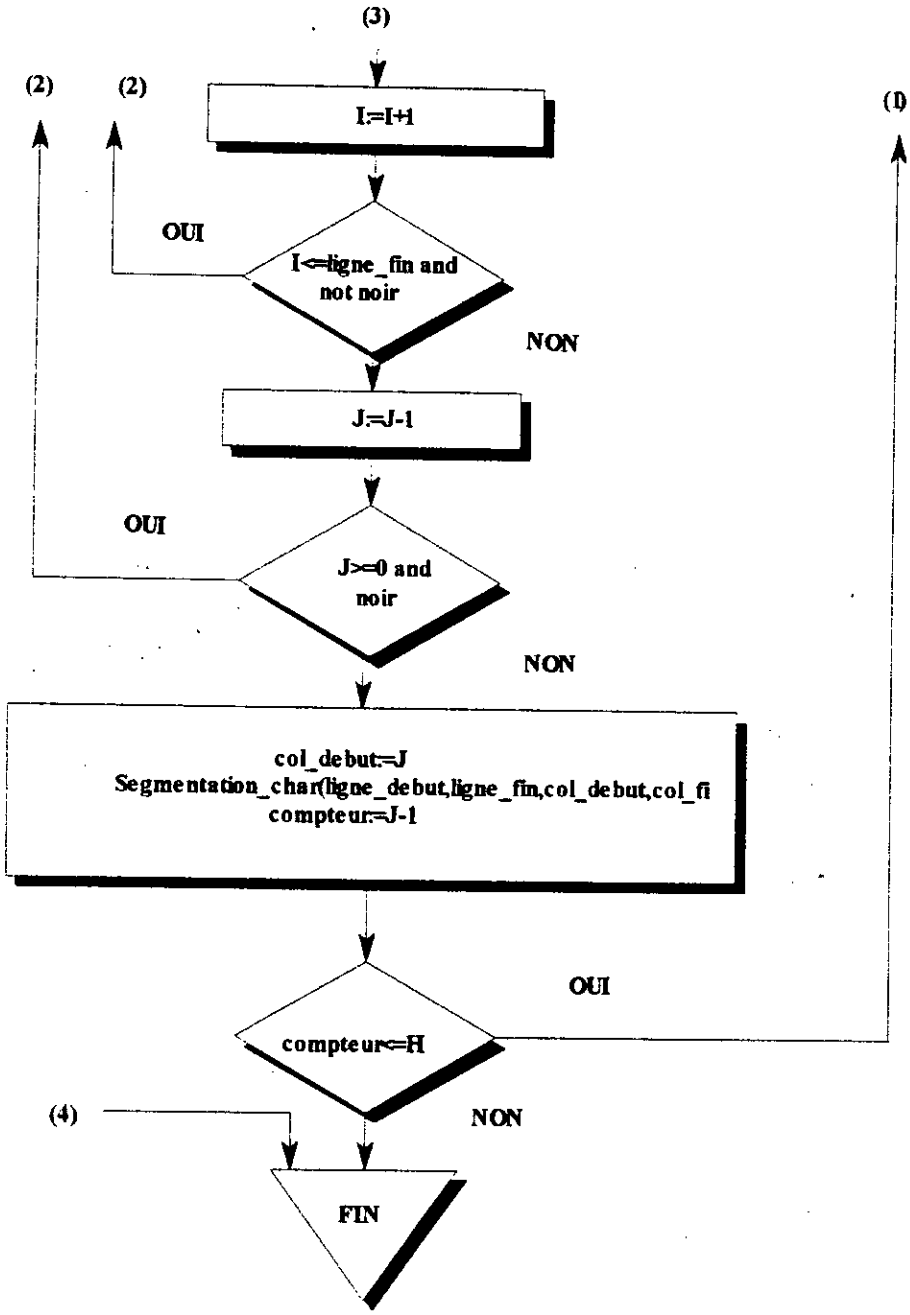


Figure 2.7 *Organigramme de la segmentation verticale*

c) Segmentation en caractère

La segmentation en caractères est , à priori , l'opération la plus complexe puisque l'on travaille sur une courbe continue évoluant d'une façon ascendante .

Cette opération est basée sur un certain nombre de critères qui vont permettre une séparation réelle des caractères .

Généralement les méthodes de segmentation en caractères reposent uniquement sur des considérations heuristiques sur la spécificité des caractères arabes . A l'intérieur d'un même mot , les caractères sont reliés entre eux par des segments horizontaux tracés de la droite vers la gauche .

Pour les critères de liaison deux résultats apparaissent :

- La liaison entre deux caractères se fait toujours dans la région centrale de la ligne de texte : C'est le niveau de la ligne de jonction.
- La ligne de jonction constitue la région qui a la plus forte concentration de pixels en lignes.

Caractéristiques: Dans ce qui suit nous présentons les différentes caractéristiques qui seront à la base des critère de séparation .

La ligne maximale: Dans une partie connexe , on procède pour chaque colonne à un balayage de toutes les lignes . La ligne maximale correspond à la ligne qui contient le premier pixel noir rencontré.

l'organigramme détaillé est représenté dans la figure 2.9.

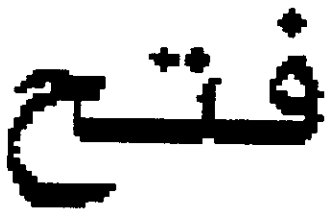


Figure 2.8. *les principaux caractéristiques*

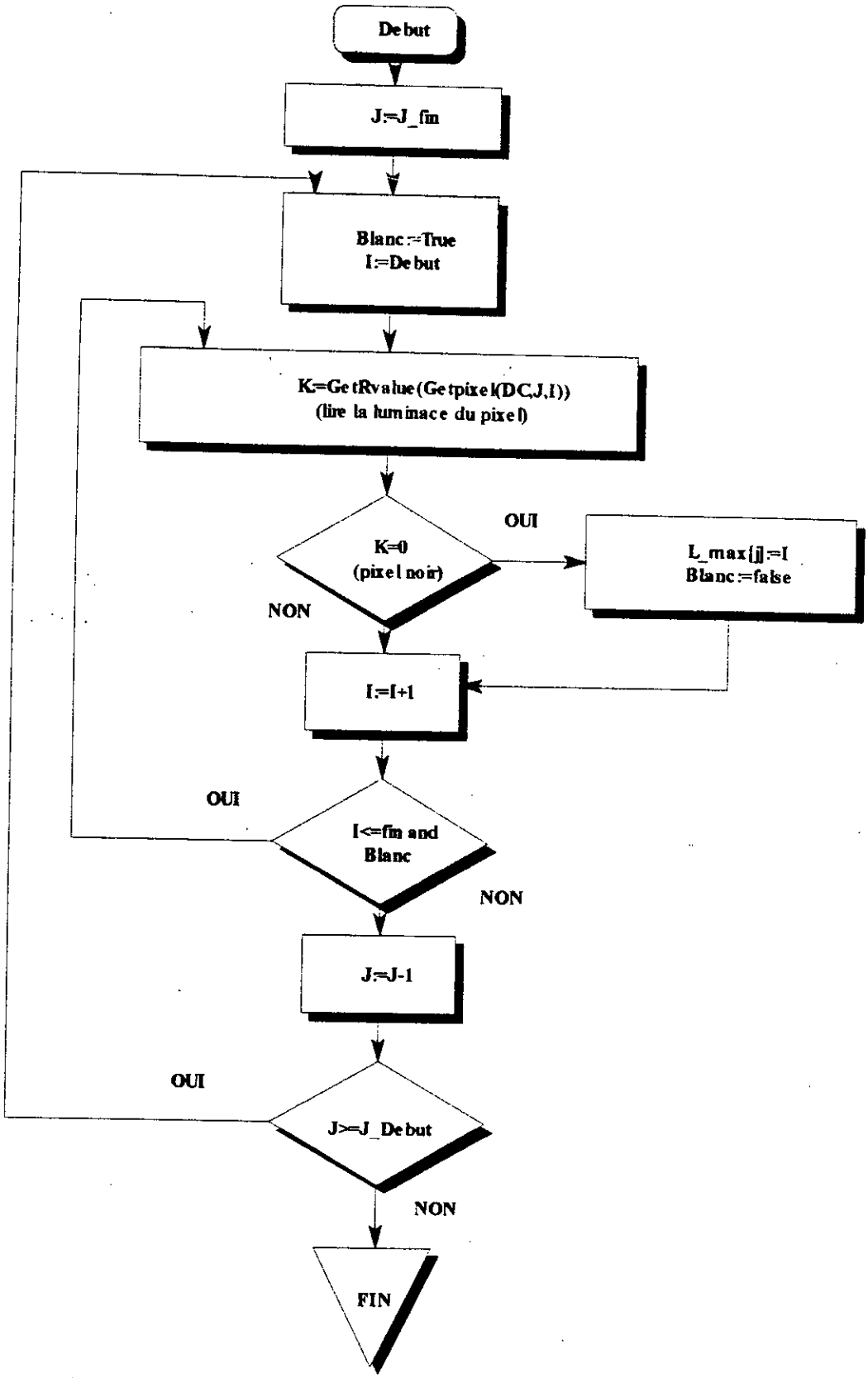


Figure 2.9. *Organigramme du calcul de la ligne max de chaque colonne*

La ligne minimale: Contrairement à la procédure précédente , le balayage débute cette fois de la dernière ligne . La ligne minimale correspond à la premier ligne contenant un pixel noir.

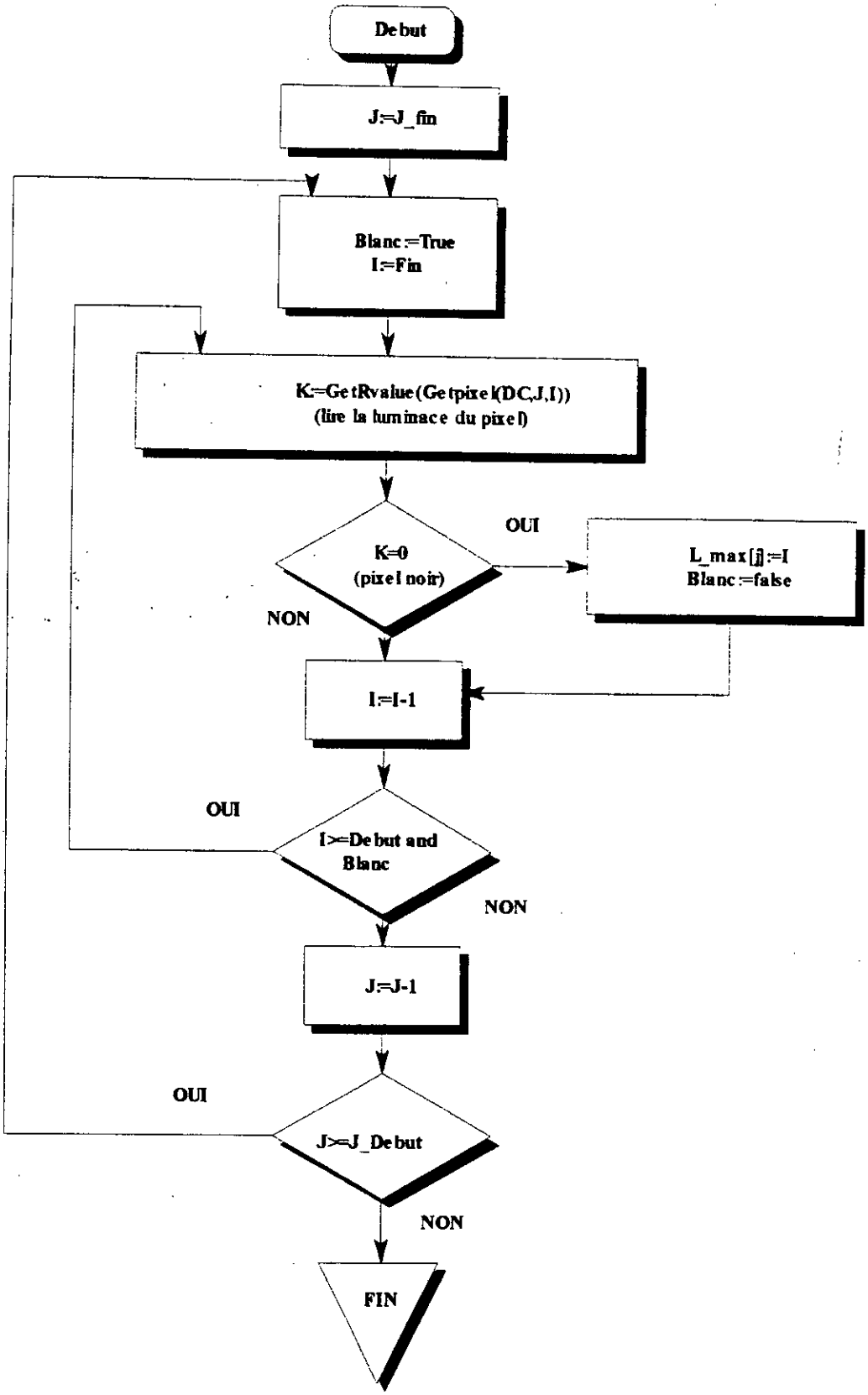


Figure 2.10 *Organigramme du calcul de la ligne min de chaque colonne*

l'Histogramme de chaque colonne: Il correspond à la somme des pixels noirs contenus dans une colonne. L'organigramme ci-dessous illustre la méthode de calcul .

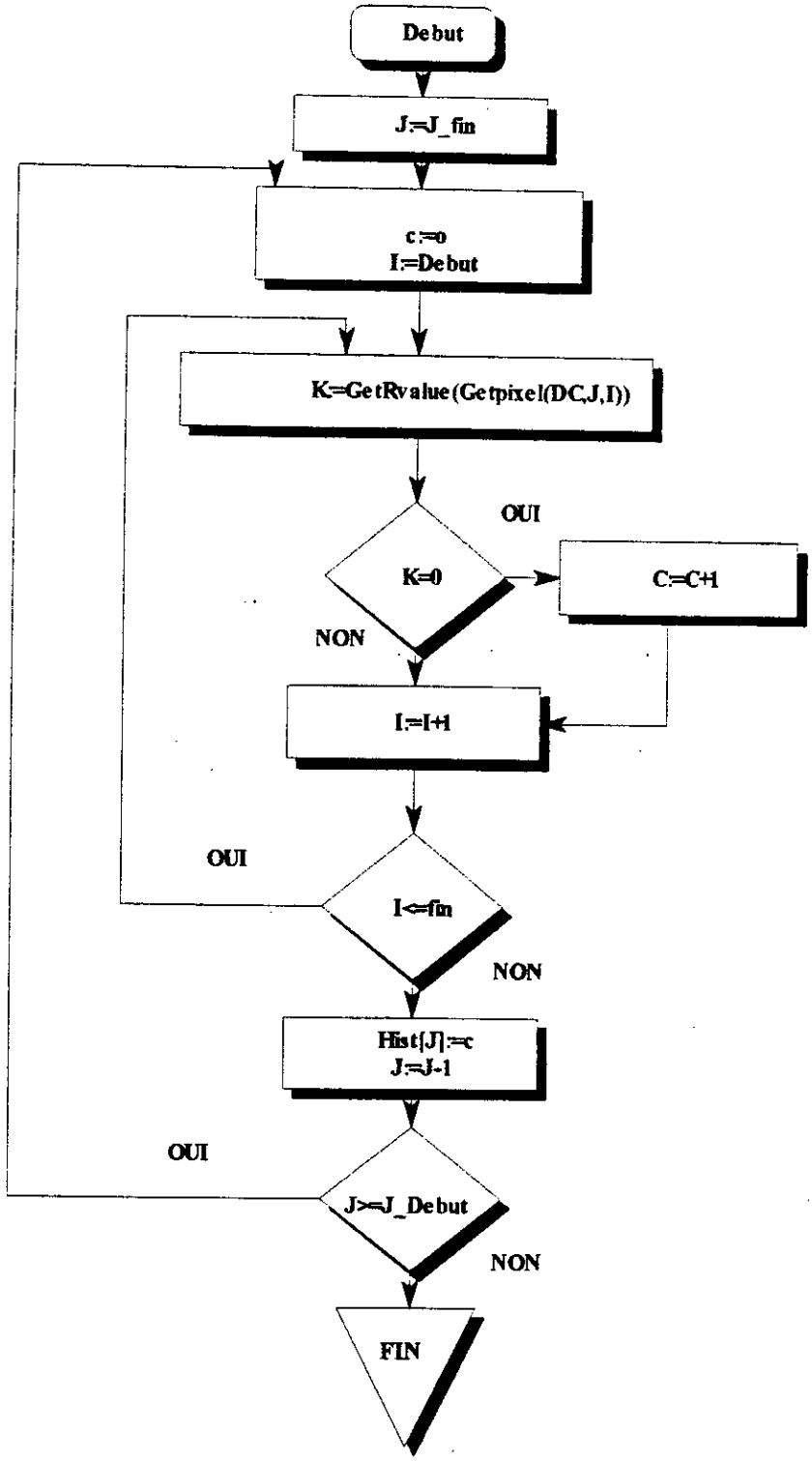


Figure 2.11. *Organigramme du calcul de l'histogramme de chaque colonne*

Calcul du seuil: Par hypothèse , le seuil correspond à la valeur qui a la plus forte redondance dans l'histogramme vertical . Donc il suffit de comparer la valeur de l'histogramme pour chaque colonne donnée avec celle des autres colonnes , le seuil étant la valeur à plus grande répétition N.

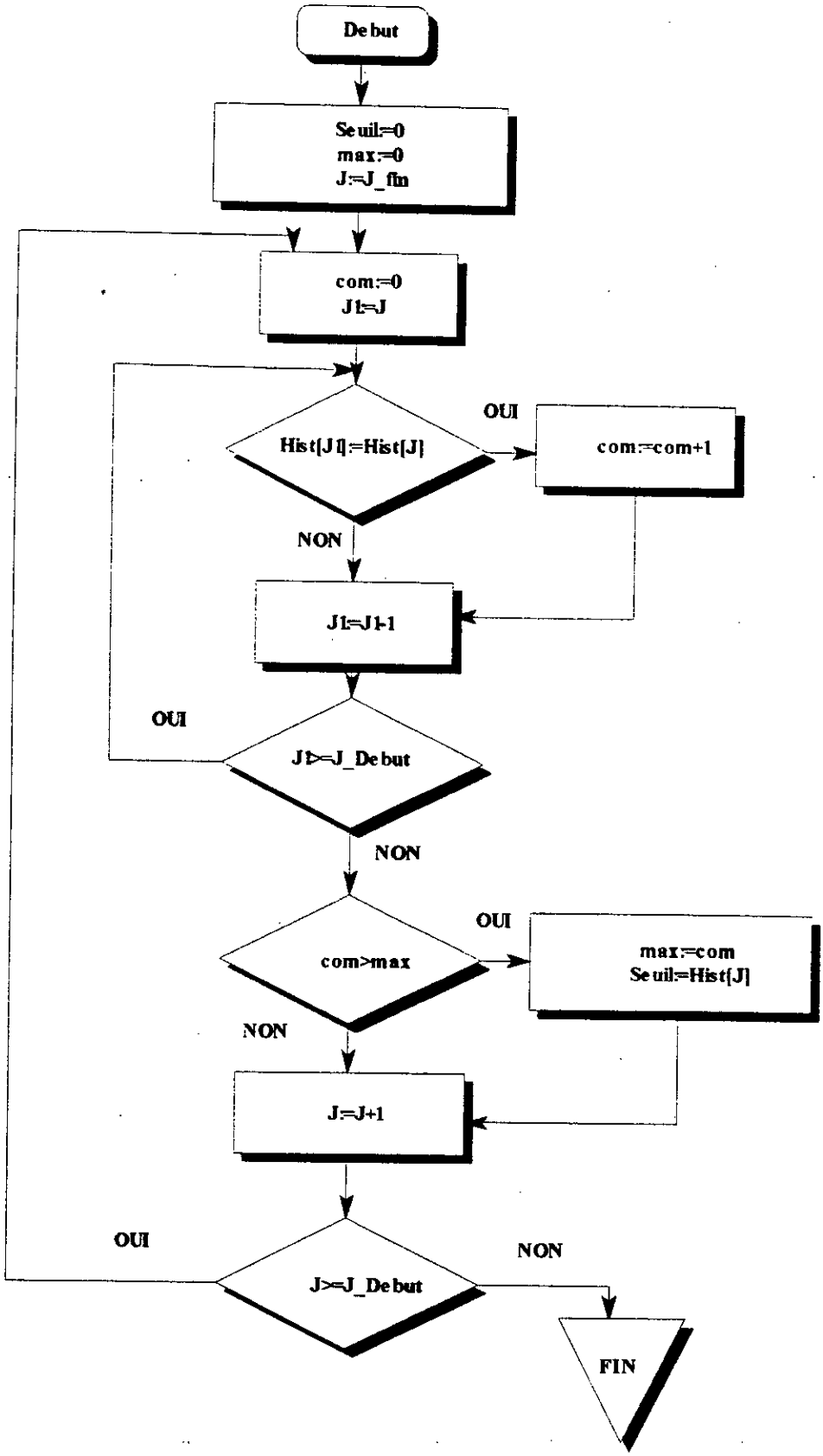


Figure 2.12. *Organigramme du calcul du seuil*

Détection de la ligne de jonction (ligne médiane): La ligne de jonction constitue la région qui a la plus forte concentration de pixels en lignes . Sa détection consiste donc à calculer l'histogramme horizontal et à extraire la ligne qui présente la plus grande redondance.

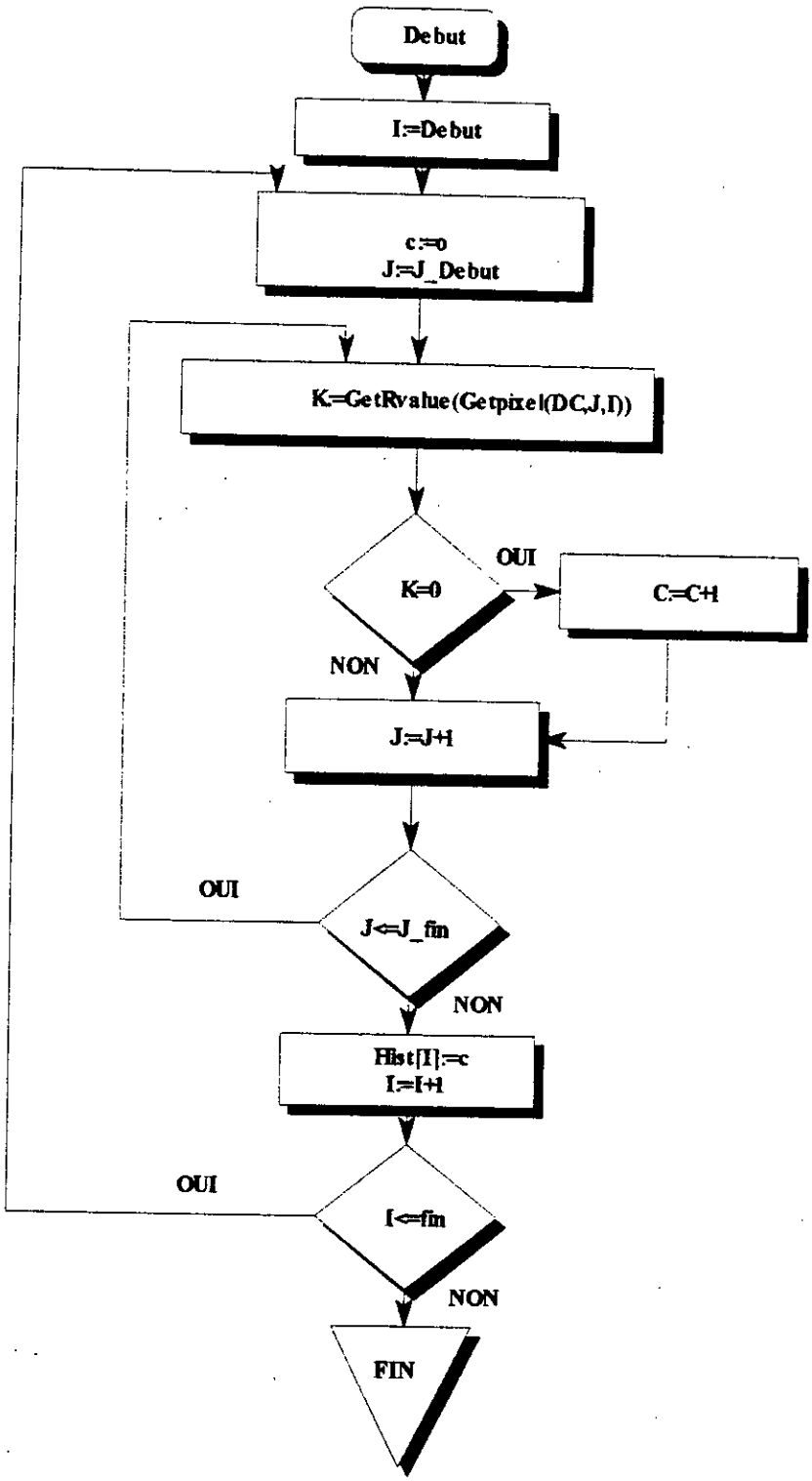


Figure 2.13. *Organigramme du calcul d'histogramme de chaque ligne*

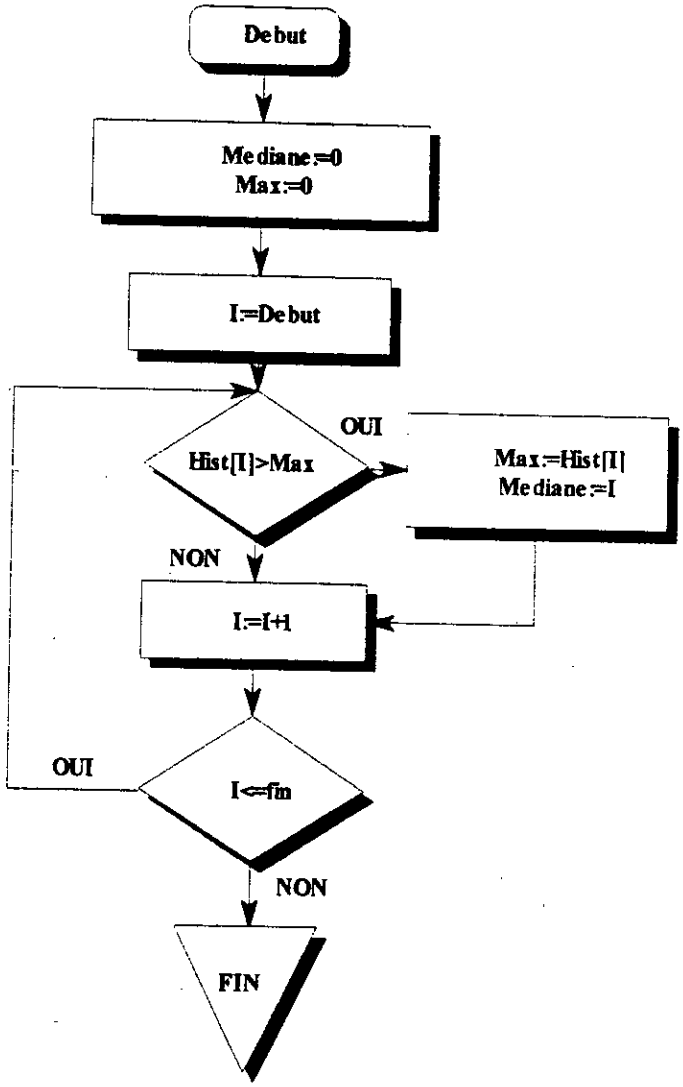


Figure 2.14. *Organigramme du calcul de ligne médiane*

Critères: Le processus de l'extraction des caractères à partir de la partie connexe est fort délicat et ce principalement à cause de l'aspect cursif de l'écriture arabe. Le principe de cette méthode peut se résumer ainsi :

Pour détecter le début du caractère il faut que:

- La différence entre la ligne minimale et la ligne maximale soit strictement supérieure au seuil .
- L'histogramme de la colonne soit supérieur au seuil .

Pour détecter la fin du caractère il faut que:

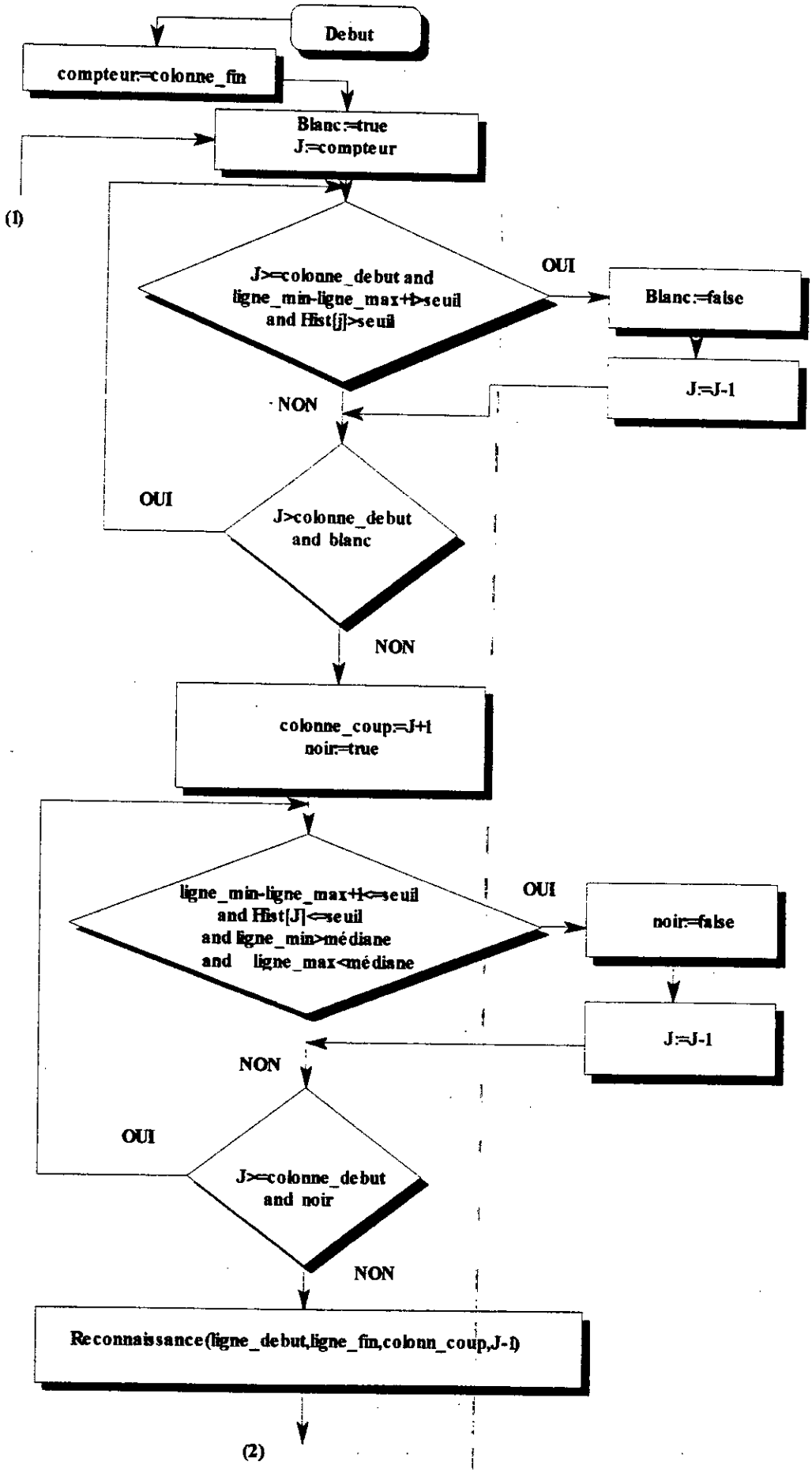
- La différence entre la ligne minimale et la ligne maximale soit égale au seuil .
- L'histogramme de la colonne soit égale au seuil .
- La ligne minimale supérieure à la ligne médiane .
- La ligne maximale inférieure à la ligne médiane.

La figure 2.15. représente l'organigramme de l'extraction des caractères.

Problèmes de la segmentation [Ham 96]

La séparation des caractères n'est pas toujours possible . Pour l'imprimé on pourrait croire qu'il suffit de couper à la rencontre d'un espace suffisant entre caractères . Seulement des accolements dus aux dispositifs d'écriture (mauvais réglage) ou à la fonte utilisée (écriture grasse , penchée) réduisent ces espaces et rendent cette tache délicate.

Un autre problème délicat est la segmentation des caractères tel que le cin , le kaf , le ba et le tha . Pour y remédier , il faut préalablement choisir des prototypes convenables lors de la phase d'apprentissage et prévoir une procédure de correction lors de l'affichage.


Debut
compteur:=colonne_fin
Blanc:=true
J:=compteur
(1)
J>=colonne_debut and
ligne_min-ligne_max+1>seuil
and Hist[j]>seuil
OUI
Blanc:=false
J:=J-1
NON
OUI
J>colonne_debut
and blanc
NON
colonne_coup:=J+1
noir:=true
ligne_min-ligne_max+1<=seuil
and Hist[J]<=seuil
and ligne_min>médiane
and ligne_max<médiane
OUI
noir:=false
J:=J-1
NON
OUI
J>=colonne_debut
and noir
NON
Reconnaissance(ligne_debut,ligne_fin,colonn_coup,J-1)
(2)

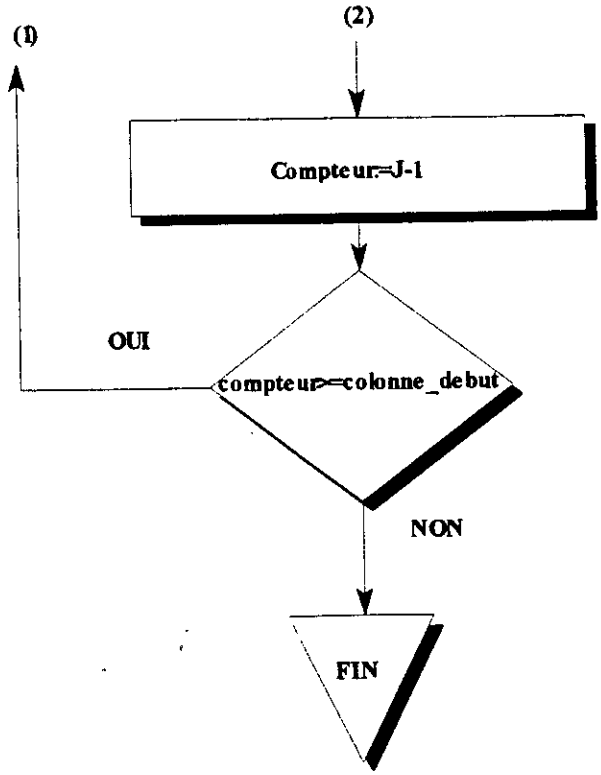


Figure 2.15. *Organigramme de l'extraction des caractères*

2.3.3 La Détection de contour

On défini le contour comme la limite entre deux régions possédant des niveaux de gris relativement distincts. Par la suit on considérera que ces régions considérées sont suffisamment homogènes pour que le contour puisse être déterminé en se basant seulement sur la discontinuité des niveaux de gris.

Le principe de la détection de contour est le calcul d'une dérivée locale : Le gradient. En effet , les contours se trouvent très souvent aux maxima du gradient des niveaux de gris. L'opération de détection de contour consiste alors à déterminer l'image gradient. Cette image est un vecteur dont la phase indique la direction de l'image dans laquelle le changement du niveau de gris est le plus important et dont le module mesure le taux de variation du niveau de gris.

l'opérateur gradient

Le gradient d'une image f(x, y) au point (x, y) est définit comme le vecteur bidimensionnel suivant :

$$\mathbf{G}[\, f(x, y)\,] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

f(x, y) représente la luminosité du point (x, y).

le vecteur **G** pointe en direction de la plus grande variation de f au point (x, y). Pour la détection de contours, on s'intéresse au module de ce vecteur, généralement appelé le *gradient* et noté G[f(x, y)] :

$$G[f(x, y)] = [\, G_x^2 + G_y^2 \,]^{1/2} \qquad (1)$$

La direction α du vecteur gradient est donnée par : $\alpha = \text{tang}^{-1}(G_x / G_y)$

Cet angle est mesuré par rapport à l'axe x.

Le premier opérateur de dérivation fut celui de *Roberts* en 1965. C'est une application directe de la formule de dérivée. Les dérivées partielles peuvent être approximées par les différences locales des intensités des points de l'image ce qui donne :

$$G_x = \frac{\partial f(x, j)}{\partial x} \approx f(x+1, y) - f(x, y)$$

$$G_y = \frac{\partial f(x, j)}{\partial y} \approx f(x, y+1) - f(x, y)$$

Ce qui revient à convoluer l'image avec les masques :

$M_x = (-1, +1)$ et $M_y = \begin{pmatrix} -1 \\ +1 \end{pmatrix}$. M_x sert à calculer G_x et M_y à calculer G_y.

On peut par la suite obtenir le module du gradient grâce à l'équation (1).

Le gradient s'interprète comme suit :

- Un gradient nul correspond à une zone homogène.
- Un gradient non nul correspond à une zone de transitions entre niveaux de gris (contour).

Il existe d'autres gradients qui sont les opérateurs de *Sobel* et *Prewitt*. Les masques correspondant à ces opérateurs sont données par :

$$M_x = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \qquad M_y = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

Masques de Sobel .

$$M_x = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \qquad M_y = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

Masques de Prewitt .

$$\mathbf{x} = \begin{pmatrix} (x-1,y-1) & (x,y-1) & (x+1,y-1) \\ (x-1,y) & (x,y) & (x+1,y) \\ (x-1,y+1) & (x,y+1) & (x+1,y+1) \end{pmatrix}$$

$\mathbf{x}$: les 8-voisins de (x,y).

Dans ce cas : $G_x = M_x \,.\, \mathbf{x}$ et $G_y = M_y \,.\, \mathbf{x}$

Ces opérateurs présentent l'avantage d'être mois sensibles aux bruits ; contrairement à l'opérateur de Roberts.

Pour diminuer le temps de calcul nécessaire pour obtenir l'image gradient, il est possible de définir l'amplitude du gradient par l'une des normes suivantes :

$A1 = \text{Max}(|G_x| , |G_y|)$

$A2 = |G_x| + |G_y|$

Malheureusement , ces deux normes introduisent des distorsions dans le calcul de l'amplitude. Pour éviter ces distorsions , on utilise des masques optimaux tels que l'opérateur de *Kirsh* qui associe un masque à chaque direction dans le voisinage du pixel considéré. Les huit masques de cet opérateur dans les huit directions 0°, 45°, 90°, 135°, 180°, 225°, 270° et 315° sont :

$$M_1 = \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \qquad M_2 = \begin{pmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{pmatrix} \qquad M_3 = \begin{pmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{pmatrix}$$

$$M_4 = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{pmatrix} \qquad M_5 = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{pmatrix} \qquad M_6 = \begin{pmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{pmatrix}$$

$$M_7 = \begin{pmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{pmatrix} \qquad M_8 = \begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}$$

Opérateurs de Kirsh .

Après le calcul des gradients suivant les huit directions , l'amplitude du gradient résultant est alors donnée pour chaque point par: $A = \text{Max}(G_i)$ avec i=1, ..., 8.

Une fois qu'on a choisi une méthode pour approximer le gradient, il existe plusieurs façons d'utiliser les résultats pour générer une image gradient g(x, y). L'approche la plus simple consiste à attribuer à g au point (x, y) le module du gradient de f au point (x, y) , c'est à dire :

$g(x, y) = G[f(x, y)]$.

Le principal inconvénient de cette méthode est que toutes les régions homogènes dans l'image apparaissent sombres dans l'image résultante g(x, y) à cause des faibles valeurs du gradient dans ces régions. Une solution à ce problème est de composer l'image résultante comme suit :

$$g(x, y) = \begin{cases} G[f(x,y)] & \text{si } G[f(x,y)] \geq S \\ f(x,y) & \text{sinon.} \end{cases}$$

où S est un seuil positif.

En faisant un bon choix de S , il est possible de garder les contours significatifs sans détruire les caractéristiques des régions homogènes. Une variation de cette approche est de fixer les contours à un niveau de gris déterminé :

$$g(x, y) = \begin{cases} Ng & \text{si } G[f(x,y)] \geq S \\ f(x,y) & \text{sinon.} \end{cases}$$

On veut quelques fois étudier la variation des niveaux de gris des contours sans interférences avec le fond de l'image . Ceci peut être accompli en formant l'image gradient suivante :

$$g(x, y) = \begin{cases} G[f(x,y)] & \text{si } G[f(x,y)] \geq S \\ Nf & \text{sinon.} \end{cases}$$

où Nf est un niveau de gris spécifié pour le fond de l'image.

Finalement, si on est intéressé seulement par la forme des contours , la relation suivante

$$g(x, y) = \begin{cases} Ng & \text{si } G[f(x,y)] \geq S \\ Nf & \text{sinon.} \end{cases}$$

donne une image gradient binaire où les contours et le fond de l'image sont représentés par deux niveaux de gris spécifiques.

Laplacien

Le Laplacien est la dérivée seconde définit par :

$L[f(x, y)] = (\partial^2 f / \partial x^2) + (\partial^2 f / \partial y^2)$.

Il existe plusieurs types de masques de Laplacien dont :

$$M_1 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \qquad M_2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \qquad M_3 = \begin{pmatrix} -1 & -2 & -1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

Ces opérateurs sont appliqués à une fenêtre de taille 3×3.

Les maxima du gradient correspondent aux passages par zéro des dérivées du deuxième ordre. On peut donc détecter les points du contour par le passage par zéro du Laplacien si le gradient en ces points est suffisamment grand .

Le principal inconvénient du Laplacien est sa grande sensibilité au bruit. C'est pour cela qu'en général il n'est pas utilisé pour la détection directe des contours. Mais il est utile lorsqu'on veut savoir si un pixel se trouve du coté sombre (l'objet) ou claire (fond de l'image) par rapport au contour. Si le pixel se trouve du coté sombre par rapport au contour alors le Laplacien est positif , sinon il est négatif.

2.4. Conclusion

Nous venons de voir dans ce chapitre les premières transformations nécessaires au traitement des formes (caractères). Bien qu'abordées souvent de manière détaillé , elle montrent les différentes aberrations du système d'acquisition contre lesquelles il faut se prémunir avant d'entreprendre toute action de reconnaissance. Actuellement , avec les progrès de l'électronique , la qualité des capteurs a beaucoup augmenté . Mais malgré cela , les systèmes commercialisés continuent à fournir des logiciels de filtrage permettant à l'utilisateur de corriger lui-même ses données et de les adapter à l'application choisie.

CHAPITRE III

RECONNAISSANCE

3.1 Présentation

La reconnaissance regroupe les deux tâches d'apprentissage et de décision qui jouent des rôles assez proches dans les systèmes OCR .

En effet , à partir de la même description de la forme en paramètres , elles tentent toutes les deux , d'attribuer cette forme à un modèle de référence .

Vu que les mêmes étapes seront suivies pour l'apprentissage et la décision , nous avons jugé judicieux d'expliquer au départ l'extraction des caractéristiques primaires et secondaires , puis les procédures d'apprentissage et de décision .

3.2 L'alphabet arabe

L'alphabet arabe comprend 28 caractères de base plus la Hamza qui a un rôle très important dans la phonétique , et le Lemalif qui est en réalité un caractère composé du Lem et du Alif.

Chaque caractère de l'alphabet arabe peut s'écrire sous quatre formes selon sa position dans le mot : début , milieu , fin , isolé .

Contrairement aux écritures latines ou bien aux caractères latins , l'arabe n'a pas de voyelles , mais plutôt des signes de ponctuation qui peuvent être placés au dessus ou bien au dessous du caractère.

La présence de groupes de points et leurs positions jouent un rôle prépondérant pour la discrimination des caractères possédant le même tracé et appartenant à la même classe. Ces groupes de points (1 à 3) peuvent avoir trois positions possibles : au dessus , au milieu ou au dessous du corps principal.

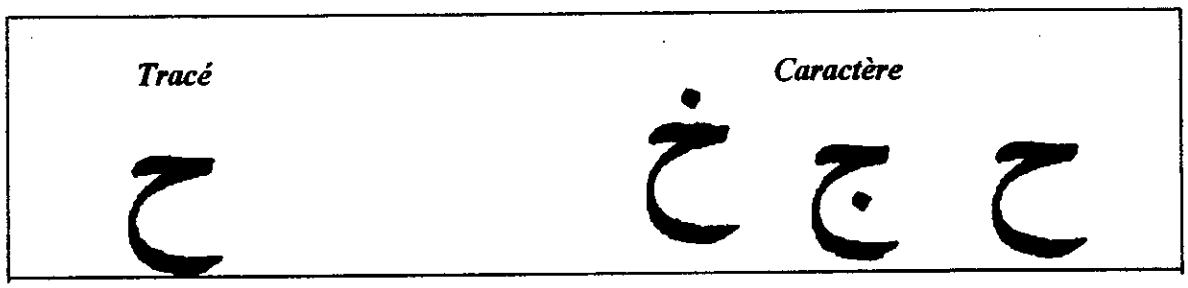


Figure 3.1 : *Caractères ayant même tracé principal.*

3.3 primitives principales

3.3.1 Choix et extraction

Il est très aisé de remarquer que les caractères arabes sont un agencement de boucles et de concavités , nous avons donc jugé intéressant de classer les caractères en familles ou classes selon ces primitives .

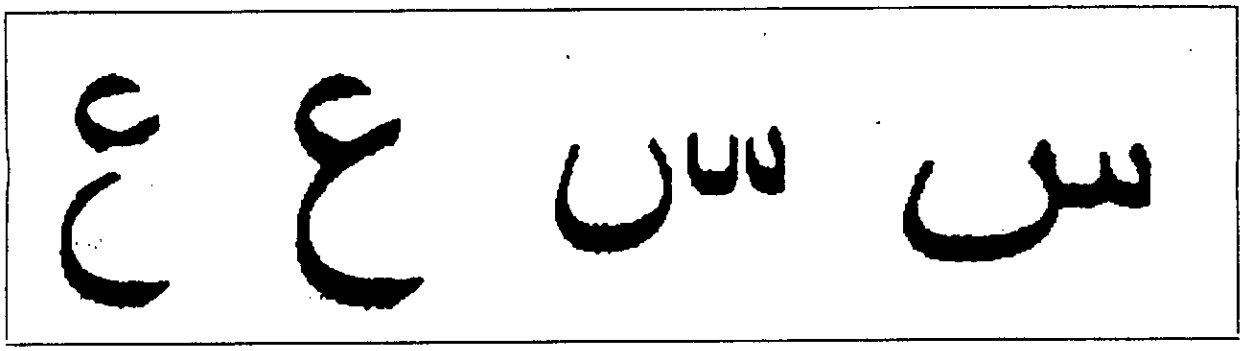


Figure 3.2 : *En général les caractères arabes sont formés d'un agacement de boucles et de concavités*

Principe de la méthode d'extraction

L'étape de segmentation permet de récupérer les coordonnées de chaque caractère composant l'image . Ceci étant fait , on charge le caractère dans une matrice . La première étape d'extraction consiste à cadrer le caractère (trouver le plus petit rectangle qui le contient) ; on procède ensuite à un 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) , car l'existence de quatre transitions ou plus signifie qu'on a croisé le caractère deux fois ou plus , donc il y a une possibilité d'existence d'une concavité ou d'une boucle (trou) . Voir figure 3.3.

Si le caractère est bien cadré alors le nombre de transitions de chaque ligne (respectivement chaque 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

au balayage et on cherche le centre des segments [T2n , T(2n+1)] tel que Ti représente la $i^{ème}$ transition . Voir figure 3.3

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

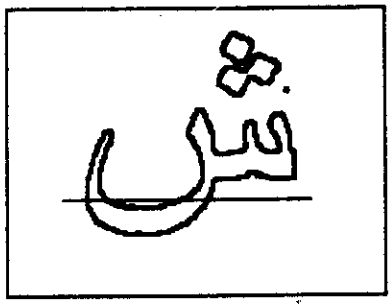


Figure 3.3: *Lignes de transitions et points essentiels*

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 concavités 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 à ce point et on décide a quel type de concavités appartient ce point.

Exemple :

- Si on croise le caractère dans tous les sens alors le point considéré appartient à une boucle.
- Si on croise le caractère dans le sens : bas , bas - droit , bas - gauche alors on est en présence d'une concavité vers le haut .Voir figure 3.4 .

Figure 3.4 *concavité vers le haut*

Cette méthode va nous donner le nombre de concavités dans les quatre 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 de boucles.

Dans cette étape on essaye de faire une liaison entre chaque deux points essentiels , s'ils sont liés (absence de tout point allumé appartenant au caractère entre les deux points) alors ils appartiennent à la même concavité ; et si l'un d'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 détectée est une fausse boucle .

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.

Figure 3.5. *Liaison entre points essentiels*

Cette méthode a donner de bons résultats et s'est avérée efficace , néanmoins elle donne parfois un résultat erroné , et cela est dû non pas à la méthode mais à la forme du caractère . On trouve soit :

1 - De fausses boucles qui sont difficiles à éliminer.
2 - Des concavités qui ne sont pas bien séparées.
3 - La détection de très faibles concavités.

3.3.2 Classification des caractères

Après extraction des primitives principales , à savoir les boucles et les concavités dans les quatre directions citées , nous allons grouper tous les caractères ayant des caractéristiques similaires dans une même classe calculée de la manière suivante :

$$Classe = 4^0 * Nb\_tr + 4^1 * Nb\_cg + 4^2 * Nb\_cd + 4^3 * Nb\_ch + 4^4 * Nb\_cb.$$

Nb_tr : nombre de trous.
Nb_cg : nombre de concavités gauches.
Nb_cd : nombre de concavités droites.
Nb_ch : nombre de concavités hautes.
Nb_cb : nombre de concavités basses.

Cette classification a été utilisée par *M.B. EL KURDY*. Elle permet de ne pas rencontrer un caractère dont les caractéristiques ne sont pas identiques avec celles des autres caractères contenus dans la même classe.

Pour les caractères arabes , les concavités basses sont négligeables ou inexistentes , donc le dernier terme (Nb_cb) est toujours nul , mais il est intéressant de le garder pour les systèmes OCR bilingues.

3.4 Primitives secondaires[Zek 95]

Ce sont les caractéristiques secondaires qui permettent de distinguer les caractères d'une même classe , il est donc très important de bien les définir.
Nous avons choisi cinq caractéristiques secondaires :

- ***La forme du caractère*** : Cette caractéristique est définie par le rapport:

Rapport 1 = W/H ou W et H sont respectivement la largeur et la hauteur du caractère .
La variable forme liée à la forme du caractère est définie comme suit:

$$\begin{cases} forme = 2 \quad si \quad Seuil1 < rapport1 < Seuil2 \\ forme = 1 \quad si \quad rapport1 > Seui2 \\ forme = 3 \quad si \quad rapport1 < Seuil1 \end{cases}$$

où le Seuil1=0.9 et Seuil 2=1.1

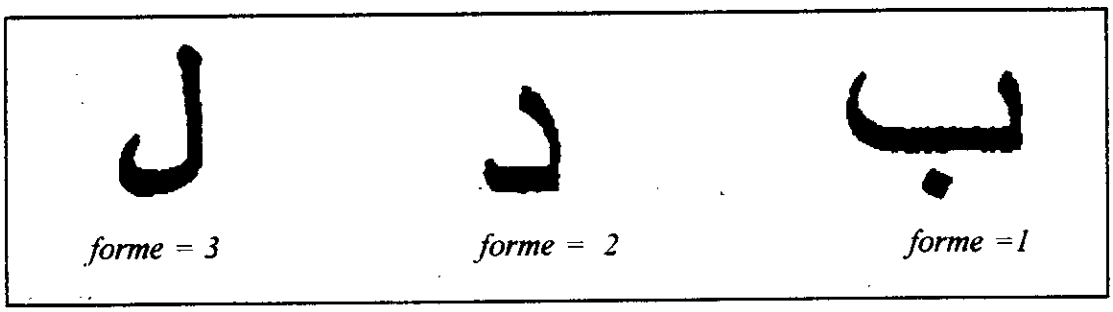


Figure 3.6 *La forme du caractère*

- *Taux de remplissage des quatre coins*

cette caractéristique donne une idée sur la distribution du caractère.

La variable à calculer est appelée *Corvar* . Sachant que W et H sont respectivement la largeur et la hauteur du caractère , on considère quatre carrés de dimensions :

(W / 4) * (W / 4) si le caractère est debout.

(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 . Le taux de de remplissage d'un coin est définit comme étant le rapport entre le nombre de pixels noirs sur le nombre total de pixels . Un coin est estimé plein (taux=1) si le taux de remplissage est supérieur à 0.5 .

On calcul la variable *Corvar* de la façon suivante :

*Corvar= Taux (Haut - Gauche) + 2 * Taux (Haut - Droit) + 4 * Taux (Bas - Droit) + 8 * Taux (bas - Gauche).*

Il est à noter que les valeurs de cette variable seront comprises entre 0 et 15 .

Si l'on observe les caractères arabes , il nous apparaît de prime abord l'existence de deux types ou deux catégories de caractères . La première catégorie regroupe les caractères portant des

points y compris la hamza , et la seconde regroupe ceux qui ne portent pas de points . De là il serait très judicieux d'exploiter cette donnée pour identifier les caractères.

Les trois dernieres caractéristiques secondaires , à savoir : l'existence , l'emplacement et le nombre de points sont donc très liées.

- ***Existence des points ou la hamza***

Cette étape très délicate ne permet aucune erreur ; l'identification du caractère en est dépendante et la moindre erreur dans cette étape compromet l'extraction des autres critères .

Le principe de l'opération consiste en la détermination du nombre d'entités qui composent le caractère et ce grâce à un balayage de la surface et un calcul du nombre de transitions .

La variable *expoint* liée à cette caractéristique est définie comme suit :

expoint = 1 si le caractère contient des points

expoint = 0 si le caractère ne contient pas de points

- ***Emplacement des points ou la hamza***

Pour définir l'emplacement des points , on se base sur le calcul des surfaces des deux entités .

L'évidence est que les points correspondent à la petite surface , donc , en comparant les surfaces entre elles , on peut conclure l'emplacement des points (bas ou haut) .

La variable *pospoint* liée à la position est definie comme suit :

pospoint =1 si le point est situé au dessus du caractère

pospoint =2 si le point est situé dans ou au milieu du caractère

pospoint =3 si le point est situé au dessous du caractère

- ***Nombre de points***

Cette étape est réalisée dans le cas où :

expoint = 1

La variable *nbpoint* liée à cette caractéristique peut prendre les valeurs 1 , 2 , 3 ou 4.

Cas de deux points :

Les deux points sont faciles à détecter . Pour le faire on procède comme suit :

On définit le rapport suivant :

$$Rapport\ 3 = Wp / Hp.$$

Avec Wp , Hp la largeur et la hauteur respectives des points :

On définit aussi la variable MaxTr comme suit :

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

Si (Rapport 3 >= 1.1) et (MaxTr = 4) alors n*bPoint = 2.*

Cas de trois points :

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

On calcul aussi la surface pleine *SurfPl* qu'occupent les points .Aprés cadraga des points , on définit la surface *Surf* qui représente la surface du rectangle qui contient les points.

On définit les rapports suivants :

Rapport 4 = SurfB / SurfH.

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

Rapport 5 = SurfPl / Surf.

Ce rapport nous donne le taux de remplissage du rectangle déjà définit.

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 (Rapport 4 >= 1.7) et (Rapport 5 <= 0.6) Alors *:nbPoint = 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) qui est élevé , donc on calcul ce nombre noté NombTr.

Si (0.9 <= Rapport 3 <= 1.1) et (NombTr <= 4) Alors ***:nbpoint = 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 <= Rpport3 <= 1.1) et (NombTr = 2) Alors : *nbpoint = 1.*

Toutes ces caractéristiques secondaires seront ensuite chargées dans un vecteur V[i] de huit octets nommé *Attribut* dans le programme de l'application .

3.5 La procédure de Décision

3.5.1 Distances utilisées

Nous avons utilisé deux distances **Dis1** et **Dis2** tel que :
La distance Dis1 est calculée entre les Formes et la distance Dis2 est calculée entre les corvar (taux de remplissage des quatre coins). C'est à dire qu'on a calculé les distances entre les deux premières caractéristiques secondaires. Les trois dernières caractéristique secondaires à savoir : expoint , pospoint , nbpoint doivent être identiques.

- ***Distance Dis1 :***

La première caractéristique secondaire est la forme du caractère . 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 .On peut donc en conclure 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 définit Dis1 comme suit :

$$Dis1 = abs (Vx[1] - Vp[1]).$$

Avec :

Vx[i] : élément d'ordre (i) du vecteur de caractéristiques du caractère x à 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 : $Dis1 <= 1.$

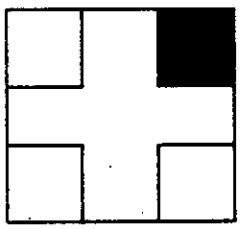
- ***Distance Dis2***

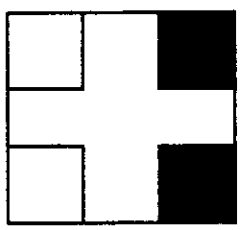
Dans ce cas on a affaire à une distance binaire . Cette distance est calculée de la façon suivante :

$$A = (Vx[2]) \; XOR \; (Vp[2])$$

pour mieux saisir nous proposons l'exemple suivant:

soit les deux taux de remplissage de la figure ci dessous:





A= (0100) XOR (0110)

3.5.2 Algorithme de reconnaissance

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

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 prototype . On aura :

CAi[j] : jéme caractéristique du iéme prototype.

Avec: :j = (1 , 2 , 3 , 4 , 5).

i = 1...Nb_pr.

Nb_pr : le nombre de prototypes de la classe.

On cherche les prototypes qui ont les même 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..Nb_pr.

CAx[j] : jéme caractéristique du caractère à reconnaître .

CAi[j] :jéme caractéristique du iéme prototype.

Les prototypes qui vérifient $Dis[i] = 0$ sont mis dans une liste appelée *Liste1* , qui contient k prototypes.

Si k = 1 le caractère est identifié.

Si k > 1 alors :

Si tous les prototypes décrivent le même caractère , c'est à dire qu'ils ont le même nom , alors le caractère est identifié, si non le caractère est ambigu.

Si k = 0 , alors on met dans une deuxième liste appelée *Liste2* les prototypes qui vérifient :

$$D = 0.$$

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 alors le caractère est rejeté.

Si k1 = 1 alors on calcule les distances Dis1 et Dis2.

Si (Dis1<=1) et (Dis2<=1) alors le caractère est reconnu.

Si non il est rejeté.

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

(Dis1 <= 1) et (Dis2 <= 1)

On obtient k2 caractères.

Si k2 = 1 alors le caractère est reconnu.

Si k2 = 0 le caractère est rejeté.

Si k2 > 1 on calcule la fréquence d'occurrence de chaque caractère .

S'il y a un seul prototype qui a une fréquence d'occurrence la plus élevée alors :

le caractère est reconnu.

Si non il y a ambiguïté.

3.6 La procédure d'apprentissage

Dans tous les systèmes de reconnaissance automatique de l'écriture , l'étape finale est l'identification et la décision . Cette étape consiste à comparer les caractères inconnus avec ceux qui sont stockés en mémoire sous forme de prototypes ou caractères de référence.

Etant donné que l'apprentissage consiste en la répartition en classes d'un grand nombre de caractères prototypes, au départ , le module de la reconnaissance ne reconnaît rien car le dictionnaire est vide. Mais au fur et à mesure que l'on entre les caractères avec leurs caractéristiques ,son vocabulaire s'enrichit et la reconnaissance devient accessible.

Une des flexibilites de ce logiciel est la possibilite de créer un dictionnaire pour chaque fonte durant l'execution .

3.6.1 Le dictionnaire

Le dictionnaire utilisé est un fichier résidant dans le disque . Il contient les caractères prototypes de référence .Sa composition est la suivante.

1 - L'entête :

Elle est formée de groupes de quatre bytes tels que :

- Le premier byte contient le numéro de la classe.
- Le deuxième byte contient le nombre de prototypes appelé Nb_pr que contient cette classe.

- Les troisième et quatrième bytes contiennent respectivement les poids faible et fort du pointeur vers le début de la classe considérée dans le fichier et cela à partir du début du dictionnaire.

2 - Les classes :

Cette partie contient les classes tel que chaque prototype est décrit sur huit bytes qui contiennent les éléments du vecteur de caractéristique V[i] :

- Le 1er byte contient la variable Forme.
- Le 2ème byte contient la variable corvar ou le taux de remplissage des quatre coins.
- Le 3ème byte contient la variable expoint.
- Le 4ème byte contient la variable pospoint.
- Le 5ème byte contient la variable nbpoint .
- Le 6ème byte contient la variable Nom c'est à dire le nom du caractère.
- Les 7ème et 8ème bytes 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 est lui aussi est déterminé.

Le dictionnaire utilisé peut contenir au maximum 80 classes , et chacune d'elles peut contenir au maximum 60 prototypes , alors on aura :

L'entête de taille : 4 x 80 = 320 bytes.

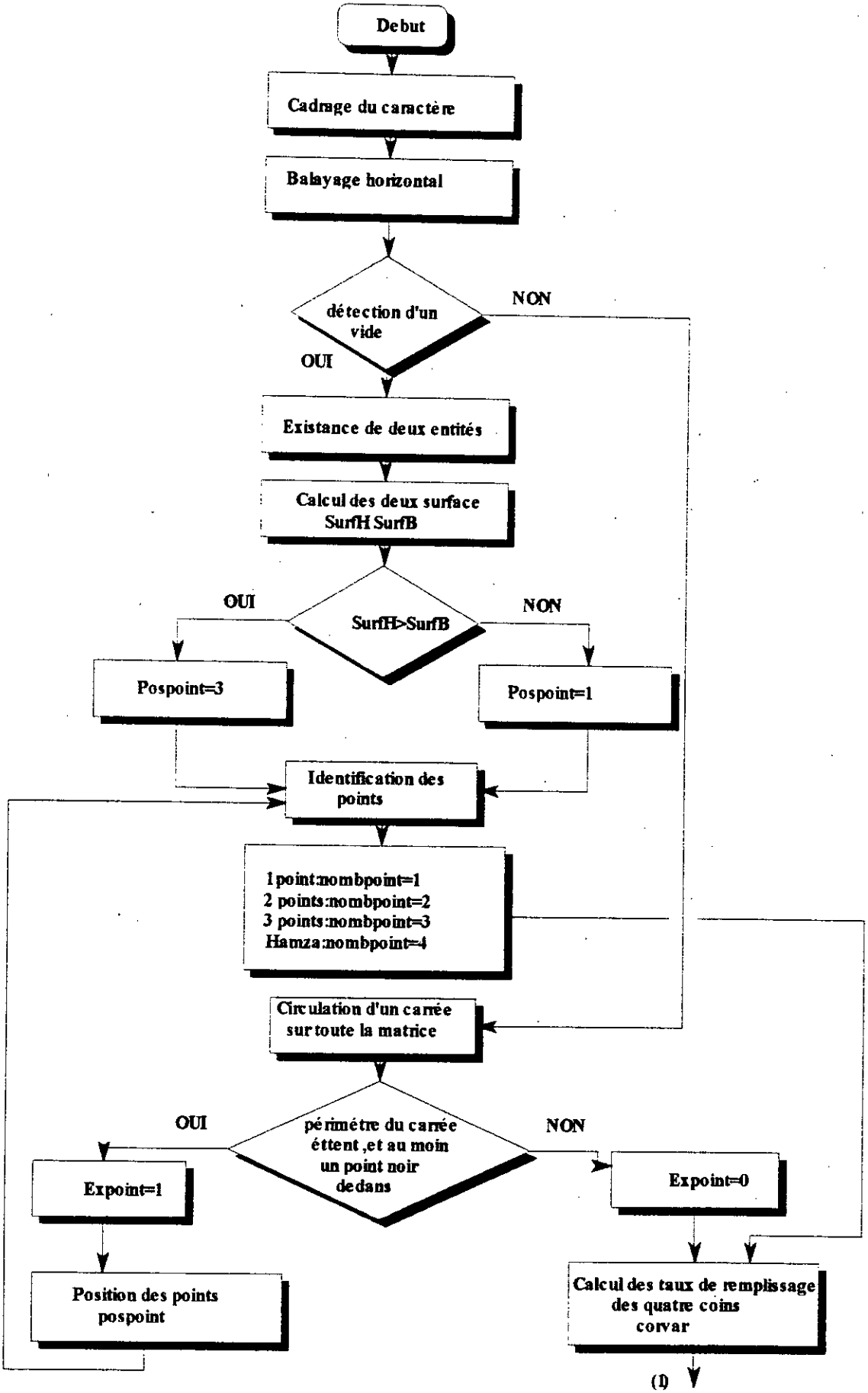
Le corps de taille : 8 x 60 x 80 = 38400 bytes.

Donc la taille du dictionnaire sera : 38720 bytes.

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

3.7 Organigramme de l'étape de reconnaissance

nous avons jugé utile de donner l'organigramme de toutes les opérations de l'étape de la reconnaissance


Debut
Cadrage du caractère
Balayage horizontal
détection d'un vide
NON
OUI
Existance de deux entités
Calcul des deux surface SurfH SurfB
SurfH>SurfB
OUI
NON
Pospoint=3
Pospoint=1
Identification des points
1 point:nombpoint=1
2 points:nombpoint=2
3 points:nombpoint=3
Hamza:nombpoint=4
Circulation d'un carrée sur toute la matrice
périmètre du carrée éttent ,et au moin un point noir dedans
OUI
NON
Expoint=1
Expoint=0
Position des points pospoint
Calcul des taux de remplissage des quatre coins corvar
(1)

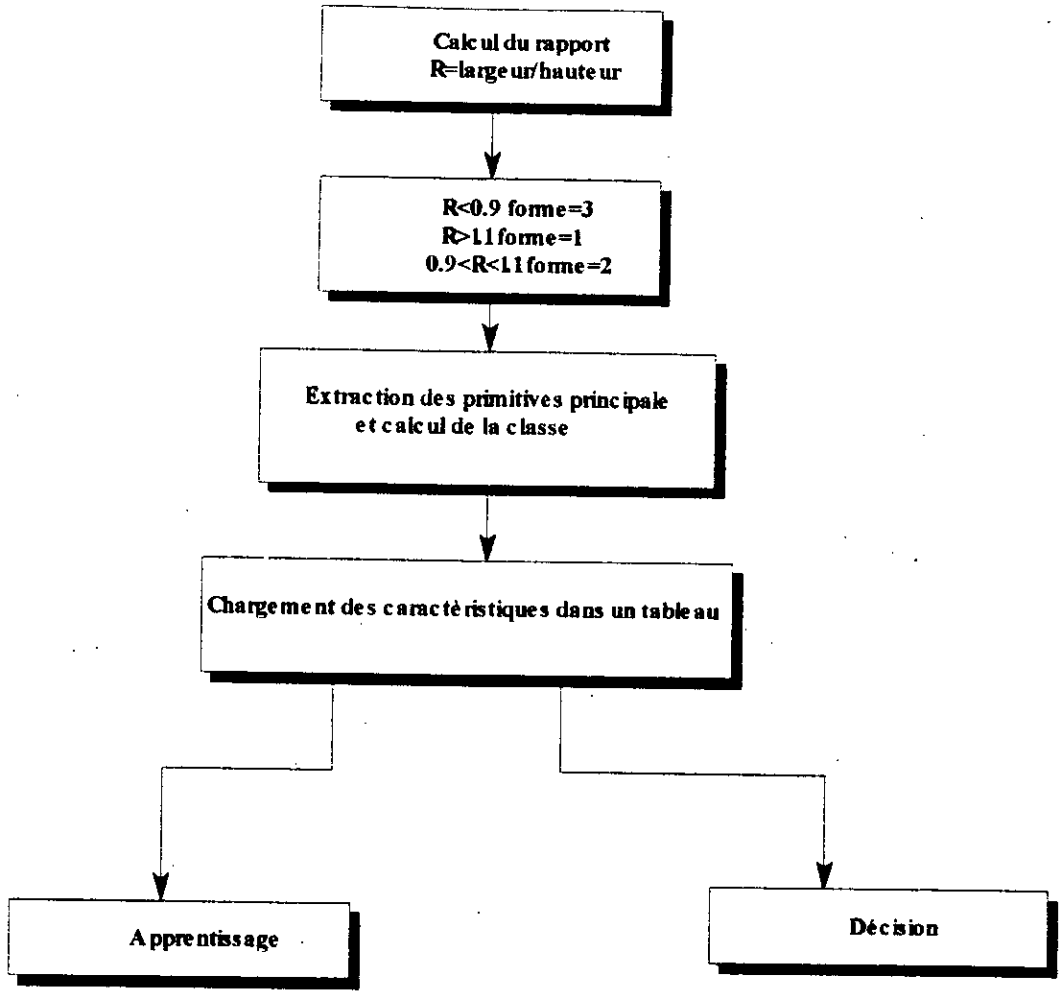


Figure 3.6 *Organigramme de la reconnaissance*

3.8 conclusion

Il faut noter les problèmes rencontrés surtout pour la détermination des seuils , car les formes du caractère différent d'un style à un autre et d'une taille à une autre et quelque fois lorsque le caractère recouvre les points il est difficile de séparer ces entités l'une de l'autre.

CHAPITRE IV

APPLICATIONS

4.1 Présentation

Dans cette partie on présente l'application des différentes méthodes de prétraitement d'image qui ont été vues précédemment en plus de l'apprentissage et la reconnaissance des caractères. Pour cela on a développé un programme qui réalise les opérations suivantes: Le filtrage , la binarisation , le calcul et l'égalisation d'histogramme , la segmentation , et en fin l'apprentissage et la reconnaissance .

On proposent aussi des outils de traitement : Détection de contour et squelittisation qui peuvent servir comme des éléments de départ pour une future conception d'OCR à base d'autre méthodes de reconnaissance.

4.2 Description du programme

Le programme réalisé permet de traiter les fichiers .TIF et BMP. Il a été développé à l'aide du logiciel Borland Pascal 7.0 sous Windows . Ce programme s'exécute sous Windows ce qui permet de profiter des avantages de cet environnement : possibilité de réduire le programme en icône , utilisation des menus , des boites de dialogues et de la souris (ce qui rend très facile la manipulation) et possibilité d'exécuter d'autres applications Windows en même temps. Le programme réalisé permet de lire les fichiers TIFF à 2 ,16 ou 256 niveaux de gris, ainsi que les fichiers BMP à 16 ou 256 couleurs. Ces images sont affichées dans la fenêtre principale de l'application . On donne ci dessous le résultat de l'exécution du programme.

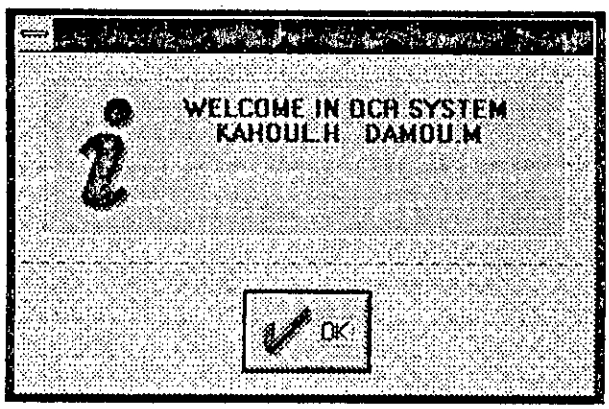


Figure 4.1. *Résultat de l'execution du programme*

Un simple clic sur le bouton OK fait apparaître la fenêtre principale de l'application

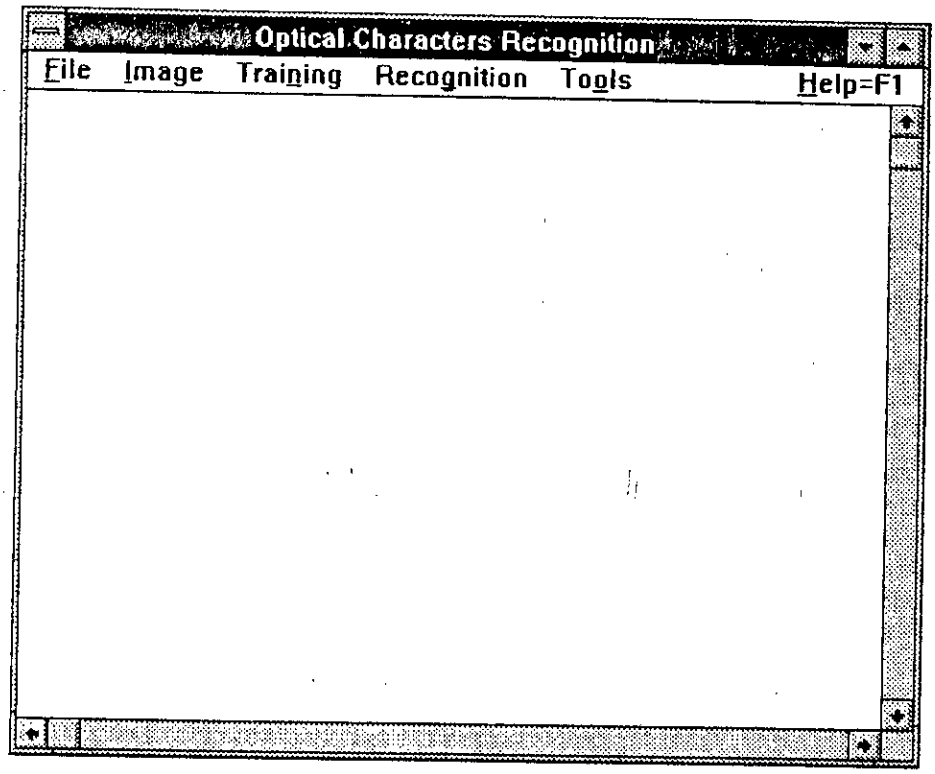


Figure 4.2. *Fenêtre principale*

L'aspect du programme rappelle celui des utilitaires de windows : le haut de la fenêtre contient la barre de titre (*Optical Characters Recognition)* et les cases du menu système , de réduction et d'agrandissement .Vous trouvez ensuite la barre des menus et , en bas et à droite , des barres de défilement permettant de déplacer la portion de l'image présentée dans la fenêtre.

4.2.1 Le menu file

Figure 4.3. *Articles du menu file*

Open: l'exécution de cette commande provoque l'affichage de la boite de chargement d'un fichier comme représenté sur la figure 4.4. Elle permet de charger une image sauvée sur disque dans la fenêtre principale.

La boîte de liste précédée de l'intitulé *Nom de fichier* permet d'introduire le nom du fichier à charger. La boîte de liste de droite permet de choisir le chemin d'accès du fichier : unité et répertoire . En bas la boite de liste permet de choisir le type du fichier TIFF ou BMP . Lorsque le nom du fichier est affiché dans la boîte de saisie , vous pouvez appuyer sur le bouton OK pour valider le chargement du fichier dans la fenêtre principale . Le bouton action *Annuler* permet d'annuler l'opération en cours.

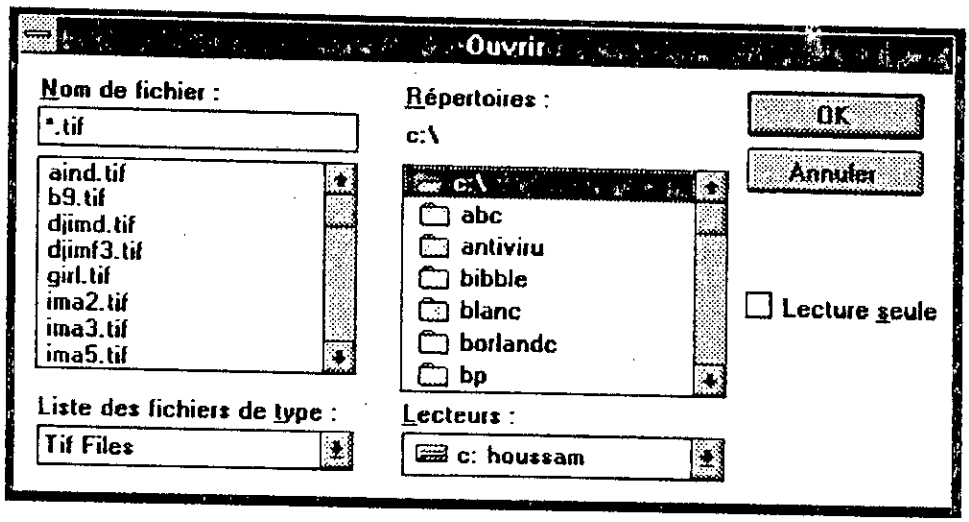


Figure 4.4. *Boite de sélection du fichier à visualiser*

Image info: Quand vous travaillez avec une image , vous pouvez voir ces informations , tel que le nombre de bits par pixel et dimensions.

L'exécution de cette commande provoque l'affichage de la boîte de dialogue ci-dessous.

Choisissez "OK" pour sortir et retourner à la fenêtre de l'image active.

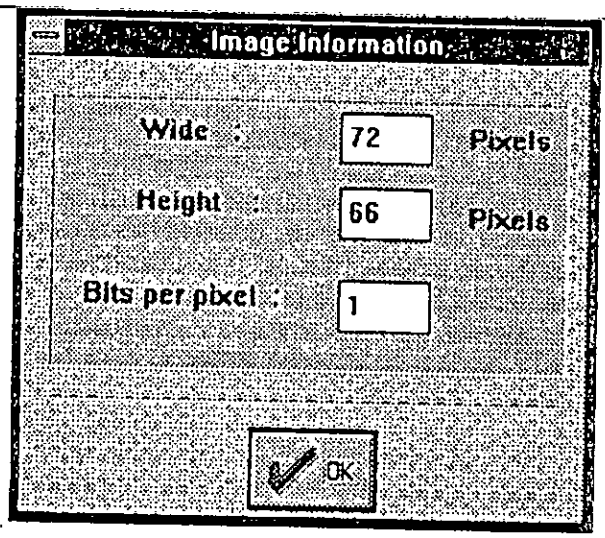


Figure 4.5. *Attribut de l'image*

Clear : Cette commande permet la suppression de l'image active.

exit (raccourci ALT-F4) : Cette commande permet de quitter l'OCR . La boîte de message ci-dessous apparaît . Un clic sur YES provoque la sortie. Le bouton action NO provoque l'annulation de la commande quitter.

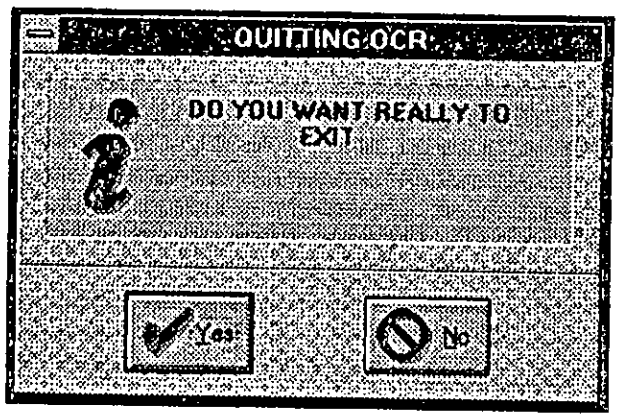


Figure 4.6. *boite de message surgissant lors de la demande de sortie*

4.2.2 Le menu Image

Le menu image contient les articles de prétraitement tel que :La binarisation , le filtrage et la segmentation.

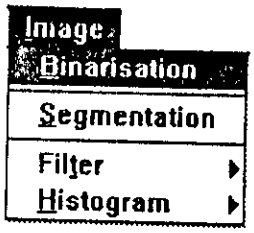


Figure 4.7. *articles du menu image*

Binarisation: Cette commande permet de transformer une image en couleurs (16 ou 256 niveaux de gris) en une image binaire en choisissant convenablement deux seuils de binarisation.

En sélectionnant cet article , une boite de dialogue apparaît . L'utilisateur doit choisir les deux niveaux de binarisation .

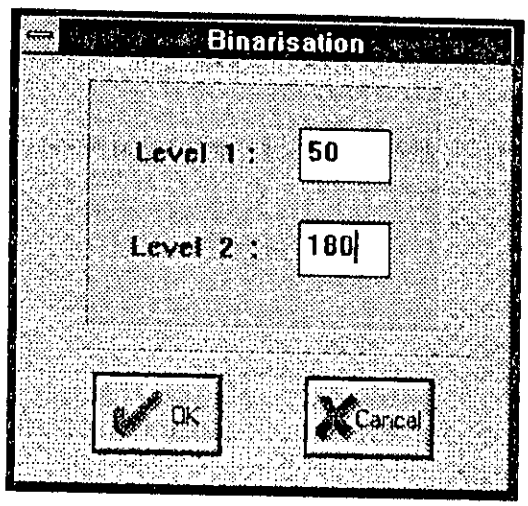


Figure 4.8. *boite de dialogue de la binarisation*

Segmentation: Dans cette partie, on propose à l'utilisateur un outil lui permettant de visualiser le résultat de la segmentation du texte , d'où il peut prévoir les erreurs qui peuvent surgir lors de la reconnaissance. L'image segmentée apparaît au dessous de l'image originale.

Filter: Ce popup contient les articles de filtrage . On laisse le soin à l'utilisateur de choisir le type du filtre qui convient le mieux pour l'image. On propose deux filtre: médian et moyen.

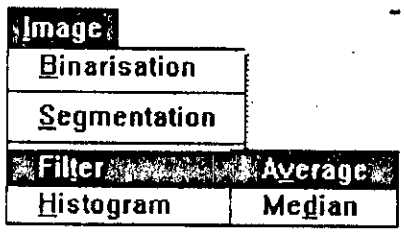


Figure 4.9. *articles du sous menu filter*

Histogram: Ce sous menu contient deux commandes : Le calcul d'histogramme et l'égalisation . Le calcul d'histogramme facilitera à l'utilisateur le choix des seuils de binarisation.

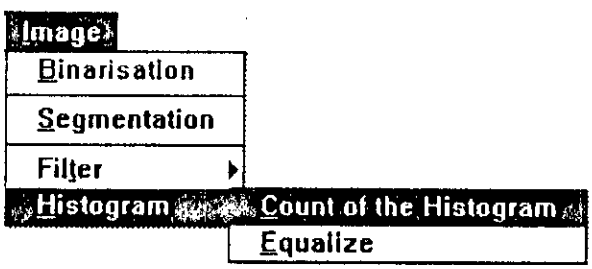


Figure 4.10. *articles du sous menu histogram*

4.2.3 Le menu training

Figure 4.11 *Menu training*

Comme son nom l'indique , ce menu contient la commande d'apprentissage :***Characters training*** Pour actionner cet article il faudrait au préalable charger un caractère dans la fenêtre principale. Lors de l'exécution , la boite de dialogue de la figure 4.12 surgit . Elle donne les caractéristiques du prototype . Un clic sur le bouton OK fait apparaître un messagebox indiquant la création d'une nouvelle classe suivi d'une deuxième boite de dialogue (figure 4.13) demandant à l'utilisateur d'introduire le nom du caractère

prototype . Le logiciel compare les caractéristiques et le nom avec la liste des caractères existant dans le dictionnaire . S'il trouve le même prototype , il renvoi le message de la figure 4.14

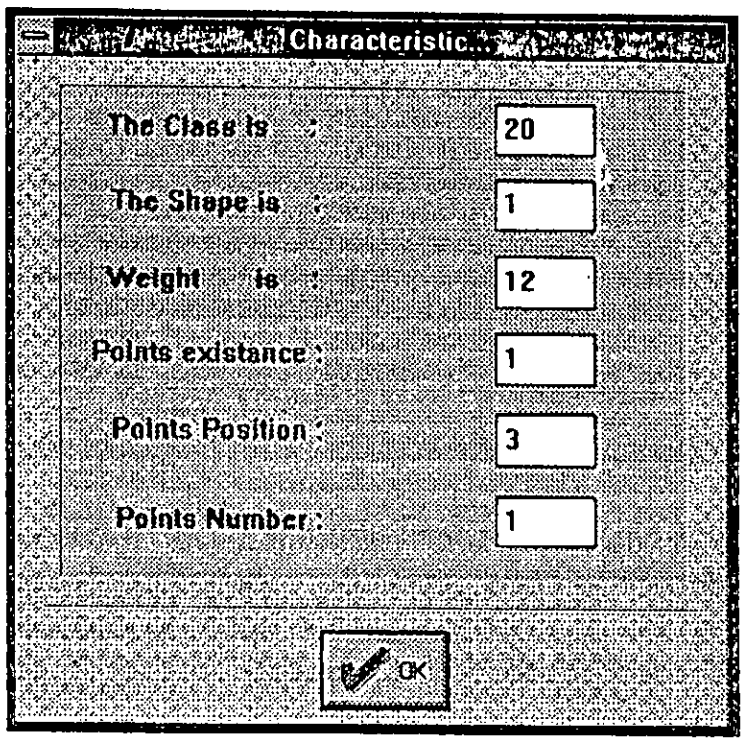


Figure 4.12. *boite de dialogue donnant les caractéristiques du prototype*

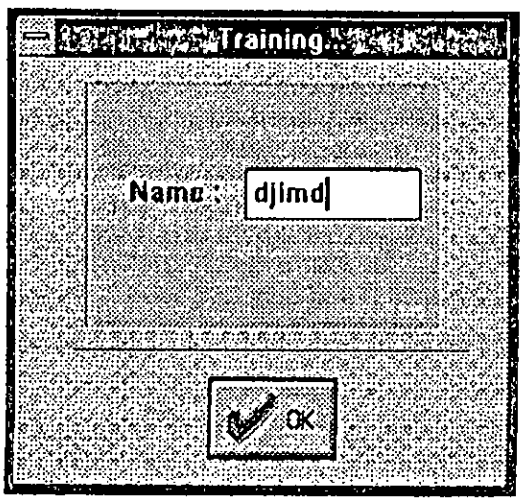


Figure 4.13. *Nom du prototype*

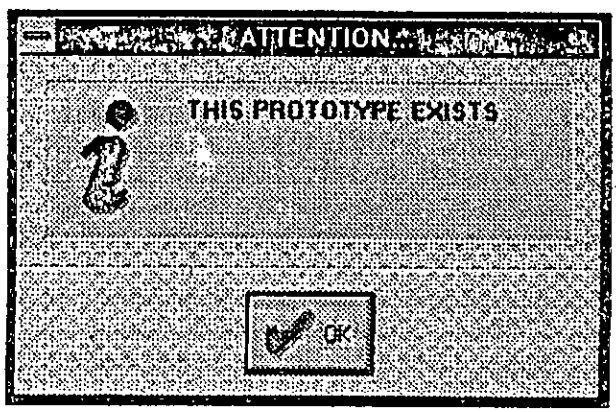


Figure 4.14. *Cas où le prototype existe*

4.2.4 Le menu recognition

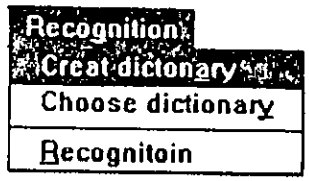


Figure 4.15. *Menu recognition*

Creat dictionary: permet de créer un dictionnaire.

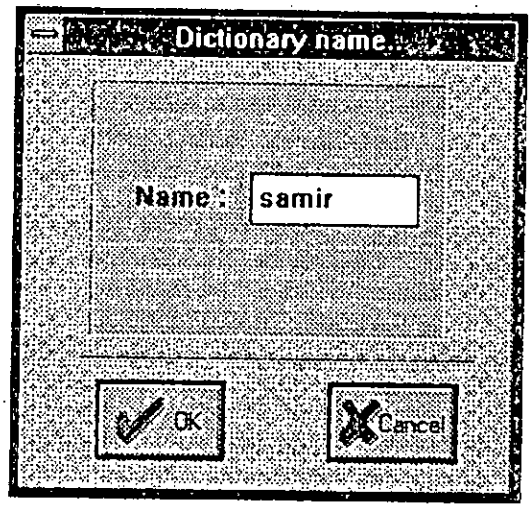


Figure 4.16. *Boîte de dialogue permettant la création du dictionnaire*

Choose dictionary: Cette commande offre la possibilité à l'utilisateur de choisir le dictionnaire de travail.

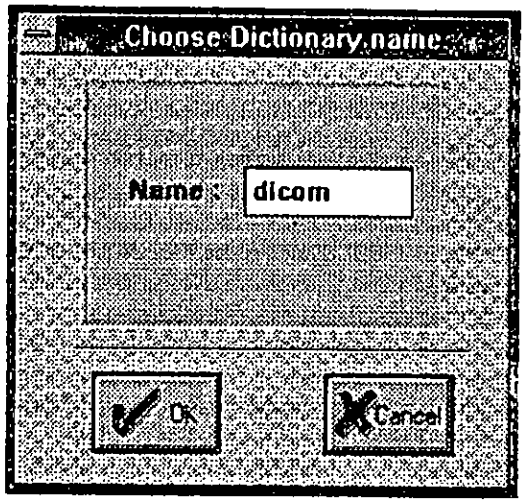


Figure 4.17. *Boîte de dialogue permettant la sélection du dictionnaire*

Recognition: Cette commande n'est active que si le texte est affiché dans la fenêtre principale. Lors de l'exécution , le logiciel affiche une succession de boîtes de message donnant le numéro du caractère si le caractère est reconnu suivi du temps de reconnaissance.

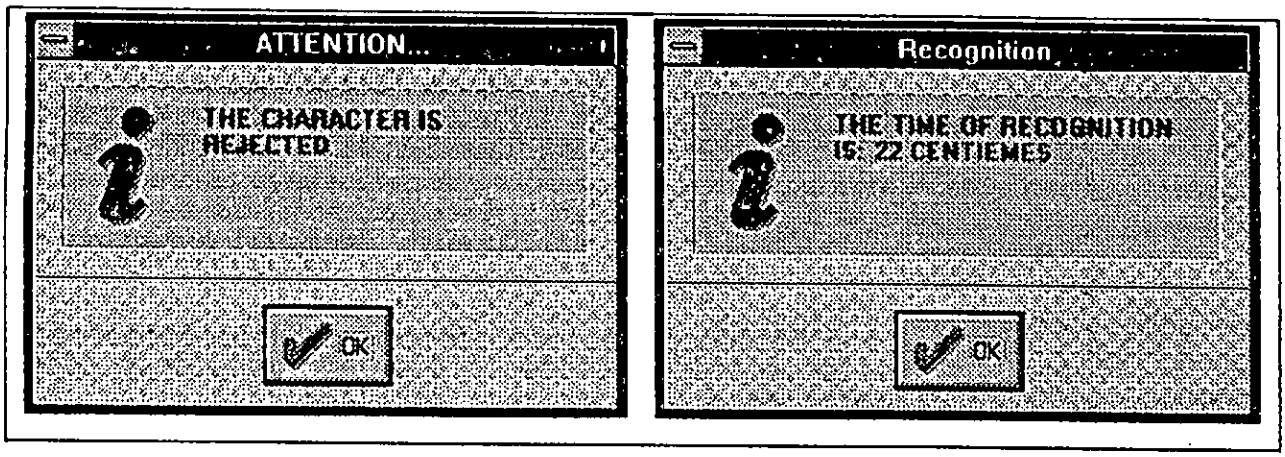


Figure 4.18. *boîtes de message surgissant lors de la sélection de l'article Recognition*

4.2.5 Le menu help

Cette partie donne à l'utilisateur une aide quant à l'utilisation du logiciel . Elle contient les quatre articles suivants:

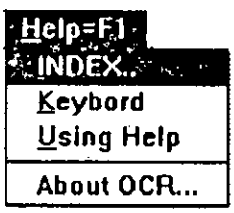


Figure 4.19. *Menu help*

Index : Il contient la liste de tous les sujets d'aide . L'utilisateur doit appuyer sur la touche F1 pour le consulter .

Keybord: *C*est une aide fournie par le logiciel sur l'utilisation des touches de clavier .Un simple exemple : l'appui sur les touches SHIFT et F1 permet de changer le curseur standard de windows en un curseur d'aide . L'utilisateur n'a qu'à cliquer sur l'article pour consulter son aide.

Figure 4.20. *Curseur d'aide*

Using help : Windows propose une aide sur l'aide qui peut être incluse sur l'ensemble des applications windows . ,Ce fichier ce nomme *Winhelp.hlp* . Nous avons jugé utile de l'intégrer sous la commande using help.

About OCR :L'exécution de cette commande provoque l'affichage d'une boîte de dialogue contenant des informations sur le logiciel.(voir la figure ci-dessous)

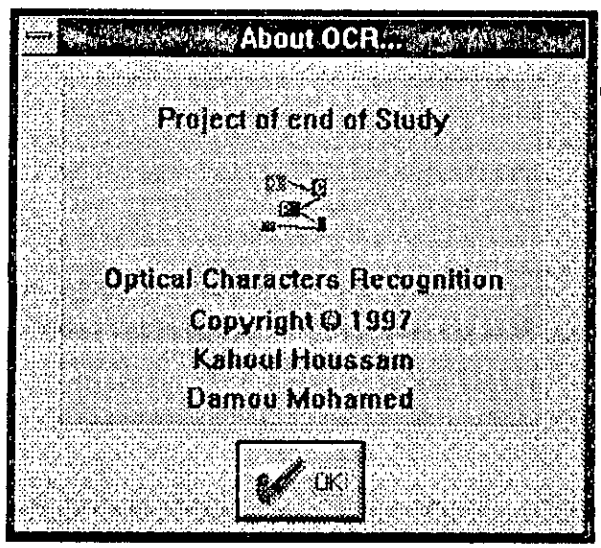


Figure 4.21. *Boîte d'informations*

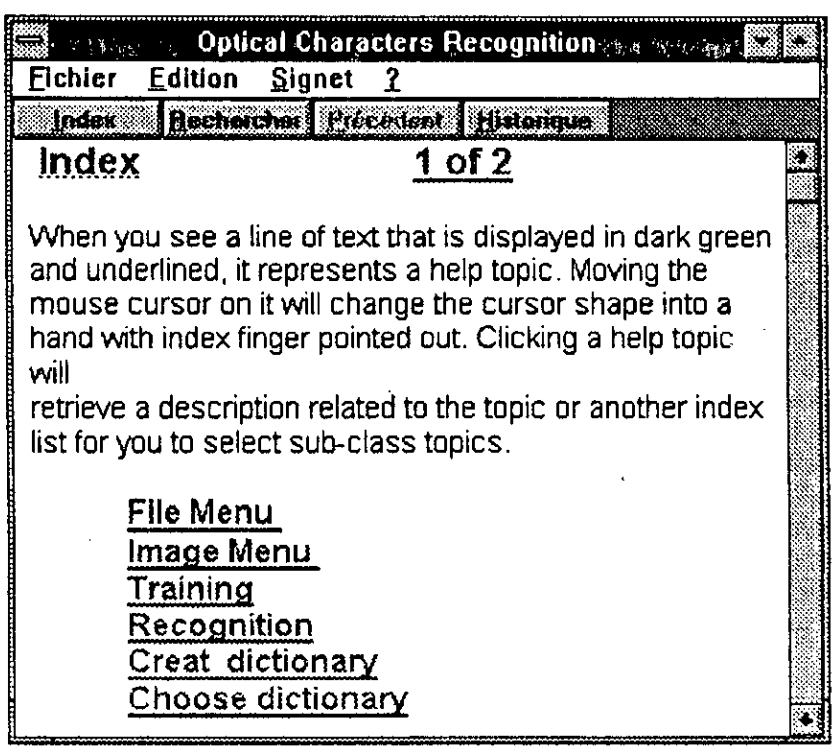


Figure 4.22. *L'index*

4.3 Exemples d'application

Comme premier exemple on prendra une image à deux niveaux de gris (noir et blanc) . La première étape consiste à débarrasser les données image du bruit d'acquisition . La figure 4.23. présente l'image originale .

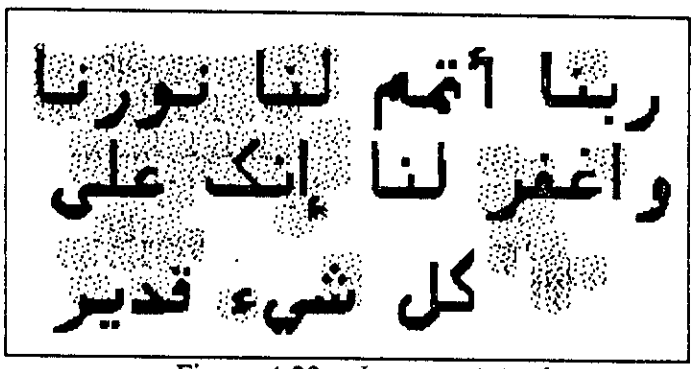


Figure 4.23. *Image originale*

le résultat du filtrage médian :

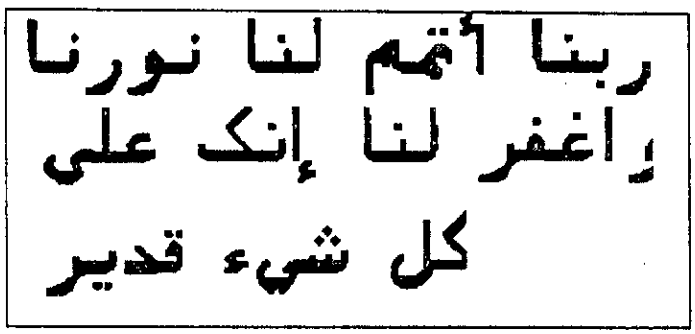


Figure 4.24. *image filtrée*

La deuxième étape consiste à segmenter le texte issu de l'opération de filtrage

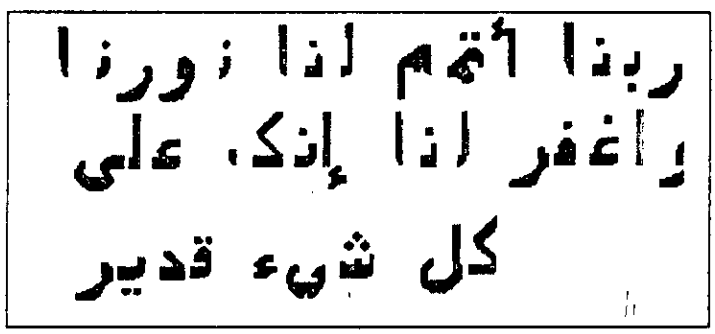


Figure 4.25. *Image segmentée*

La dernière étape est la reconnaissance . L'utilisateur doit choisir au préalable le dictionnaire de travail . Comme nous l'avons vu précédemment , une succession de boîtes de message apparaissent donnant le numéro du caractère suivi du temps de reconnaissance.

On donne ci-dessous le résultat des outils que nous avons proposé.

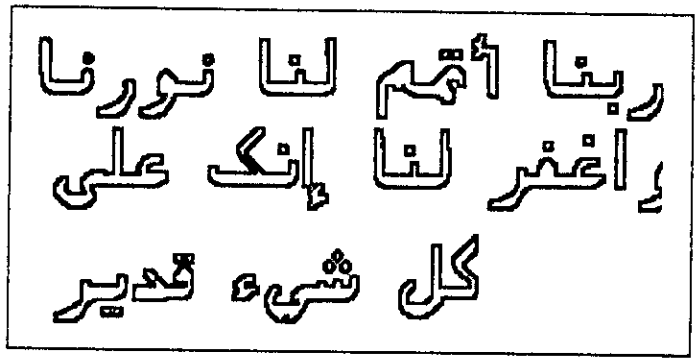


Figure 4.26. *Détection de contour en utilisant le masque de Sobel*

4.5 Résultats

Nous avons évalué les statistiques de notre système selon deux critéres à savoir les images et les textes :

• Pour les textes : Nous avons obtenu , sur deux textes , les résultats suivants:

- **82.76%** de reconnaissance pure.
- **05.45%** de rejection.
- **11.79%** de substitution.

• Pour les caractères : Nous avons obtenu , sur 151 caractères :

- **95.52%** de reconnaissance pure.
- **00.69%** de rejection.
- **03.79%** de substitution.

Nous estimons que ces résultats sont satisfaisants.

CONCLUSION GENERALE

Conclusion Générale

Nous avons tenté dans ce modeste travail de mettre au point un système OCR aussi complet que possible destiné en particulier aux caractères arabes .

Tous les travaux précèdent réaliser au sein du département d'électronique s'avèrent limité à l'identification d'un seul caractère à la fois sous le système MS-DOS.

En prenant ces différent travaux comme point de départ nous avons essayer d'élargir l'application à l'identification des paragraphe et de texte tout en profitant de l'avantage de l'environnement Windows .

Par ailleurs nous avons établi de nombreuses procédures qui permette une futur conception de système OCR grâce à d'autres méthodes.

En dépit du manque de la partie affichage du à l'indisponibilité de documentation relative à la connaissance de la structure des éditeurs , nous estimons que les résultats obtenu sont satisfaisante et à la hauteur de nos espérances.

Annexe

Comme premier exemple , on prendra une image à 256 niveaux de gris et cela pour mieux voir le résultat des prétraitements conçus spécialement pour ce type d'images , vu que le logiciel est basé sur la détection des pixels noirs et blancs.

Bien que l'image que nous allons proposer n'est pas un texte , mais peut donner une idée sur l'utilisation de ces commandes.

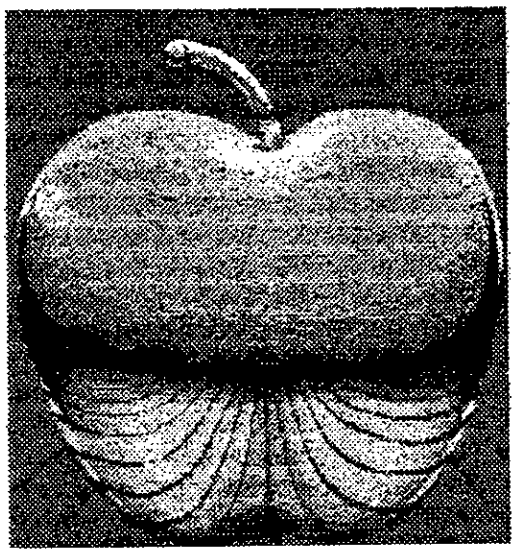


Figure 5.1. *Image originale*.

Pour diminuer le bruit présent dans cette image , on effectue un filtrage médian ce qui donne :

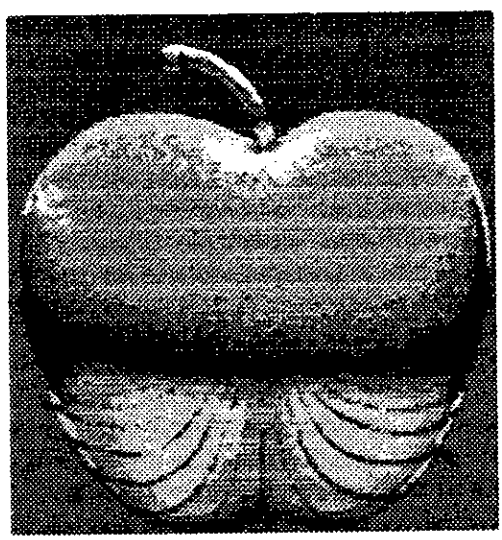


Figure 5.2. *Image après un Filtrage Médian.*

Si on veut binariser cette image , il faudra d'abord calculer son histogramme afin de bien choisir les seuils de binarisation .

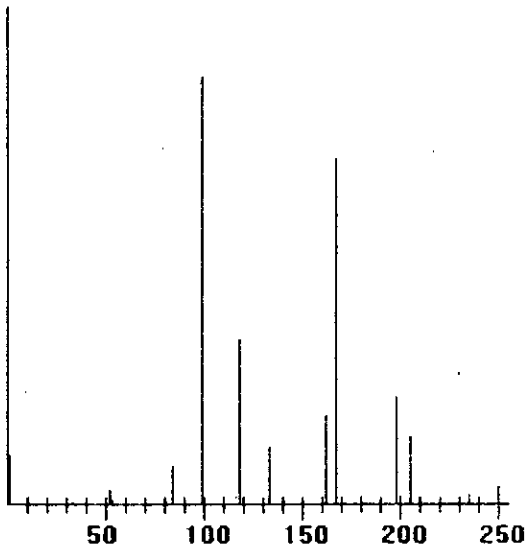


Figure 5.3. *Histogramme*

Histogramme de l'image : L'axe des abscisses représente les niveaux de gris et celui des ordonnées le nombre de points de l'image possédant ces niveaux de gris.

En se reportant à l'histogramme , on prend les seuils de binarisation S1 = 160 et S2 = 255. Tous les niveaux de gris entre ces deux seuils seront considérés comme faisant partie de l'image. Les autres niveaux seront attribués au fond de l'image (blanc).

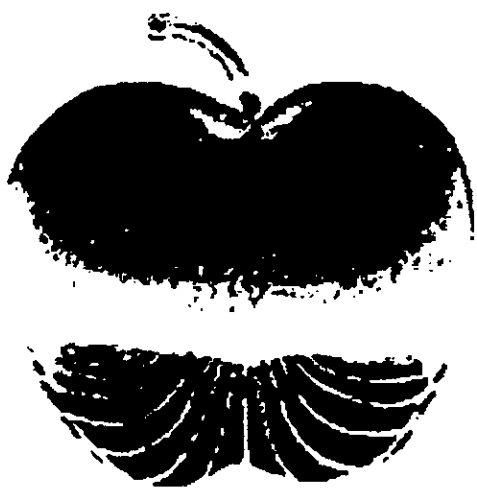


figure 5.4. *Image binarisée.*

En se qui concerne l'égalisation d'histogramme , on peut constater l'amélioration de contraste de l'image que réalise cette opération grâce à l'exemple suivant :

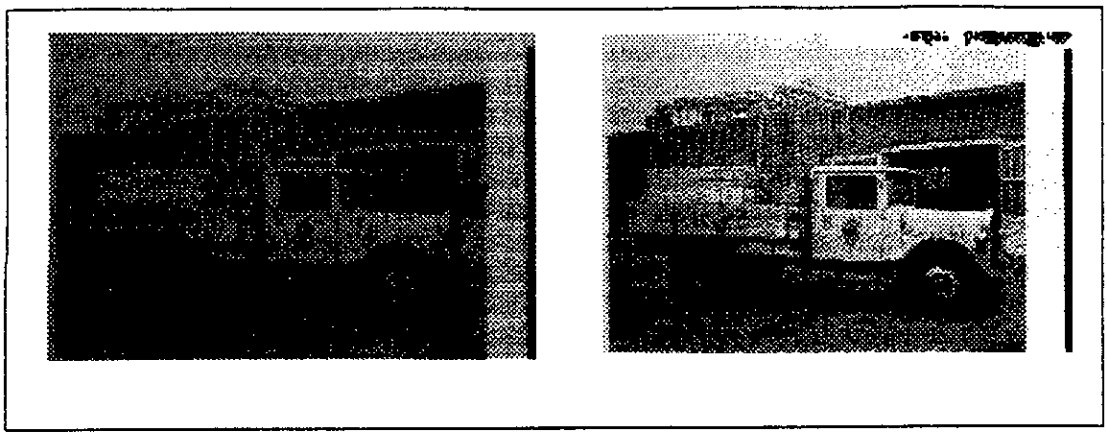


Figure 5.5. *Résultat de l'égalisation*

Les régions sombres de l'image possèdent maintenant une plus grande luminosité. On peut ainsi mieux percevoir les détails.

Comme deuxiéme exemple on prendra une image à 2 niveaux de gris (noir et blanc). Ce sera celle du caractère arabe " Chin" en position début (Figure suivante).

la détection des contours de ce caractère donne :

Opérateur de Sobel

Opérateur de Prewitt

On détermine le squelette de la figure 5.6 :

figure 5.6 *Algorithme de Zuang Suen*

BIBLIOGRAPHIE

Références bibliographiques

[Bel 92] A. BELAID et Y. BELAID, *"Reconnaissance des formes méthodes et applications"*, Inter édition 1992.

[Ben 94]: Amar. Benhouhou, " Contribution à la conception et la réalisation d'un système de reconnaissance de caractères arabes imprimés multipolices", Thèse de Magister, INI, octobre 1994.

[Ham 96] N. HAMIDI et K. BELGROUNE , *"Réalisation d'une phase de prétraitement pour la reconnaissance des caractères arabes"*, Thèse d'ingéniorat ,ENP 1996.

[Man 86] J. MANTAS , " An Overview of Character Recognition Methodologie " , Pattern Recognition,19(6):425-430 , 1972

[Poor 91] PC Magazine, *"Looking at the tiff specification from the inside"*.

[Sou 92]: C.Souchard, " Les formats des fichiers image sur PC ", PC expert, pp141 - 144, Juillet-Août 92.

[Ste 70] H.E.STEVENS ," *Introduction to the Special Issue on Optical Character Recognition(OCR)"* , Pattern Recognition , 2:147-150 ,1970 .

[Zek 95] B.ZEKRINI et A.ZEROUATI "*Etape de décision et Reconnaissance des caractères arabes multifonte , multitaille on utilisant une méthode structurelles"* , Thèse d'ingéniorat , ENP 1995.