

République Algérienne Démocratique et Populaire
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

Direction de l'Enseignement et de la Recherche

Génie ~~Electrique et Informatique~~

Option : Acquisition et Traitement de l'Information

المدرسة الوطنية المتعددة التخصصات
BIBLIOTHEQUE
Ecole Nationale Polytechnique

THESE DE MAGISTER

Présentée par
Melle AIT-DAOUD SIHAM
ingénieur d'état en Electronique (USTHB)

THEME

Reconnaissance optique et multiforme
de caractères arabes imprimés par une méthode
structurelle

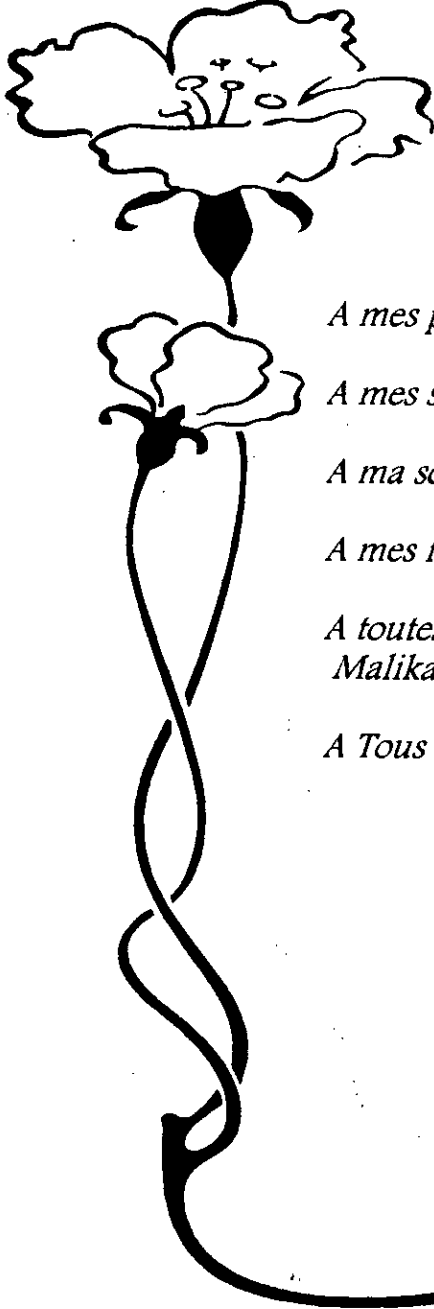
Soutenue le 30. 06.1997 devant le jury composé de:

Président: Mr D.BERKANIMaître de conférence (ENP)
Rapporteur: Mme L.HAMAMIChargée de cours (ENP)
Examineurs: Mr A.FARAHMaître de conférence (ENP)
Melle M.GUERTIMaître de conférence (ENP)
Mr L.SAADAOUChargé de cours (ENP)

Remerciements

- Je tiens à remercier M^{me} L.Hamami qui a bien voulu me faire confiance en me chargeant de cette étude et qui m'a constamment guidé et encouragé.
- Je voudrais aussi exprimer ma profonde gratitude à tous les membres du jury qui ont bien voulu me faire honneur en jugeant ce travail.
- Je désire également témoigner ma sincère reconnaissance à M^r A.Benhouhou enseignant à l'INI, M^r Sehad de CDTA et M^r S.Bouteflika maître de conférence à l'institut d'informatique (USTHB) pour m'avoir prêté leurs précieux documents qui ont servi à cette étude.
- Je suis très reconnaissante à M^{lle} Allet Nadia et M^{lle} Delloul Badiaâ de l'ENSI pour leur disponibilité, leurs conseils et leurs documentations.
- Mes remerciements vont à M^r M.Tellache chargé de cours à l'institut d'électronique de l'USTHB, ainsi qu'à M^r Ouahab Samir responsable du centre de calcul de l'IE (USTHB) pour avoir mis à ma disposition tous le matériel nécessaire pour mener à bien cette recherche.
- Que tous mes collègues et amies qui m'ont toujours aidé avec beaucoup de gentillesse trouve ici l'expression de ma sincère affection en particulier Douridj Saïda.

DEDICACES



A mes parents

A mes soeurs Selma, Widad et Fitna

A ma soeur Naoual et son époux Nabil

A mes frères Samir, Nabil et Smaïl

*A toutes mes amies Lynda, Saïda, Zineb, Assia,
Malika, Cherifa, Samia,...*

A Tous ceux qui me sont chers

Siham

| | |
|---|----|
| Introduction générale | 1 |
| Chapitre 1: Généralités sur la RF | |
| I. Définition de la RF..... | 3 |
| II. Méthodes utilisées pour la RF..... | 4 |
| III.1 Méthodes statistiques..... | 4 |
| III.2 Méthodes connexionistes..... | 6 |
| III.3 Méthodes structurelles..... | 8 |
| Conclusion..... | 11 |
| Chapitre 2: Reconnaissance de l'écriture. | |
| I. Historique..... | 13 |
| II L'OCR..... | 13 |
| II.1 Les problèmes de l'OCR..... | 14 |
| II.2. Quelques définitions utilisées en OCR..... | 15 |
| II.3. Les différents aspects de l'OCR..... | 17 |
| II.4. Les techniques de l'OCR..... | 17 |
| Chapitre 3: Acquisition et prétraitement de l'image. | |
| I. Acquisition des documents..... | 21 |
| II. Représentation d'une image..... | 23 |
| III. Les principaux formats de fichiers bitmap..... | 24 |
| IV. Prétraitement et amélioration de l'image..... | 30 |
| Chapitre 4: Analyse et extraction des caractéristiques | |
| I. Segmentation d'une image..... | 42 |
| II. Analyse et extraction des paramètres..... | 44 |
| III. Techniques d'extraction des paramètres..... | 45 |
| Chapitre 5: Réalisation du système de reconnaissance | |
| I. Présentation de l'écriture arabe..... | 55 |
| II. L'alphabet arabe..... | 57 |
| III. Reconnaissance des caractères arabes..... | 60 |
| III.1 Choix de la méthode..... | 60 |
| III.2 Acquisition des caractères..... | 60 |
| III.3 Filtrage..... | 60 |
| III.4 Extraction des primitives..... | 62 |
| III.5. Classification des caractères..... | 68 |
| IV. Caractéristiques secondaires..... | 70 |
| V. Dictionnaire..... | 73 |
| VI. Reconnaissance et identification..... | 74 |
| Conclusion générale | 77 |
| Bibliographie | |
| Annexe | |

Chapitre 1

Généralités sur la RF

Introduction générale

La reconnaissance optique des caractères ou OCR (optical character recognition) est l'un des travaux de recherche menés depuis quelques décennies dont le but est de faciliter la communication Homme-Machine. En particulier l'introduction de textes écrits sur un support papier dans la mémoire de l'ordinateur, sous forme d'une suite de nombres ASCII, peut grandement contribuer à cet objectif. Les principales applications de la reconnaissance de l'écriture sont :

- L'assistance et l'éducation pour la reconnaissance et la traduction en braille, et l'apprentissage de l'écriture et de la lecture pour les non voyants.
- Les banques et les assurances pour l'authentification de chèques pour les banques, (identification du signataire et sa signature) et la vérification de clauses des contracts pour les assurances.
- La poste pour le tri automatique du courrier.
- La police et la sécurité pour l'authentification des manuscrits et l'identification du scripteur.
- Bureautique pour l'archivage automatique de documents
- Système d'entrée de textes pour la traduction automatique en d'autres langues.

En général, un document est composé de blocs de différents types (texte, graphique, photos,...). Le bloc de texte peut être segmenté en paragraphes, en lignes, en mots et enfin en caractères.

Dans le cadre de la reconnaissance des caractères arabes imprimés ou dactylographiés, nous rencontrons de nombreux problèmes dus principalement à la cursivité de l'écriture, aux multiples formes d'un même caractère suivant sa position dans le mots (isolée, initiale, médiane et finale), à la variation de la fonte ou du style d'écriture et enfin aux distorsions et aux bruits qui apparaissent après l'acquisition des données.

Dans les deux premiers chapitres de ce mémoire, nous donnons un aperçu sur les méthodes utilisées en reconnaissance des formes en général et de l'écriture (OCR) en particulier. Le chapitre 3 quand à lui traite les modes d'acquisition d'un document et le stockage d'une image binaire ainsi que le prétraitement et le filtrage. L'analyse et l'extraction des caractéristiques sont expliquées au chapitre 4, et enfin nous terminons par le chapitre 5 qui décrit le système OCR réalisé. Celui ci est composé de quatre modules comme tous les logiciels de reconnaissance à savoir le prétraitement, l'extraction des primitives, la classification et enfin l'identification et la reconnaissance proprement dite.

Introduction

L'énorme quantité d'informations apportées par les systèmes informatiques de saisie et de stockage de mesures que l'homme est amené à traiter ces dernières années, a favorisé le développement de branches comme la reconnaissance des formes (RF) pour l'interprétation automatique des données.

I. Définition de la RF

La reconnaissance des formes est l'ensemble des méthodes et techniques de traitement de l'information qui permettent d'attribuer un objet au sens général du terme (Forme) à une famille (classe) prédéterminée avec laquelle cette forme possède des caractéristiques communes ou similaires [22].

Un système de reconnaissance des formes est décrit par les étapes suivantes:

a) Acquisition de la forme

Elle se fait grâce à un capteur tel qu'un microphone pour la parole ou la caméra et le scanner pour le traitement des images.

b)- Paramétrisation

C'est l'étape la plus importante et la plus délicate. Elle consiste à choisir l'espace de représentation de la forme .

3- Apprentissage

Cette phase sert à construire le dictionnaire contenant les caractéristiques d'un ensemble de prototypes afin de servir de référence pendant la reconnaissance .

4- Décision

C'est l'étape finale du système de reconnaissance qui consiste à identifier la forme inconnue en la comparant à des prototypes.

II. Méthodes utilisées pour la RF

II.1. Méthodes statistiques

Les méthodes statistiques se fondent sur une caractérisation statistique des paramètres des formes étudiées. Ces méthodes permettent de prendre une décision de classification d'une forme inconnue suivant un critère de : " Probabilité maximale d'appartenance à une classe ". Elles peuvent être résumées par l'arbre suivant :

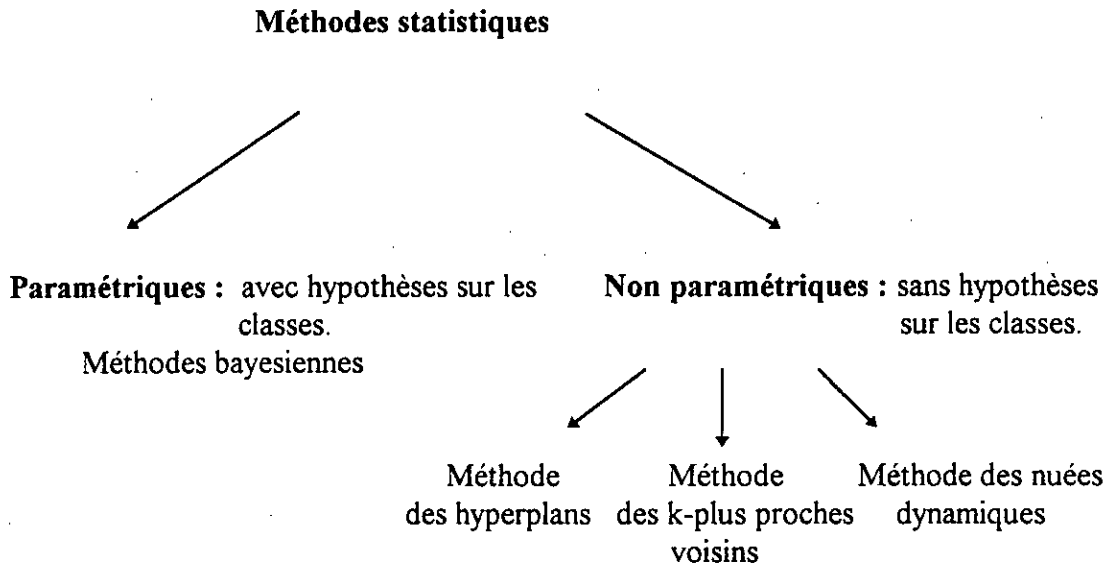


Fig. 1. Méthodes statistiques

II.1.1. Méthode de BAYES [18]

On suppose donc être dans un espace métrique Ω de dimension d . Soient $\omega_1, \omega_2, \dots, \omega_m$ les m classes possibles. Ces classes sont munies d'une probabilité à priori : P_i (calculée à partir des fréquences d'occurrence) avec $i=1 \dots m$.

Les formes à reconnaître sont des vecteurs aléatoires X , distribués selon les lois de probabilités $P(X/\omega_i)$: densité conditionnelle à l'appartenance à une classe donnée. On notera ω^\wedge une règle de décision, c'est à dire une fonction qui applique l'espace des représentations dans l'espace des classes. $\omega^\wedge(X) = \omega_i$ signifie que la classe d'indice i a été attribuée au vecteur représentant la forme X .

Soit $\lambda(\omega_i, \omega_j)$ une fonction qui nous indique le coût de la décision erronée suivante : " la forme X a été affectée à la classe ω_i alors qu'en réalité elle aurait du être affectée à la classe ω_j ."

Le rôle de théorie de la décision statistique est de fournir la fonction ω^\wedge qui minimise le coût moyen par décision précise . Pour cela , on choisit la règle de décision bayésienne $\omega^{\wedge*}$ qui est telle que :

$$\omega^{\wedge*}(X) = \omega_i \quad \text{si } l^i(X) \leq l^j(X) \quad \text{pour } j=1 \dots m$$

où $l^i(x) = \sum_{k=1}^m \lambda(\omega_j, \omega_k) * P(\omega_k/X)$ est le risque de prendre la décision $\omega^\wedge(x) = \omega_i$

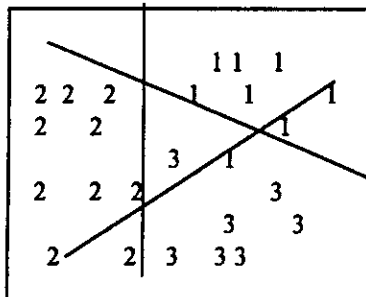
Pour chaque classe d'indice k, $P(\omega_k/X)$ et $P(X/\omega_k)$ sont reliés par la règle de BAYES qui s'écrit :

$$P(\omega_k/X) = \frac{P(X/\omega_k) * P_k}{P(X)}$$

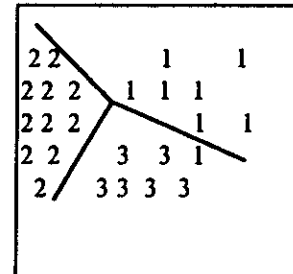
II.1.2. Méthode des hyperplans

Dans cette méthode, on cherche des surfaces discriminantes entre les classes. Le cas multiclasse peut se ramener au cas à deux classes en utilisant deux méthodes distinctes: soit en séparant chaque classe de toutes les autres par un hyperplan, soit en séparant les classes deux à deux. Le premier cas mène à de larges zones d'indécision, le second demande plus de calculs pour effectuer la décision de reconnaissance.

La méthode des hyperplans consiste alors à chercher une fonction permettant de séparer plusieurs classes différentes. Pour des raisons de simplicité, cette fonction est généralement représentée par une droite (hyperplan). Il arrive que les classes ne soient pas linéairement séparables mais qu'elles soient formées de sous-classes qui, elles, sont linéairement séparables. C'est le cas de la fig. 3 [6].



a) Séparation du 1^{er} type



b) Séparation du 2^{ème} type

Fig. 2. Séparation de 3 classes par la méthode des hyperplans

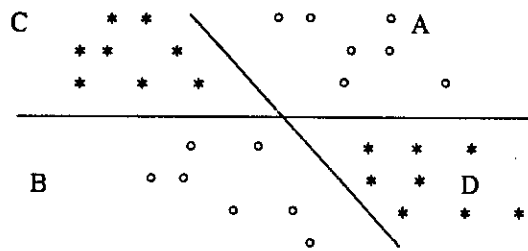


Fig. 3. Séparation des classes non linéairement séparables.

II.1.3. Les k-plus proches voisins

Cette méthode consiste à prendre la décision d'appartenance en fonction de l'ensemble d'apprentissage sans faire d'hypothèse a priori sur la forme des classes ou sur celles de leurs séparatrices. L'idée est extrêmement simple: la forme à reconnaître est un point dans l'espace R^d . On cherche autour de lui les k-plus proches voisins de l'ensemble d'apprentissage. Ils sont donc affectés d'un numéro de classe. La forme à reconnaître sera catégorisée dans la classe majoritaire parmi les k numéros ainsi déterminés. Cette technique est cohérente non paramétrique et possède un temps de calcul proportionnel au nombre de formes d'apprentissage [18].

II.2. Méthodes connexionnistes

Elles relèvent du champ général des « réseaux de neurones formels » qui intéresse à la fois les neurobiologistes, les physiciens et informaticiens. Certains y voient, en effet, un début de modélisation des processus de traitement de l'information par le cerveau.

En ce qui concerne la RF, nous allons présenter brièvement les réseaux multicouches de neurones formels qui sont pour le moment l'apport le plus significatif de ces études. L'élément de calcul unitaire d'un tel réseau est le neurone formel dont le comportement est décrit par la fig. 4. A l'instant t, des excitations a_i ($i=1...n$) sont appliquées aux n connexions d'entrées, chacune possédant un certain poids w_i . A l'instant t +1, la sortie S du neurone formel prend la valeur :

$$S = f(\sum w_i a_i)$$

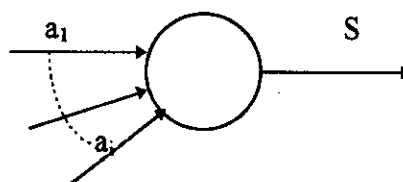


Fig. 4. Réseau de neurone formel.

où w_i est la pondération ou coefficient synaptique et f une fonction non linéaire, continue, dérivable, quasi-linéaire autour de zéro et visant à limiter la valeur absolue

de S. La fonction f est généralement représentée par la fonction de Heavyside ou de type sigmoïde [18] .

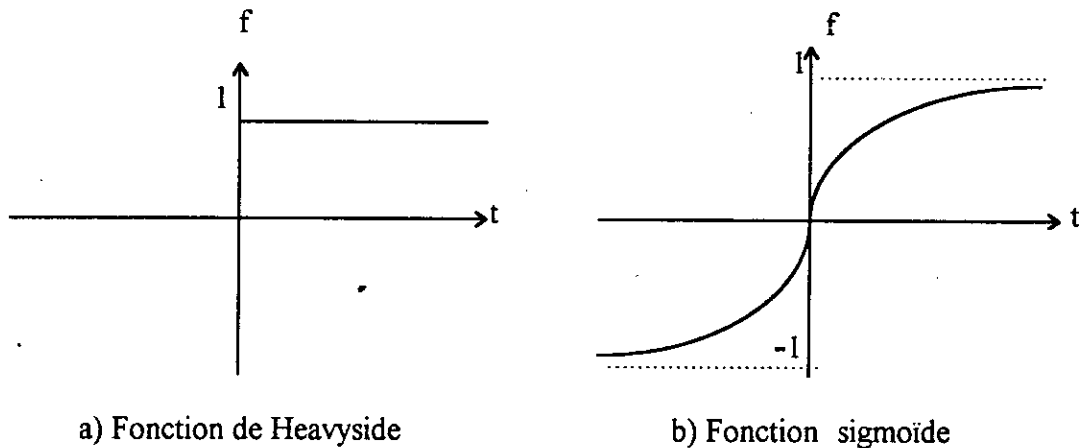


Fig. 5. Fonctions de seuillage

Dans la figure b, la fonction f appelée fonction sigmoïde est définie comme suit:

$$f(t) = \frac{\exp(\beta t) - 1}{\exp(\beta t) + 1}$$

f peut être aussi égale à $2H(t) - 1$ où $H(t)$ est la fonction de Heavyside [25] .

La structure d'un ensemble de neurones formels en réseau multicouches est réalisée selon le schéma de la fig. 6. Les sorties de certains neurones excitent certains autres éléments: une couche d'entrée reçoit les données, une couche de sortie code le résultat du calcul effectué sur elles. Entre ces deux couches, un nombre variable de couches cachées reliées l'une à l'autre dans le sens de la propagation du calcul. La fig. 6 représente un réseau à deux couches cachées, les données entrées à l'instant t produiront une sortie lisible à l'instant $t+3$.

L'utilisation d'un réseau multicouche en classificateur dépend du codage des sorties. Dans la méthode généralement employée, chaque unité de la couche de sortie représente une classe possible pour les données d'entrées. L'introduction sur la couche d'entrée d'une donnée inconnue induit après propagation dans les couches internes, des valeurs sur la couche de sortie. L'élément ayant la valeur la plus grande permet de choisir à quelle classe affecter les données d'entrées.

Une fois l'architecture du réseau (nombre de couches, nombre d'unités par couche, suppression éventuelle de connexions) a été décidée, la phase d'apprentissage doit permettre de choisir les poids w_i des connexions menant à chaque neurone

formel. Pour cela, on initialise les poids de connexion aléatoirement puis, on les fait évoluer vers un optimum en les obligeant à classer correctement des données d'apprentissage : le principe est de représenter une donnée dont on connaît la classe en entrée, et de comparer la sortie qu'elle produit avec celle qui code la classe idéale . L'erreur numérique permet de corriger les poids de façon à diminuer cette erreur. On note que dans le cas monocouche et f linéaire on retrouve la méthode de séparation par des hyperplans [18].

Conclusion sur les méthodes connexionistes

Les performances de cette méthode sont comparables à celles des k-plus proches voisins (kppv) mais si le temps d'apprentissage est assez long, celui de la décision est très rapide .

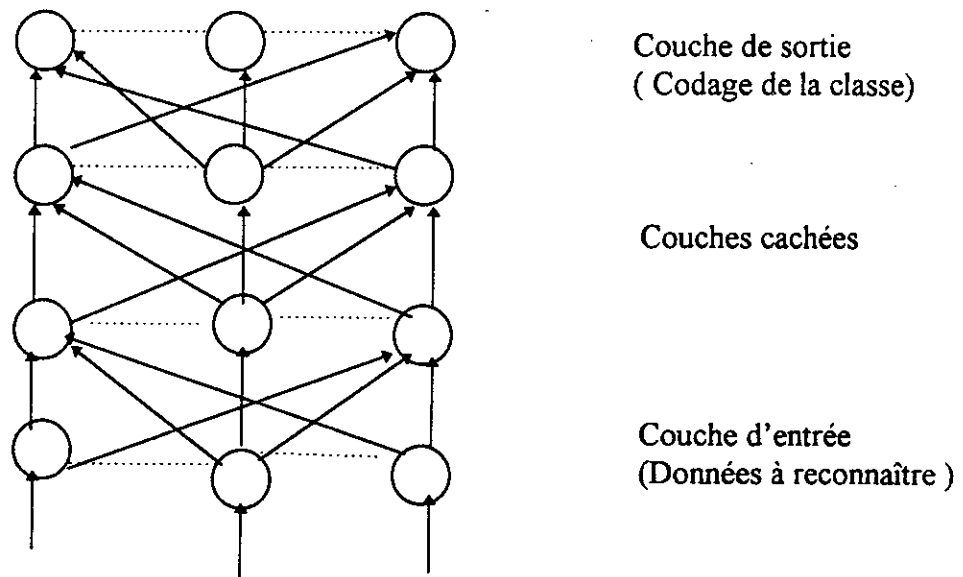


Fig. 6. Réseau à deux couches cachées.

II.3. Méthodes structurelles

Les approches vues précédemment ne permettent pas de prendre en compte l'information structurelle et contextuelle d'une information. Ces informations sont pourtant importantes dans de nombreux problèmes de reconnaissance des formes . les méthodes structurelles s'intéressent à la structure des formes et à leur description en terme d'assemblage de sous formes primitives. on distingue les méthodes descriptives et les méthodes syntaxiques [18].

II.3.1. Les méthodes syntaxiques

Dans ces méthodes, on cherche et on exploite les règles de construction des formes à partir de leurs composants d'où l'analogie que l'on peut faire avec l'analyse grammaticale ou syntaxique.

En effet, à partir d'un ensemble d'éléments primitifs de formes, et d'un ensemble de règles de combinaison de ces composants, on constitue d'autres composants intermédiaires qui seront combinés à leur tour avec d'autres éléments.

II.3.1.1 Structure de chaînes

Une description simple de la structure d'une forme peut être faite en la codant à un ensemble fini. En théorie des langages, un tel ensemble de symboles s'écrit :

$$X = \{ a, b, \dots, n \}$$

où X est appelé alphabet et a, b, \dots, n sont les formes élémentaires ou primitives. Une phrase sur X est définie comme une suite ordonnée d'éléments de X représentée par simple juxtaposition (concaténation) de ces éléments .

$$x = [a_1 a_2 \dots a_p] \quad \text{avec } a_i \in X.$$

Exemple

$X = \{ a, b, c \}$ est un alphabet.

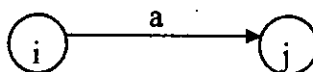
$x = bcaab$ est une phrase de X .

Dans le cas où une forme à étudier a fait l'objet d'une description en chaînes, il faut pour pouvoir la reconnaître être capable de la comparer à des prototypes décrits dans le même formalisme [22].

II.3.1.2 Automates et grammaires

Ce sont des outils mathématiques dont le rôle est d'engendrer des ensembles de phrases munis d'une structure commune. Le principe de leur application en RF consiste à modéliser une classe de formes par une grammaire. Une forme inconnue représentée par une phrase est testée pour déterminer si la grammaire peut l'engendrer.

Un automate (automate fini) est représenté par un graphe dont appelle les nœuds des états et dont les flèches entre les nœuds portent des symboles de l'alphabet. l'élément de base d'un automate est donné par la figure suivante:



II.3.2 Méthodes descriptives

Dans ces méthodes, on retrouve celles dites structures des graphes. Un graphe est défini par un couple $G = (V,E)$ où V est un ensemble fini de nœuds ou de sommets, et E un ensemble de couples ordonnés de sommets appelés arcs.

Exemple

$$V = \{ 1,2,3 \}$$

$$E = \{ (1,2), (2,1), (1,3), (2,3), (3,4), (4,2) \}$$

Dans cette approche (celle des graphes), on retrouve des graphes de décomposition hiérarchique comme ceux de (Et /Ou) où la racine correspond à la forme elle même et les feuilles aux primitives (Fig 7). Il existe aussi des graphes relationnels attribués où les nœuds représentent les sous formes et les arcs les relations entre ces formes.

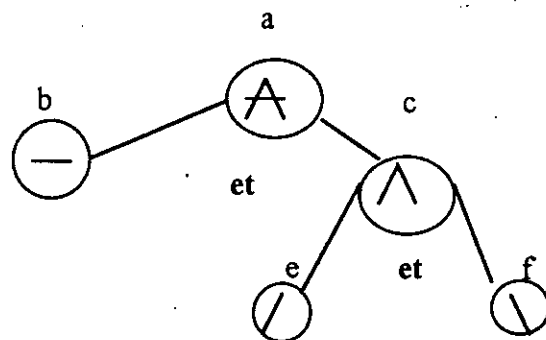


Fig. 7. Graphe Et/Ou

a est reconnu si b et c le sont .

c est reconnu si e et f le sont.

Les chiffres indiquent les points d'attache des composants du caractère .

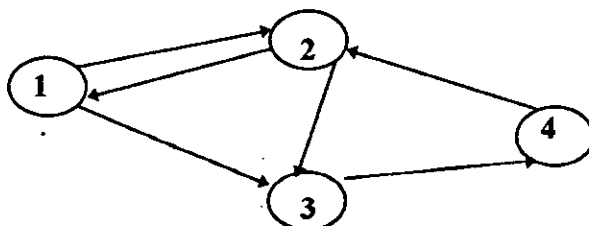


Fig. 8. Exemple de graphe orienté

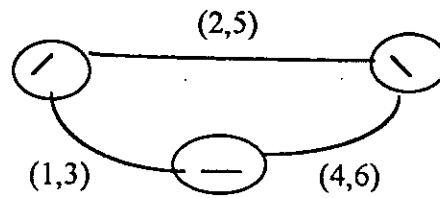


Fig. 9. Graphe relationnel attribué

Conclusion sur les méthodes citées

On peut conclure sur les méthodes vues précédemment (méthodes structurelles et statistiques) qu'aucune de ces deux approches n'est meilleure que l'autre. Certains problèmes se résolvent naturellement mieux à partir d'une approche statistique (ceux où le nombre de mesures est important). Il est donc très avantageux pour un structuraliste d'introduire un peu de décision statistique à un certain niveau de son problème, de même pour un statisticien de tenir compte des propriétés structurelles pour éviter des calculs exhaustifs [22].

Chapitre 2

Reconnaissance de l'écriture

Introduction

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

I. Historique :

C'est en 1940 qu'apparaissent les premières applications informatiques de l'OCR (optical character recognition). Ainsi, en 1975, les japonais utilisaient couramment des lecteurs qui déchiffraient le code postal inscrit à la main ou tapé à la machine. Dans la même période, le français **CONTER** construisait un système de lecture optique automatique de textes imprimés destinés aux non voyants. Par la suite, la compagnie américaine **KURZWEIL** a réussi à améliorer le système précédent en proposant des machines à lire pour aveugles formulant le texte à haute voix par synthèse vocale [21].

Pour le monde arabe, les travaux dans ce domaine sont relativement restreints. La plupart des chercheurs ont utilisé la tablette graphique comme moyen d'acquisition car elle présente l'avantage de restituer fidèlement le sens du tracé. Parmi ces travaux, on note le projet **IRAC** (interactive recognition arabic character) réalisé à l'université de Nancy 1 par **Adnan AMIN**. Ce projet a été le premier à avoir abordé sérieusement le problème de la reconnaissance de l'écriture arabe (imprimée et manuscrite). D'autres travaux aussi intéressants ont suivi celui d'**AMIN** notamment en Egypte, en Tunisie, en Syrie où **Mohamed Bassam EL KURDY** a conçu en 1991, un système OCR pouvant reconnaître un texte imprimé en utilisant la segmentation en caractères avec des méthodes très rapides et enfin en Arabie Saoudite où les recherches de **I. ABUHAIBA** ont donné des résultats encourageants pour le manuscrit [2][3].

En Algérie, les travaux les plus importants ont été ceux réalisés par Mme **Farida MATOUGUI** (CDTA) en 1990 avec son système appelé **SIRCAM** (système interactif de reconnaissance de caractères arabes isolés et manuscrits) et Mr **Ahmed BENHOUBOU** (INI) en 1994. Les principales caractéristiques du premier travail sont l'utilisation d'une caméra ccd pour l'acquisition des différents caractères et la squelettisation pour extraire les primitives (nombre de boucles, nombre de tracés, position de points isolés et leur nombre) [21]. Le second travail est capable de segmenter un texte imprimé en caractères en se basant essentiellement sur des méthodes structurales (extraction des traits caractéristiques: pavés et précurseurs, nombre de carrefours ou intersection des traits et le nombre de boucles) [8].

II. 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 se résume à numériser la page entière et la deuxième phase consiste à partir d'un fichier graphique (BMP, PCX...) à distinguer des formes que l'on identifie comme des lettres de l'alphabet.

II.1. 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 soit 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.

Un système OCR est décrit par le schéma suivant :

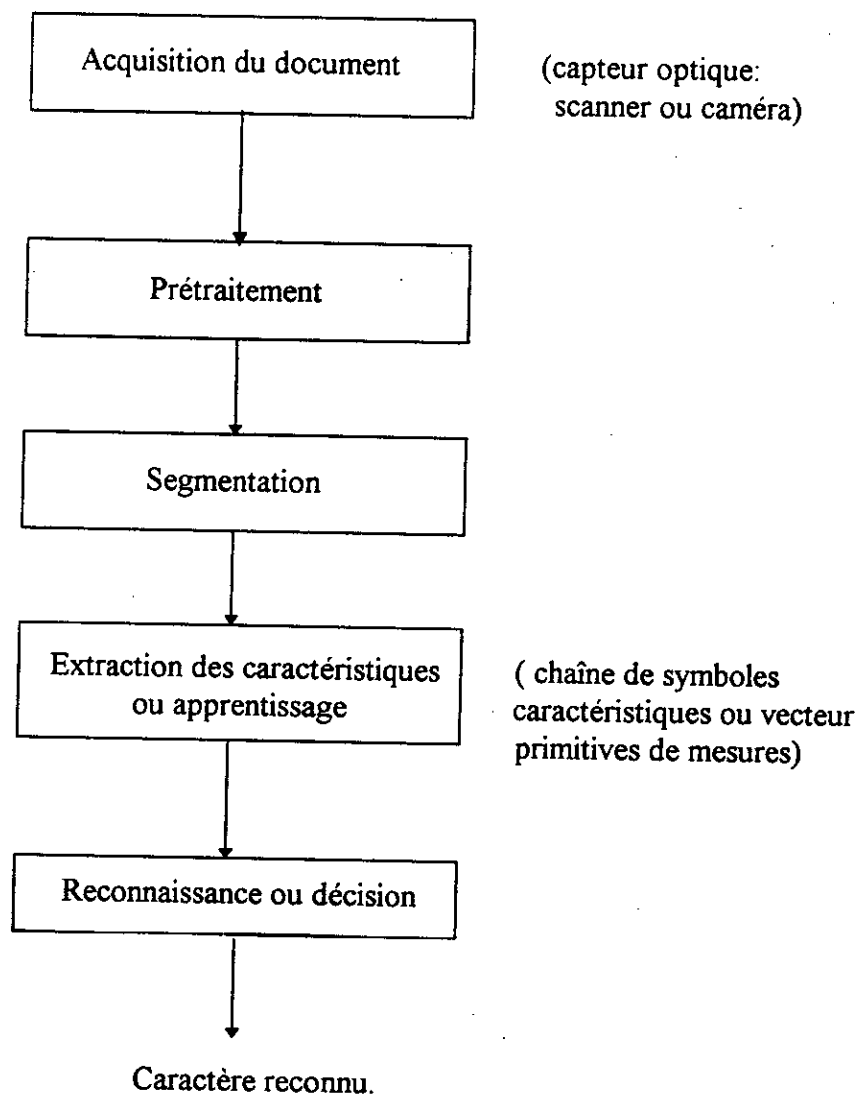


Fig. 10 . Diagramme fonctionnel d'un système OCR

II.2. Quelques définitions utilisées en OCR

II.2.1. Primitive

On appelle primitive une forme élémentaire constituée par une agglomération de pixels pertinents à laquelle il est possible de donner une interprétation [19].

II.2.2. Propriétés topologiques et géométriques

a) *Propriétés topologiques*

Elles correspondent à la définition des parties de la forme comme les cavités, les boucles et le nombre de parties composant la forme.

b) *Propriétés géométriques*

Elles correspondent aux distances entre parties, aux convexités, concavités, allongement, rapport de taille ...

II.2.3. Représentation

C'est la chaîne d'éléments d'un alphabet fini permettant de représenter (de coder) une information.

II.2.4. Identification

Acte de donner un nom à une représentation également le résultat de cette action.

II.2.5. Apprentissage

L'apprentissage se charge d'acquérir la reconnaissance et de l'organiser en classes ou modèles de références. Un bon apprentissage est réalisé en fournissant un bon choix de formes de référence.

a) *Apprentissage supervisé*

Il est dit supervisé si les différentes familles de formes sont connues a priori et si la tâche d'apprentissage est guidée par un superviseur ou professeur.

b) *Apprentissage non supervisé ou sans professeur*

Ce mode d'apprentissage nécessite un nombre élevé d'échantillons et des règles de construction précises et non contradictoires pour bien assurer la formation des classes [6].

II.2.6. Définitions typologiques

a) *Fonte*

Une fonte est un ensemble de caractères d'une police particulière à une taille particulière.

b) *Police*

Une police indique le style ou le graphisme des caractères et leurs attributs (italique, gras, souligné, ombré...).

c) *Mot*

Un mot est une juxtaposition de lettres de type bas de casse ou haut de casse.

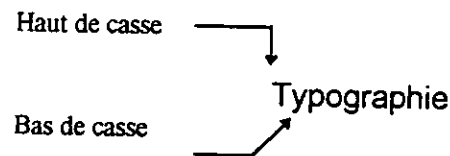


Fig. 11. Composition d'un mot.

d) *Alignement*

Tous les caractères d'une même fonte s'alignent en pied des lettres courtes: a, e, ... Cette ligne est appelée ligne de base et sert de référence pour des calculs d'interligne et d'inclinaison. La ligne médiane est la ligne passant par le haut des lettres courtes et constitue avec la ligne de base une zone de forte densité de points appelée bande centrale [6].



Fig. 12. Alignement des caractères d'un mot

II.3. Les différents aspects de l'OCR

II.3.1. 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 est 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 [6].

II.3.2. Reconnaissance On-line (en ligne ou en temps réel)

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

II.3.3. Reconnaissance Off-line (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.

II.4. Les techniques de l'OCR

II.4.1. 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 ces pixels sont présents. Cependant à cause de la rigidité de ces méthodes (conçus à partir d'emplacement 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 qui n'est plus employée aujourd'hui que pour des produits spécifiques (banques, poste) et présente l'avantage d'être très rapide, relativement fiable et surtout très facile à mettre en œuvre. Mais elles correspondent à une série de caractères bien particuliers dans une police prédéfinie avec une taille fixe.

II.4.2. Méthodes structurales

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

a) *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.

b) *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.

c) *Méthode syntaxique*

Chaque caractère est représenté par une phrase 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 phrase 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 phrase pour qu'elle soit acceptée par la grammaire.

II.4.3. Méthode des experts

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 [8].

Dans ce cas chaque système expert est conçu pour reconnaître un seul caractère. L'avantage de cette technique est la rapidité, dès qu'un caractère est identifié, on passe à l'autre. Le seul problème de cette méthode est la complexité de programmation d'un expert [12].

II.4.4. Méthode heuristique

Les méthodes 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 au problèmes à résoudre et ne peuvent donc pas être généralisées à d'autres.

Chapitre 3

*Acquisition et prétraitement
de l'image*

I. Acquisition des documents

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. Pour cela, il existe deux modes d'acquisition: le mode dynamique et le mode statique.

I.1. Mode dynamique

Dans les systèmes de reconnaissance en temps réel, les capteurs doivent être capables de transmettre les données en temps réel à l'ordinateur. Les capteurs les plus utilisés sont les tablettes graphiques sensibles, sur lesquelles on écrit à l'aide d'un stylo spécial. Les mouvements du stylo sont enregistrés magnétiquement ou électrostatiquement et transmis à l'ordinateur sous forme d'une succession de points. Ce moyen offre l'avantage de restituer fidèlement le tracé dans l'ordre où il a été exécuté, ce qui constitue déjà une étape avancée de la reconnaissance.

I.2. Mode statique

I.2.1. 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. Le scanner balaye le texte ligne par ligne et digitalise chacune en une série plus ou moins importante de points. 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 ppp. Une plus grande résolution n'apporte rien à la précision, au contraire, elle augmente le nombre de points à traiter et génère ainsi du bruit appelé grain du papier [6].

a) *Scanner à main*

C'est un scanner qu'il faut déplacer à la main sur le document à numériser.

b) *Scanner de table*

b.1. Scanner à défilement

Le document est entraîné à travers une fente pour passer devant la barrette CCD (charge coupled device). Ce type de scanner n'accepte que les feuilles volantes, 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.

I.2.2. Caméra

a) Les caméras à tube

Ces caméras sont composées d'une cible photoconductrice qu'explore un faisceau électronique. L'exploration se fait ligne par ligne. A chaque ligne correspond en sortie un signal analogique proportionnel à l'intensité.

b) Les caméras CCD

Elles sont constituées par un assemblage de photodiodes. Chacune d'entre elles délivre une intensité proportionnelle à un point de l'image appelé pixel. Contrairement aux caméras à tubes qui fournissent un signal continu qu'il faut échantillonner, pour isoler chaque pixel, les caméras CCD donne directement l'intensité pour chaque point de l'image puisqu'à chaque point correspond une photodiode. Il existe deux types de caméras CCD: les caméras matricielles et les caméras unilignes.

c) Les caméras matricielles

Les photodiodes sont assemblées en une matrice n lignes et p colonnes. Cette matrice peut être carrée, les résolutions courantes sont alors de 100×100 ou 256×256 ou bien rectangulaire (300×512).

d) Les caméras unilignes

Les photodiodes sont agencés suivant une seule ligne. Elles ont l'avantage de posséder une résolution supérieure à 2048 points. L'obtention d'une image nécessite cependant le déplacement de la caméra afin d'acquérir plusieurs lignes pour obtenir une image bidimensionnelle [29].

II. Représentation d'une image.

II.1. Définition d'un pixel (picture element)

Un pixel est une zone de morphologie simple et unique (généralement rectangulaire) pouvant se confondre à un point fixe sur l'image ayant comme seule caractéristique visuelle, une valeur de niveau de gris ou de couleur affectée ou déterminée dans le même emplacement de l'image réelle ou d'un objet graphique quelconque. Dans ce cas, la description de tout objet graphique à l'aide de pixels se fait par juxtaposition (ensemble de pixels ordonnés) [24].

II.2. L'image numérique

L'image numérique peut revêtir différentes formes. Tout d'abord, comme information numérisée, elle peut être le contenu d'une mémoire, d'un disque, ou d'une bande magnétique, c'est à dire un certain code dont la matérialisation sous forme d'image ne peut résulter que du décodage inverse et de sa lecture par un dispositif d'affichage. D'autre part, comme représentation d'un processus de génération, elle peut être un programme qui ne produira le codage effectif qu'une fois qu'il aura été exécuté dans la mémoire centrale. Ainsi, l'image numérique en tant qu'objet de perception n'existe pas et seule sa représentation est pertinente [28]. Elle peut être stockée sous une forme BITMAP ou une forme vectorielle, nous allons définir dans ce qui suit la représentation BITMAP.

II.2.1. Représentation BITMAP

Une image bitmap peut être considérée comme une fonction $I(x,y)$ où x et y sont les coordonnées d'un point de l'image, et I une fonction d'intensité lumineuse et de couleur. L'origine physique de la couleur est le spectre lumineux qui peut être décrit en termes d'énergie comme une fonction continue de la longueur d'onde. Comme les spectres continus sont impossibles à réaliser, nous sommes donc obligés de les représenter par un échantillonnage. Le nombre d'échantillons doit être fixé au départ: un seul échantillon pour un espace monochrome, et trois pour un espace de couleur (RVB).

Numériser une image, c'est échantillonner le signal $I(x,y)$ par un nombre fini de points de côté (x_{max}, y_{max}) et de pas (dx, dy) en réalisant une mesure moyenne de l'intensité lumineuse qui sera stockée dans un mot de la mémoire écran. Le produit $x_{max} \cdot y_{max}$ est ce qu'on appelle la résolution de l'image. Le décodage de l'image se fera ensuite simplement en affichant au point de coordonnées x et y de l'écran, un pixel de dimension dx, dy et d'intensité de la valeur mémorisée.

II.2.2. Codage de Freeman

Il est principalement utilisé pour les images consistant en des lignes tracées sur un fond uniforme c'est à dire les images où seule une description de ces lignes est nécessaire [29].

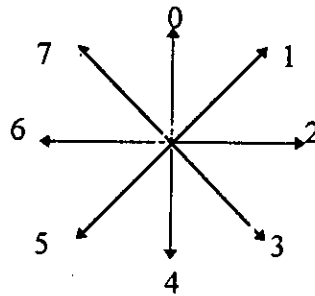


Fig. 13. Codage de freeman

III. Les principaux formats de fichier bitmap

III.1. Le fichier 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 Ms-Dos.

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 un multiple de 4 et que le format BMP a été créé pour PC. En effet, selon que l'on travaille avec processeur Intel ou Motorola, le 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 PC (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) [28].

III.1.1. 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 le fichier par les deux lettres ASCII «BM». Les 4 suivants, indiquent la taille totale du fichier, viennent ensuite 4 octets réservés à zéro puis les 4 derniers contiennent la valeur de l'offset de la partie bitmap.

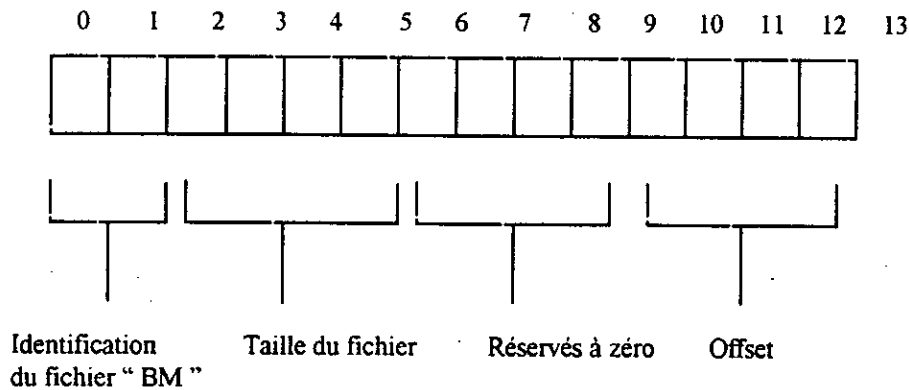


Fig. 14. Entête du fichier BMP

b) Informations sur l'image: BITMAP INFO 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. |
| 4 | Hauteur de l'image en couleur. |
| 2 | Nombre de surface de l'image. |
| 2 | Nombre de bits par pixel (1, 4, 8 ou 24). |
| 4 | Méthode de compression (0: non compression). |
| 4 | Taille de l'image en octets. |
| 4 | Résolution verticale de l'image en pixel par mètre. |
| 4 | Résolution horizontale de l'image en pixel par mètre. |
| 4 | Nombre de couleurs utilisées (0: toutes les couleur). |
| 4 | Nombre de couleurs importantes (0: toutes). |

III.2. Le fichiers TIFF

Le fichier TIFF conçu en 1986 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 une 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 des entités 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 ou 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 près 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[26].

III.2.1. Structure du fichier

Un fichier TIFF commence toujours par un header ou entête contenu sur huit octets et dont le rôle de chaque byte est donné ci dessous:

Byte 0-1: ils indiquent le type de processeur utilisé: II pour Intel et MM pour Motorola. Dans le premier cas, l'octet le moins significatif est listé le premier, par contre dans le second processeur c'est l'octet le plus significatif qui est le premier lu.

Byte 2-3: Ils donnent la version utilisée. Ces bytes contiennent toujours la valeur $2A_{hex}$ (42 décimale).

Byte 4-7: Donnent l'adresse du premier IFD calculée à partir du début du fichier.

Nous avons vu précédemment qu'un IFD est un tableau. Il contient au maximum 45 tags ou étiquettes. Chacune d'elles stocke une information codée sur 12 octets. Les deux premiers correspondent au type d'information représentée(largeur d'image, nombre de bits par pixel, type de compression, échelle de gris...), les deux octets qui suivent indiquent le type (au sens informatique) des données selon les valeurs contenues suivis de quatre octets dont le rôle est d'indiquer le nombre de données. Enfin, les quatre derniers contiennent soit la donnée elle même, soit un pointeur sur la section supplémentaire où l'information manquante sera trouvée.

Remarque

Si le fichier contient une seule image, la fin du premier IFD donne un zéro sinon elle indiquera l'adresse du prochain IFD.

| Valeur lue | Type |
|------------|--|
| 1: | Byte (entier non signé sur 8 bits). |
| 2: | ASCII (Sur 8 bytes: 7 octets en code ASCII et le 8 ^{ème} octet est mis à zéro). |
| 3: | Short (entier non signé sur 16 bits). |
| 4: | Long (entier non signé sur 4 octets). |
| 5: | Rationnel (long integer sur 8 octets: 4 pour le numérateur et 4 pour le dénominateur). |

III.2.2. Rôle de quelques étiquettes (Tags)

| N° de l'étiquette | Rôle |
|-------------------|--|
| 273: | Adresse du premier pixel de l'image. |
| 256: | Largeur de l'image. |
| 257: | Longueur de l'image. |
| 277: | Nombre d'échantillons par pixel. Valeur = 1: image monochrome. Valeur = 3: image RGB(rouge, vert et bleu). |
| 258: | Nombre de bits par pixel (image binaire: 1bit, 16 niveaux gris ou de couleurs : 4 bits, 256 niveaux de gris ou de couleurs:8 bits et enfin une image à plus de 16 millions de couleurs : 24 ou 32 bits). |
| 259: | Type de compression: 1: Absence de compression. 2: CCITT (RLE: Run Length Encoding). 5: LZW. |
| 256: | 0: indique que la valeur 0 des données bitmap représente le blanc et 1 le noir pour une image monochrome 1: la valeur 0 des données bitmap représente la couleur noire et 1 la couleur blanche. 2: codage RGB. Les 8 premiers bytes codent le rouge, les seconds le vert et les derniers codent les pixels de couleur bleue. |

Il existe d'autres fichiers graphiques tels que les fichiers GIF avec une compression LZW (Lempel-Ziv-Welch), les fichiers PCX de Paintbrush avec une compression RLC (Run Length Codage), les TGA,...

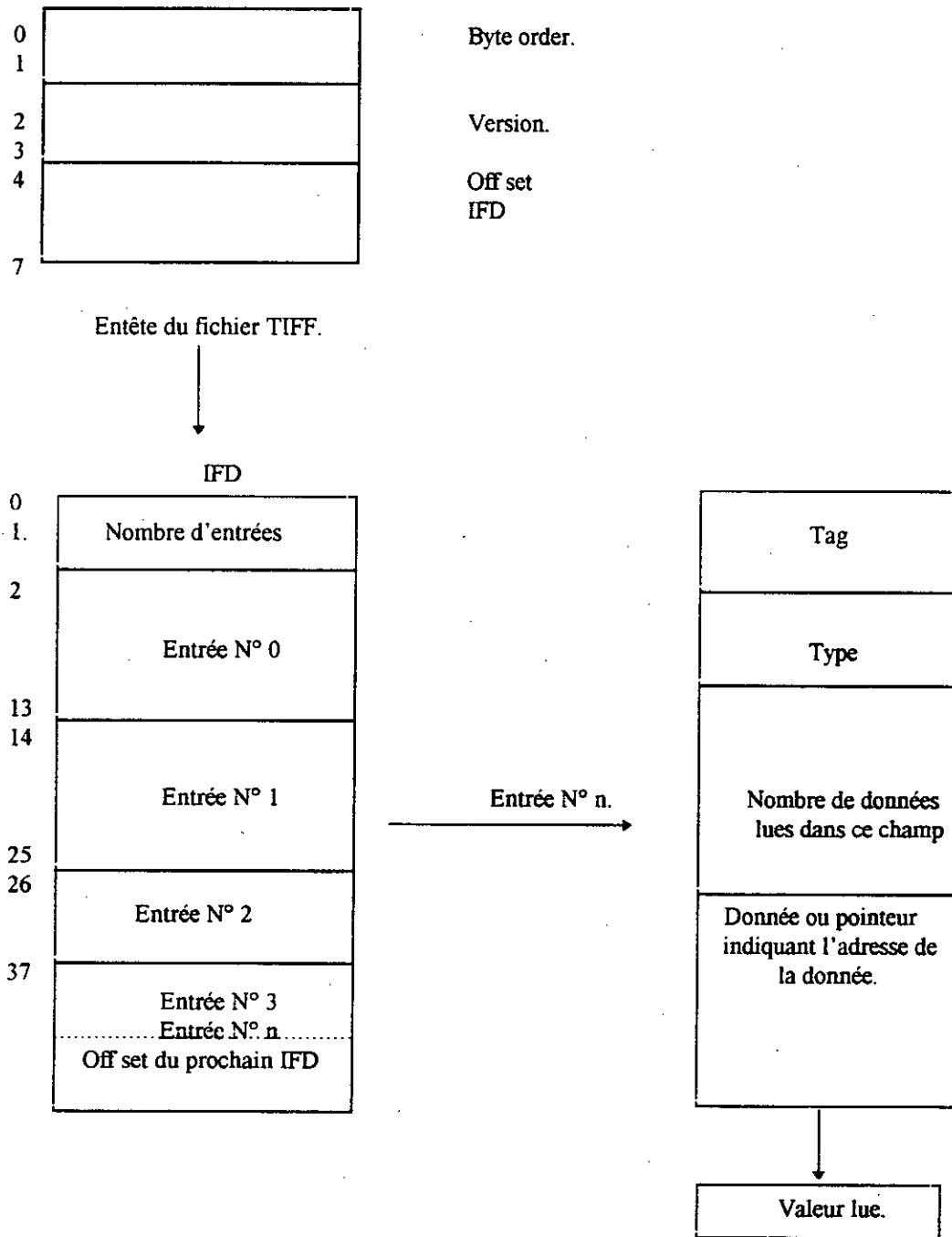


Fig. 15. Structure d'un fichier TIFF

IV. Prétraitement et Amélioration de l'image

L'image brute est une structure complexe dans son contenu. Ceci est dû à plusieurs facteurs, d'une part la grande quantité d'informations qui y réside, d'autre part le nombre important de processus indépendants de l'utilisateur qui tendent à y introduire des distorsions indésirables. On peut citer les principales perturbations:

- Le bruit électronique généré par le capteur.
- Les défauts du système d'échantillonnage et de quantification.
- L'éclairage qui peut engendrer des ombres, des reflets etc.

L'amélioration des images consiste en un ensemble de méthodes destinées à améliorer l'aspect visuel d'une image. L'ensemble des techniques ayant cette fonction fait partie de ce qu'on appelle les prétraitements.

IV.1. Définition d'un bruit

Le bruit est à l'origine d'un grand nombre de difficultés lorsqu'on désire analyser un signal quelconque. En traitement d'image, le bruit présent dans l'image numérisée est le résultat du bruit électronique du capteur et de la qualité du digitaliseur. Comment ce bruit se matérialise-t-il ? Supposons que nous avons une surface rigoureusement homogène. L'image numérique devra être une matrice dont tous les éléments ont la même valeur. Or, on remarque qu'autour d'une valeur moyenne proche du réel, certains pixels prennent des valeurs différentes plus ou moins éloignées. Ceci est le résultat de la présence d'un bruit.

Par approximation, on peut caractériser le bruit par un bruit impulsionnel c'est à dire un bruit affectant de brusques variations à des pixels isolés. Pour lutter contre les effets du bruit, il est nécessaire d'opérer des transformations qui pour chaque pixel tiennent compte de son voisinage. Ces techniques sont le filtrage, le lissage, la squeletisation, la normalisation et la vectorisation [6].

IV.2. Techniques de lissage et filtrage

IV.2.1. Techniques de lissage

L'image de caractères peut être entachée de bruit dû à l'acquisition et souvent à la qualité du document conduisant soit à des absences de points ou à une surcharge de points. Les techniques de lissage permettent de résoudre ces problèmes par des opérations locales appelées nettoyage et bouchage.

Pour l'opération de nettoyage, on utilise les masques suivants:

| | | |
|---|---|---|
| 0 | 0 | 0 |
| x | 1 | 0 |
| 0 | 0 | 0 |

Masque (a)

| | | |
|---|---|---|
| x | 0 | 0 |
| 1 | 1 | 0 |
| x | 0 | 0 |

Masque (b)

x: Désigne un point quelconque (1 ou 0).

Le masque (a) permet l'élimination des points isolés ou situés à l'extrémité des contours en l'appliquant dans les huit directions qui sont: nord, nord-est, nord-ouest, est, ouest, sud, sud-est, sud-ouest.

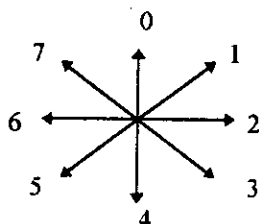


Fig. 16. les huit directions

La figure (b) représente un masque qui élimine les points formant des angles droits ou des excroissances de contour appliquée dans les huit directions citées plus haut.

L'opération de bouchage consiste à égaliser les contours et à boucher les trous internes à la forme du caractère. Le bouchage des trous isolés est utilisé quand le voisinage d'un point de fond correspond au masque suivant:

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Masque c)

| | | |
|---|---|---|
| x | 1 | 1 |
| 0 | 0 | 1 |
| x | 1 | 1 |

Masque d)

La correction des irrégularités des contours dues à un effritement de la forme se fait par le masque (d) en l'appliquant dans les huit directions.

IV.2.2. Filtrage

a) *Filtres linéaires*

L'emploi de ces filtres est limité car le bruit est rarement de nature linéaire [19]. Parmi eux, on cite celui de Gauss.

$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-x^2 - y^2}{2\sigma^2}\right)$$

σ : Ecart-type.

$$I_F = I * G.$$

Avec:

* : Produit de convolution.

G: courbe gaussienne dont l'équation est:

$$G(x, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

I: Image à filtrer.

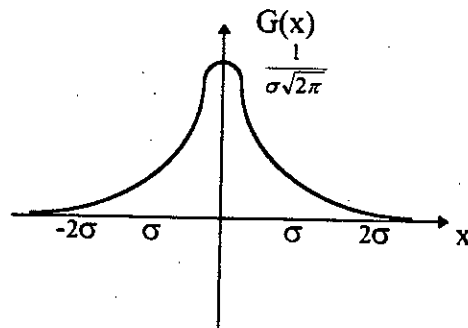


Fig. 17. Courbe gaussienne.

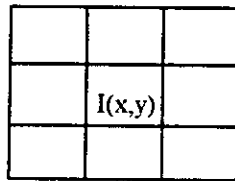
b) *Filtres non linéaires*

Les plus répandus sont les filtres morphologiques et médian. Ces filtres ont l'avantage de supprimer les bruits de haute fréquence, de lisser les bords, de nettoyer les détails et de boucher les petits trous. Le plus célèbre est le filtre médian de Tuckey.

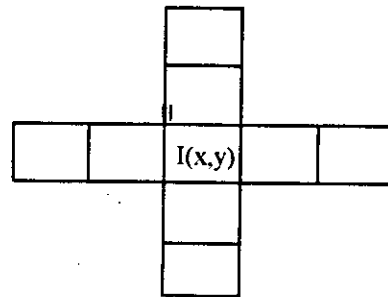
b.1 Principe du filtre de Tuckey

Pour chaque pixel de l'image, on procède comme suit:

- On classe les pixels voisins par valeurs croissantes.
- On prend la valeur médiane des pixels classés et on l'affecte au pixel courant [29].



a) Fenêtre 3x3



b) Masque en croix.

Fig. 18. Filtre médian.

b.2. Les filtres morphologiques [29]

Chaque image binaire peut être codée comme un élément logique donc 0 ou 1.

b.2.1. Dilatation

Elle permet d'éliminer les pixels noirs isolés au milieu des parties blanches de l'image. Le processus est réalisé en promenant sur l'image une fenêtre carrée de taille 3x3 pixels et en effectuant pour chaque pixel de l'image le OU logique de ses huit voisins. Le pixel est mis à 1 si le résultat vaut 1, sinon il reste inchangé.



Image originale



Image dilatée

Fig. 19. a) Dilatation

b.2.2.Erosion

C'est le procédé dual de la dilatation. L'érosion permet d'éliminer les pixels blancs isolés. On effectue le ET logique des huit pixels voisins.

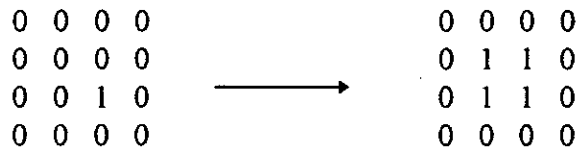


Image originale

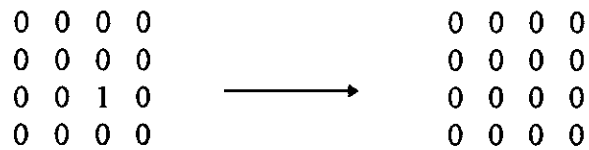


Image obtenue après érosion

Fig. 20. Erosion.



a) Dilatation



b) Erosion.

Remarque:

Le pixel 1 représente le blanc et 0 le noir.

b.2.3.Ouverture

C'est une opération d'érosion suivie d'une dilatation.

b.2.4.Fermeture

C'est une dilatation suivie d'une érosion.



Fig. 21. Ouverture et fermeture d'une image.

c) *Lissage d'Amin [5]*

Soit X un pixel d'une matrice 3x3.

$$\begin{array}{ccc} A & B & C \\ D & X & E \\ F & G & H \end{array}$$

On applique successivement les deux formules suivantes.

$$X = X + B \times G \times (D + E) + D \times E \times (B + G).$$

$$X = X \times (G + D \times B + B \times E + A \times H + C \times F)$$

\times : ET logique.

$+$: OU logique.

Ce lissage permet de réduire les irrégularités sur le contour du tracé du texte comme les points anguleux et les points non isolés au milieu de parties blanches.

IV.3. Techniques de normalisation

La taille des caractères peut varier d'une fonte à une autre et même au sein d'une même fonte après agrandissement ou réduction ceci peut causer une instabilité des paramètres. Une technique de prétraitement consiste à ramener les caractères à la même taille.

IV.3.1. Algorithme de SRIHARI

Il s'agit de transformer l'image du caractère de dimension h_i et l_i en une image h_n et l_n . Une étape intermédiaire consiste à produire une image de dimension h_n et l_n telle que:

$$p = \frac{h}{h_i} \quad \text{et} \quad l_n = p * l_i$$

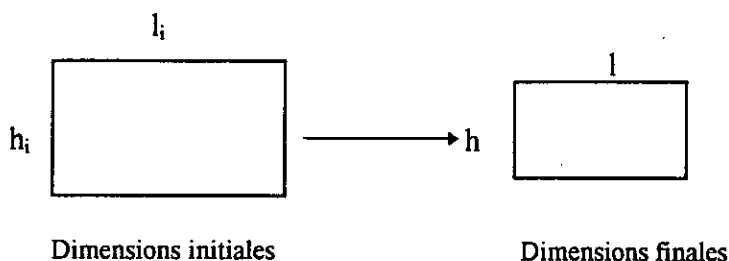


Fig. 22. Normalisation d'une taille

La normalisation en hauteur est obtenue en transformant chaque pixel noir $I(x,y)$ de l'image du caractère en $I(px,py)$. la normalisation en largeur de l'image ainsi obtenue se fait par l'examen de deux cas. Si $l_n < l$, alors l'image normalisée en hauteur est centrée dans une surface hl . Si par contre $l_n > l$, alors l'algorithme balaye à noir tout pixel de coordonnées $((x/l_n)*l,y)$ si le pixel (x,y) l'est aussi. Un autre algorithme intéressant appelé algorithme de Hussain est détaillé dans l'ouvrage de A.Belaïd [6].

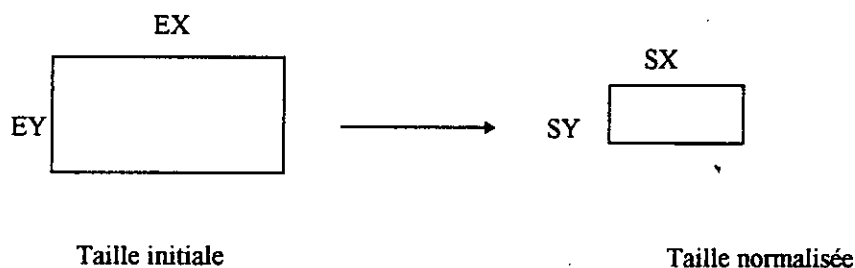
Le dernier algorithme que nous allons citer dans ce paragraphe est celui utilisé par Mme Guamaz [15].

$$X_{new} = X_{old} * \left(\frac{S_x}{E_x} \right)$$

$$Y_{new} = Y_{old} * \left(\frac{S_y}{E_y} \right)$$

X_{old}, Y_{old} : coordonnées d'un élément de la matrice du caractère d'entrée.

X_{new}, Y_{new} : coordonnées de l'élément correspondant dans la matrice normalisée.



Ces méthodes peuvent engendrer des déformations surtout lorsque le caractère a une taille inférieure à celle normalisée. Pour résoudre ces problèmes Mme Guamaz présente dans sa thèse des solutions pour le remplissage des trous obtenus après normalisation.

IV.4. Techniques de squelettisation

Le but de la squelettisation est de simplifier l'image du caractère en une image plus facile à traiter. Seulement le résultat n'est pas toujours fiable car il s'accompagne souvent d'une perte irréversible d'informations ainsi que de barbelures.

Les critères retenus pour les méthodes de squelettisation sont les suivants:

- L'épaisseur du squelette doit être de un.
- Le squelette doit conserver les propriétés topologiques de la forme comme le nombre de parties connexes, le nombre de trous...
- Le squelette doit respecter les propriétés métriques de la forme comme la longueur totale et la distance entre les différentes parties de la forme.

Parmi les algorithmes les plus célèbres, on trouve ceux de THOME(78), de ZHANG et SUEN, MARTON...

IV.4.1. Algorithme de THOME

Cet algorithme est caractérisé par sa rapidité et sa simplicité. Il utilise seize masques appliqués parallèlement à l'image [21].

| | | | |
|-------|-------|-------|-------|
| x 1 x | x 0 x | x 1 x | x 1 x |
| 1 p 0 | 1 p 1 | 0 p 1 | 1 p 1 |
| x 1 x | x 1 x | x 1 x | x 0 x |
| 0 0 0 | 0 0 0 | 1 1 x | x 1 1 |
| 0 p 0 | 0 p 0 | 0 p 0 | 0 p 0 |
| x 1 1 | 1 1 x | 0 0 0 | 0 0 0 |
| 0 0 x | 0 0 1 | x 0 0 | 1 0 0 |
| 0 p 1 | 0 p 1 | 1 p 0 | 1 p 0 |
| 0 0 1 | 0 0 x | 1 0 0 | x 0 0 |
| x 0 0 | 0 0 x | x 1 x | x 1 x |
| 1 p 0 | 0 p 1 | 0 p 1 | 1 p 0 |
| x 1 0 | x 1 x | 0 0 x | x 0 0 |

x : Désigne un pixel quelconque c'est à dire 1 ou 0.

p : Est un point non essentiel.



Fig. 23. Algorithme de THOME

IV.5. Méthode séquentielle

Dans cette méthode, on applique séquentiellement les huit masques donnés ci-dessous pour l'amincissement de l'image. La squelettisation s'effectue par une succession d'opérations d'amincissement jusqu'à l'obtention d'une structure stable ne pouvant plus être amincie [29].

| | | | |
|---------------|--------------------|--------------|-------------------|
| 5 x 2 | 6 5 x | 7 6 5 | x 7 6 |
| 6 1 3 | 7 1 2 | x 1 x | 4 1 5 |
| 7 x 4 | x 4 3 | 4 3 2 | 3 2 x |
| Est. | Sud-est. | Sud. | Sud-ouest. |
| 4 x 1 | 3 4 x | 2 3 4 | x 2 3 |
| 3 1 6 | 2 1 7 | x 1 x | 5 1 4 |
| 2 x 5 | x 5 6 | 5 6 7 | 6 7 x |
| Ouest. | Nord-ouest. | Nord. | Nord-est. |

$$F_s = (1*2*3*4) * (\overline{5+6+7})$$

*: ET logique.

x: point quelconque(1 ou 0).

Algorithme d'amincissement

Début:

Pour chacun des huit masques faire
 Pour chaque point de l'image Im
 Faire

```

calculer Fs.
Si Fs alors
    Im(x,y)=0.
Sinon
    Im1(x,y)=Im(x,y).
Fsi.
Fp.
Fp.
Fin.

```

Algorithme de squelettisation

Début:

```

Tant que image non stable
    Faire
        Amincissement.
Ftq.

```

Fin.

La stabilité est détectée par l'absence de changement entre l'image amincie et l'image originale.

Algorithme de détection de la stabilité

Début:

Stable= vrai.

```

Pour chacun des huit masques Faire
    Pour chaque point de l'image Faire
        Si Fs alors
            Im1(x,y)=0.
            Stable= faux.
        Sinon
            Im1(x,y)=Im(x,y).
        Fsi.
    Fp.

```

Fp.

Fin.



Fig.24. Algorithme séquentiel.

Conclusion sur ces deux méthodes

D'après les résultats obtenus, on remarque que la méthode de Thomé est plus rapide que la méthode séquentielle seulement elle présente plus de déformations dans certains caractères.

IV.5.1. Squelettisation par vectorisation

Cette méthode est issue des travaux de vectorisation de l'image par codage de segments (Run Length Encoding : RLE). L'image est balayée ligne par ligne, les segments noirs et blancs de chaque ligne sont extraits puis codés par des paramètres relatifs à leur position et longueur. L'analyse de ces paramètres permet ensuite d'extraire les propriétés géométriques et topologiques. Parmi les algorithmes qui sont fondés sur ce principe, on cite celui de Pavlidis(86) appelé LAG (Line Adjacency Graph) [6].

Conclusion

Les techniques de squelettisation ont la réputation d'être lourdes à mettre en œuvre, coûteuses en temps de calcul et donnant lieu à de nombreuses erreurs.

Chapitre 4

Analyse et extraction des paramètres d'une image

I. Segmentation d'une image

Lors de l'analyse d'une image, il est nécessaire d'avoir une bonne segmentation du support afin de ne pas fausser le calcul des paramètres, qui, parfois dépendent beaucoup de l'état de l'image segmentée.

La segmentation peut être vue sous différents aspects (statistiques ou déterministes) mais elle demeure cependant basée sur deux notions essentielles.

- Définition des classes (critères d'uniformité de niveau de gris, critère de détection de frontières...).
- Regroupement des points d'une même classe (notion de région, notion de contour).

Plusieurs méthodes ont été définies au cours des recherches en analyse d'images et il est certain qu'il existe un lien étroit entre le problème abordé et la méthode employée. Ce lien peut être décrit par la nécessité d'utiliser des critères de couleur, de formes ou de textures. Ce choix est lié au contexte présent dans l'image.

I.1. Définition

Etant donné une image discrète de support X composé de points de coordonnées (i,j) , le processus de segmentation peut être défini comme la recherche d'une partition de X en sous ensembles x_i ($i = 1 \dots M$) tel que si P représente un prédicat d'uniformité sur un ensemble de points consécutifs, on ait :

- $X = \bigcup_{i=1}^M x_i$
- x_i : ensemble connexe.
- $P(x_i)$ est vrai.
- $P(x_i \cup x_j)$ est faux pour i différent de j [11].

I.2. Présentation des méthodes de segmentation

On peut distinguer deux façons d'aborder la segmentation.

- Les méthodes associées aux régions qui permettent de partitionner l'image.
- Les méthodes associées aux frontières qui permettent de découper l'image.

Pour décomposer un texte en caractères, on peut distinguer les deux types de contact qui sont : le contact par accollement où les dessins des caractères sont effectivement joints et le contact par chevauchement où les caractères ne se touchent pas mais sont inséparables verticalement à cause du rétrécissement de l'espace.



a) Contact par accolement



b) Contact par chevauchement

Fig. 25. Type de contact.

I.2.1. Méthode de projection

On projette verticalement les pixels noirs sur l'axe des abscisses. La zone de la segmentation correspond alors à un creux sur l'histogramme représenté par une faible valeur de la projection cumulée.

I.2.2. Découpage par marquage des régions

Dans le cas d'un chevauchement, on découpe la région litigieuse en plusieurs zones horizontales de huit pixels de haut. Dans chaque zone, on marque les colonnes de pixels blancs les plus à gauche et les plus à droite par G et D. Le G le plus à droite et le D le plus à gauche sont respectivement considérés comme les points de segmentation du caractère de gauche et du caractère de droite. si cette méthode échoue, on segmente à la frontière de la grille [6].

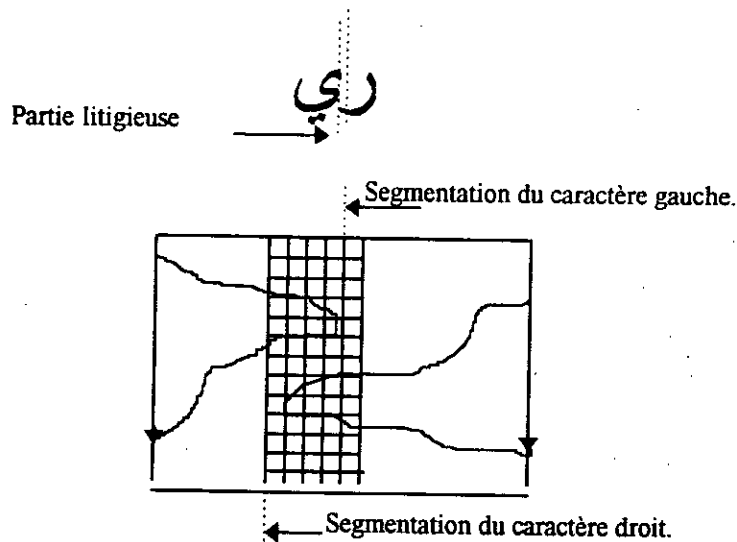


Fig. 26. Séparation de deux caractères liés par chevauchement.

II. Analyse et extraction des paramètres

L'analyse du caractère passe par l'analyse de sa forme et l'extraction de ses traits caractéristiques. Le problème qui se pose est le choix de ces traits et la distinction des différentes formes. On cherche souvent des formes comme des allongements, des courbures, des concavités, des croisements, des extrémités...

Pour trouver ces caractéristiques, il est nécessaire de donner quelques définitions importantes qui seront utilisées ultérieurement.

II.1. Voisinage, Repère

L'image est considérée comme un ensemble fini d'un espace discret. Dans cet espace, on introduit une topologie permettant de décrire et d'analyser l'image de manière quantitative: repère, voisinage...

II.1.1. Repère

Dans un espace discret, une image est souvent liée à un repère de travail dans lequel on précise les coordonnées de l'origine de l'image. C'est ce qu'on appelle hauteur et largeur de l'image.

II.1.2. Voisinage

Avant de définir le voisinage, nous allons définir d'abord une distance.

a) Distance

Soient p_1 et p_2 deux pixels de l'image tels que leurs coordonnées sont respectivement (x_1, y_1) et (x_2, y_2) . On définit les deux distances suivantes:

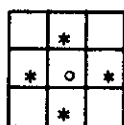
$$d_1(p_1, p_2) = \text{Sup}(|x_1 - x_2|, |y_1 - y_2|).$$

$$d_2(p_1, p_2) = |x_1 - x_2| + |y_1 - y_2|.$$

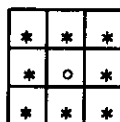
Ces distances permettent de définir la notion des points voisins. On dit que x est voisin de y pour la distance d_0 si et seulement si :

$$d(x, y) \leq d_0 \quad \text{avec } d_0 \text{ un seuil de distance (en pixels).}$$

Pour $d_0 = 1$, les voisins au sens de d_1 sont les huit points les plus proches de p , tandis que l'ensemble des voisins de p au sens de d_2 est limité par les quatre points les plus proches.



Les 4-voisins



Les 8-voisins

Fig. 27. Voisins d'un pixel.

o: le pixel considéré.

*: Voisin du pixel

b.1 Distance euclidienne entre deux pixels

$$d[(x_1, y_1), (x_2, y_2)] = [(x_1 - x_2)^2 + (y_1 - y_2)^2]^{1/2}$$

On remarque qu'en utilisant cette technique, les huit pixels n'ont pas la même distance:

Exemple:

$$d[(x, y), (x + 1, y)] = 1.$$

$$d[(x, y), (x + 1, y + 1)] = \sqrt{2}.$$

III. Techniques d'extraction des paramètres

III.1. Analyse du contour

III.1.1. Définition d'un contour d'une image

On définit un point du contour comme un point de l'image appartenant à la frontière entre deux régions voisines. D'une manière générale, ce point est le lieu où on observe une brusque variation de la fonction de niveau de gris.

Le processus d'extraction de contour peut être décomposé en plusieurs étapes distinctes qui sont:

- **Mise en évidence du contour**

Elle s'obtient par une différentiation de l'image. Elle peut s'appliquer à des images binaires ou non.

- **Réduction des contours**

Dans cette étape, on réduit l'épaisseur du contour à un seul pixel.

- **Binarisation des contours**

Cette étape définit quels sont les contours que l'on considère comme importants et ceux que l'on rejette.

- **Description des contours**

C'est la dernière opération sur le contour. Elle permet d'organiser les points du contour en structures simples telles que segments de droites, arcs de cercle, etc.

III.1.2. Opérateur de dérivation

a) *Opérateur de Roberts ou du gradient [29]*

C'est l'application directe de la formule d'une dérivée. Pour chaque pixel $I(x,y)$ d'une image I , les dérivées en x et y sont:

$$A_x = I(x+1, y) - I(x, y).$$

$$A_y = I(x, y+1) - I(x, y).$$

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

$$H_x = \begin{pmatrix} -1 & +1 \end{pmatrix} \quad H_y = \begin{pmatrix} -1 \\ +1 \end{pmatrix}$$

L'amplitude est alors :

$$A = \sqrt{A_x^2 + A_y^2}.$$

Et la direction du gradient est:

$$D(x,y) = \arctg \left(\frac{A_y}{A_x} \right)$$

Pour des raisons de simplicité et de rapidité de calcul, on préfère utiliser les deux approximations suivantes:

$$A = \text{Max}(|A_x|, |A_y|)$$

ou

$$A = |A_x| + |A_y|$$



a) Image Originale



b) Contour de l'image

Fig. 28. Extraction du contour par l'opérateur de Roberts

b) *Masque de Sobel*

$$H_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

$$H_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

c) *Masque de Prewitt*

$$H_x = \begin{pmatrix} -1 & 0 & -1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

$$H_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

d) *Gradient amélioré [27]*

Le contour de l'image est obtenu en calculant quatre gradients pour chaque fenêtre 4x4 de l'image. La valeur du gradient amélioré est donnée par l'expression ci-dessus représentant la moyenne géométrique des éléments A, B, C, et D.

$$\text{Grad}(a) = (A * B * C * D)^{1/4}$$

*: produit simple.

$$\text{Avec la fenêtre } 4 \times 4 = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix}$$

$$A = |f - k| + |j - g|.$$

et

$$B = |a - p| + |m - d|.$$

$$C = |b - o| + |i - h|.$$

$$D = |c - n| + |e - l|.$$

L'intérêt de cette méthode est de réduire les perturbations dues à la dérivation.

e) Opérateur de Kirsh [6]

On approche la dérivée dans les huit directions et on conserve celle qui a le module le plus élevé. Pour i allant de 0 à 7, on calcule $H_i(x,y)$ de la manière suivante:

$$H_i(x, y) = \left| 5 * (a_i + a_{i+1} + a_{i+2}) - 3 * (a_{i+3} + a_{i+4} + a_{i+5} + a_{i+6} + a_{i+7}) \right|.$$

a_i sont les voisins du point (x,y) .

$$\begin{pmatrix} a_0 & a_1 & a_2 \\ a_7 & (x,y) & a_3 \\ a_6 & a_5 & a_4 \end{pmatrix}$$

III.1.3. Opérateur Laplacien

Comme son nom l'indique, la détection du contour s'obtient en dérivant deux fois la fonction $I(x,y)$.

$$\Psi^2 = \frac{\delta^2 I}{\delta x^2} + \frac{\delta^2 I}{\delta y^2}.$$

Dans ce cas, on cherche les passages par zéro du laplacien qui correspondent aux maximums de la dérivée première. Une approximation discrète de cet opérateur est donnée par l'expression suivante:

$$L(x, y) = I(x, y) - \frac{1}{4} * (I(x, y + 1) + I(x, y - 1) + I(x - 1, y) + I(x + 1, y))$$

et dont le masque est :

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

On notera que le laplacien fournit directement des contours d'un pixel épaisseur. Pour diminuer le bruit, JF Canny a développé un algorithme qui dérive la convolution d'une image avec une gaussienne.

Soit $G(x, y) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right)$ une gaussienne.

La dérivée de l'image I filtrée est:

$$\psi_f = \psi(G * I) = f_x + f_y$$

où f_x et f_y représentent respectivement les dérivées suivant x et y. D'où :

$$\psi_f = (G * I)_x + (G * I)_y$$

$$\psi_f = (G_x * I) + (G_y * I)$$

Les dérivées partielles de la courbe de Gauss sont données par les deux expressions suivantes:

$$G_x(x, y) = -\frac{x}{\sigma^2 \sqrt{2\pi}} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right)$$

$$G_y(x, y) = -\frac{y}{\sigma^2 \sqrt{2\pi}} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right)$$

On peut donc écrire :

$$G_x(x, y) = G_x(x) * G(y)$$

$$G_y(x, y) = G_y(y) * G(x)$$

$$\psi_f = G_x(x) * G(y) * I + G_y(y) * G(x) * I$$

La direction et l'amplitude sont alors données par les relations suivantes:

$$A = (f_x + f_y)^{1/2}$$

$$\text{Dir} = \text{Arctg} \left(\frac{f_y}{f_x} \right)$$

On peut conclure de ces relations que plus σ augmente plus les contours perdent en finesse dans les détails.

III.1.4. Amincissement des lignes de contour

Soit $I(x,y)$ l'image considérée, $A(x,y)$ l'amplitude des gradients et $\text{Dir}(x,y)$ leur direction.

Algorithme

Pour chaque point $A(x,y)$ du contour, on détermine les points adjacents qui se trouvent dans la même direction du gradient. Soit $A(x_1,y_1)$ et $A(x_2,y_2)$ ces points. L'algorithme consiste à comparer l'amplitude de $A(x,y)$ à celle de ses points adjacents, si celle-ci est supérieure alors $A(x,y)$ est conservé sinon, il est mis à zéro.

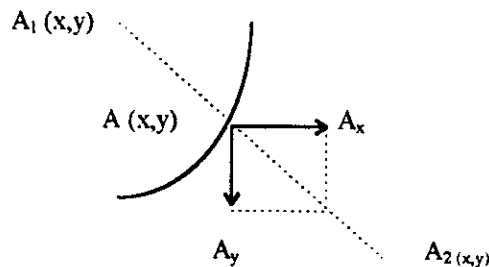


Fig. 29. Binarisation du contour.

III.1.5. Description des contours

Une fois le contour obtenu, une autre phase est nécessaire pour représenter chaque objet sous une forme réduite. Le principe est d'isoler chaque contour et de décrire chacun d'eux suivant un code.

III.1.6. Codage des contours

a) *Suivi du contour*

Algorithme de Duda et Hart

- * Balayer l'image jusqu'à la rencontre d'un point de l'objet.
- * Tourner à gauche.
- * Tester le point suivant.
 - Si c'est un point d'objet tourner à gauche.
 - Si c'est un point fond tourner à droite.
- * Fin du suivi du contour en revenant au point de départ.

Cet algorithme permet de déterminer un seul contour dans une image et de plus les résultats ne sont pas satisfaisants [27].

Algorithme de Chottera et Shridar

L'algorithme de Chottera et Shridar est capable de détecter les contours de tous les éléments de l'image, seulement, il s'applique uniquement sur des images binaires. Les détails de cet algorithme sont donnés dans la thèse de Saâdaoui [27].

b) Codage de Freeman

Ce codage est une représentation exacte de chaque pixel du contour et utilise la position relative d'un pixel du contour par rapport au précédent. On définit un masque qui détermine le code pour chacune des huit directions possibles.

Exemple:

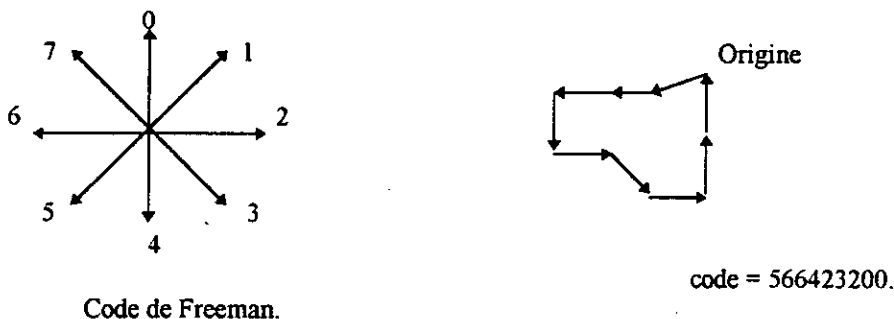


Fig. 30. Codage de Freeman.

III.2. Approximation par segments de droite

Cette approche ne restitue pas toute l'information contenue dans les contours mais en réalise une approximation dont la précision est fixée par l'utilisateur. Plus cette précision est faible plus l'information est restreinte. Cette méthode consiste à sous échantillonner le contour par un pas de taille P en effectuant un suivi de celui ci. on relie le point de départ R et le point courant P_c par un segment de droite.

On calcule alors les coordonnées de P_i (P_i un point du contour) tel que la distance de P_i au segment soit maximale. Soit d_i cette distance, si d_i est supérieure à d_0 (distance seuil ou précision de l'approximation), on découpe le segment en deux $[R P_i]$ et $[P_i P_c]$ et on recommence l'opération pour chaque segment jusqu'à ce que les segments trouvés soient suffisamment proche du contour.

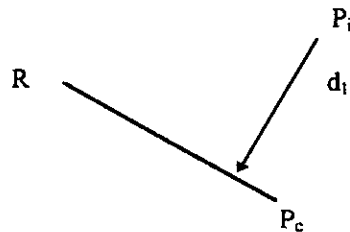


Fig. 31. Exemple d'approximation par segments de droites.

III.3. Transformée de Hough

Une droite peut être représentée dans un plan par sa distance r par rapport à l'origine et un angle θ . Elle est donc représentée par un point dans le plan de coordonnées (θ, r) . Pour chaque point du contour, on calcule l'équation de la droite qui passe par ce point et dont la direction est orthogonale à celle du gradient.

Remarque

Pour des points de contour voisins et de directions du gradient voisines, on obtient sur le plan (θ, r) une accumulation de points dans une même région (Fig.32). Dans ce cas, on effectue généralement la moyenne de ces points dans le plan (θ, r) et on essaie de construire la droite par la transformée inverse et de la borner dans la région considérée.

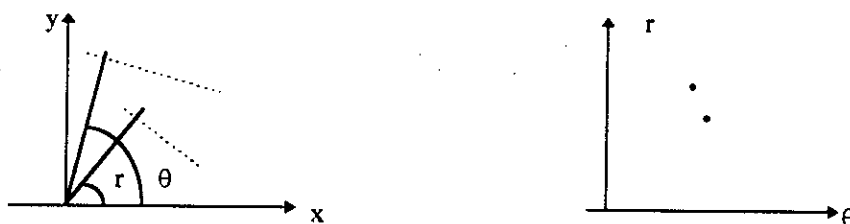


Fig. 32. Transformée de Hough.

III.4: Approximation par polynômes

Le principe est identique à celui vu dans l'approximation par des droites. On sous échantillonne le contour et on calcule les coefficients du polynôme désiré.

III.5. Approximation par des polygones

III.5.1. Méthode de la corde ou de Ramer

Cette méthode consiste à découper l'ensemble des points de contour en deux sous ensembles aussi longtemps que chaque sous ensemble ne peut être approché par un segment de droite avec une précision satisfaisante.

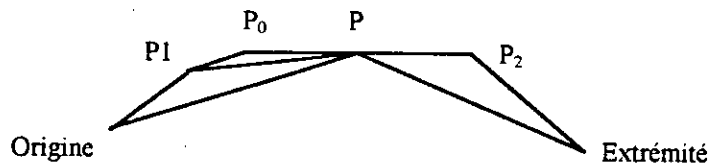


Fig. 33. Algorithme de Ramer.

Algorithme

Début

Seg [origine, extrémité], point-rupture \leftarrow origine, distmax \leftarrow 0 .

Pour chaque point $p \in$ [origine, extrémité]

Faire

si distance (p, seg) > distmax alors

point-rupture \leftarrow p et distmax \leftarrow distance(p, seg).

si distmax > seuil alors

approx (origine, point-rupture)

approx (point-rupture, extrémité)

sinon retour seg

Fsi.

Fp.

Fin.

Il existe d'autres algorithmes plus ou moins complexes tels que ceux de Roberts, Wall et Danielson...[6].

Chapitre 5

*Réalisation du système de
reconnaissance*

I. Présentation de l'écriture arabe

I.1. Historique

L'écriture arabe a connu plusieurs styles qui se développèrent à travers les siècles. Parmi ces styles, une écriture angulaire de style koufique (de la ville de Koufa en Irak) a servi au début de l'islam à copier le Coran. Seulement à cause de sa structure complexe, elle est devenue par la suite une écriture décorative. Une autre structure plus souple appelée le Naskhi est venue ensuite pour remplacer rapidement la première. Cette écriture est la plus utilisée de nos jours par les imprimeries, les machines à écrire et les logiciels de traitement de texte.

Les traits essentiels qui caractérisent la calligraphie du Naskhi est la continuité et la cursivité (ronde). Les formes curvilignes révèlent un grand mouvement ondulatoire imprimé par la main qui progressent continuellement de droite à gauche et de haut en bas donnant ainsi un espace planaire plus important par rapport aux écritures latines. De même la jonction entre les lettres constitue une caractéristique propre à l'arabe imprimé. La nature de la liaison varie en fonction de deux formes voisines. Cette continuité engendre des altérations de formes curvilignes de base et certains assemblage de lettres appelés ligature qui sont fréquemment utilisés dans les textes arabes. Une ligature est une combinaison particulière de caractères tracés et liés l'un au dessus de l'autre.



Fig. 34. Exemple de ligatures.

I.2. Styles d'écriture

Les styles d'écriture les plus représentatifs et les plus utilisés sont par ordre d'importance le Naskhi, le Koufi, le Rekkai [8].

I.2.1. Style Naskhi

La majorité des écritures actuelles d'imprimeries sont une transpiration directe de ce style. Il est caractérisé comme nous l'avons déjà cité par une forte cursivité et une simplicité comparativement aux autres écritures. C'est ce principal style qui sera utilisé ultérieurement dans notre étude.

I.2.2. Style Koufi

C'est le style d'écriture le plus ancien. Il se présente comme une écriture à tendance rectangulaire essentiellement utilisée comme écriture ornementale. Ce style se

caractérisé par des lettres avec des hauteurs centrales fortement prononcées et une épaisseur constante des traits.

I.2.3. Style Rekkai

Né en Turquie, il se présente comme une écriture très simplifiée caractérisée par des formes peu cursives, on cite par exemple la disparition des dents de scie de la lettre sin au début et au milieu du mot.

العلم نور
العلم نور
العلم نور
العلم نور
العلم نور
العلم نور

Fig. 35. Quelques styles utilisés en langue arabe.

II. L'alphabet arabe

II.1. Les consonnes

L'alphabet arabe comprend 28 caractères de base. Le caractère "Hamza" : ء non considéré comme tel joue un rôle important dans la phonétique. Il peut être utilisé comme une lettre ordinaire indépendante, placé au dessus des lettres considérées comme des voyelles longues, au dessous de lettre "Alif", au dessus de la forme finale et médiane de "Ya" (Fig. 36).

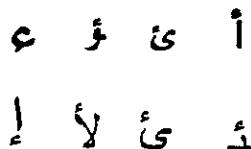


Fig. 36. Différentes positions de la lettre hamza.

Un autre caractère qui est considéré comme le 29^{ème} caractère de l'alphabet est le "Lam-alif" : qui n'est autre que le graphisme contracté des deux lettres "Lam" et "Alif".

Chaque caractère peut s'écrire selon deux à quatre formes différentes en fonction de son contexte dans le mot:

- Initiale: au début du mot.
- Médiane : au milieu du mot.
- Finale : à la fin du mot et lié à la lettre précédente.
- Isolée: non lié.

| Isolée | Début | Milieu | Finale |
|--------|-------|--------|--------|
| ا | ا | ا | ا |
| آ | آ | آ | آ |
| أ | أ | أ | أ |
| ؤ | ؤ | ؤ | ؤ |
| ي | ي | ي | ي |
| ى | ى | ى | ى |

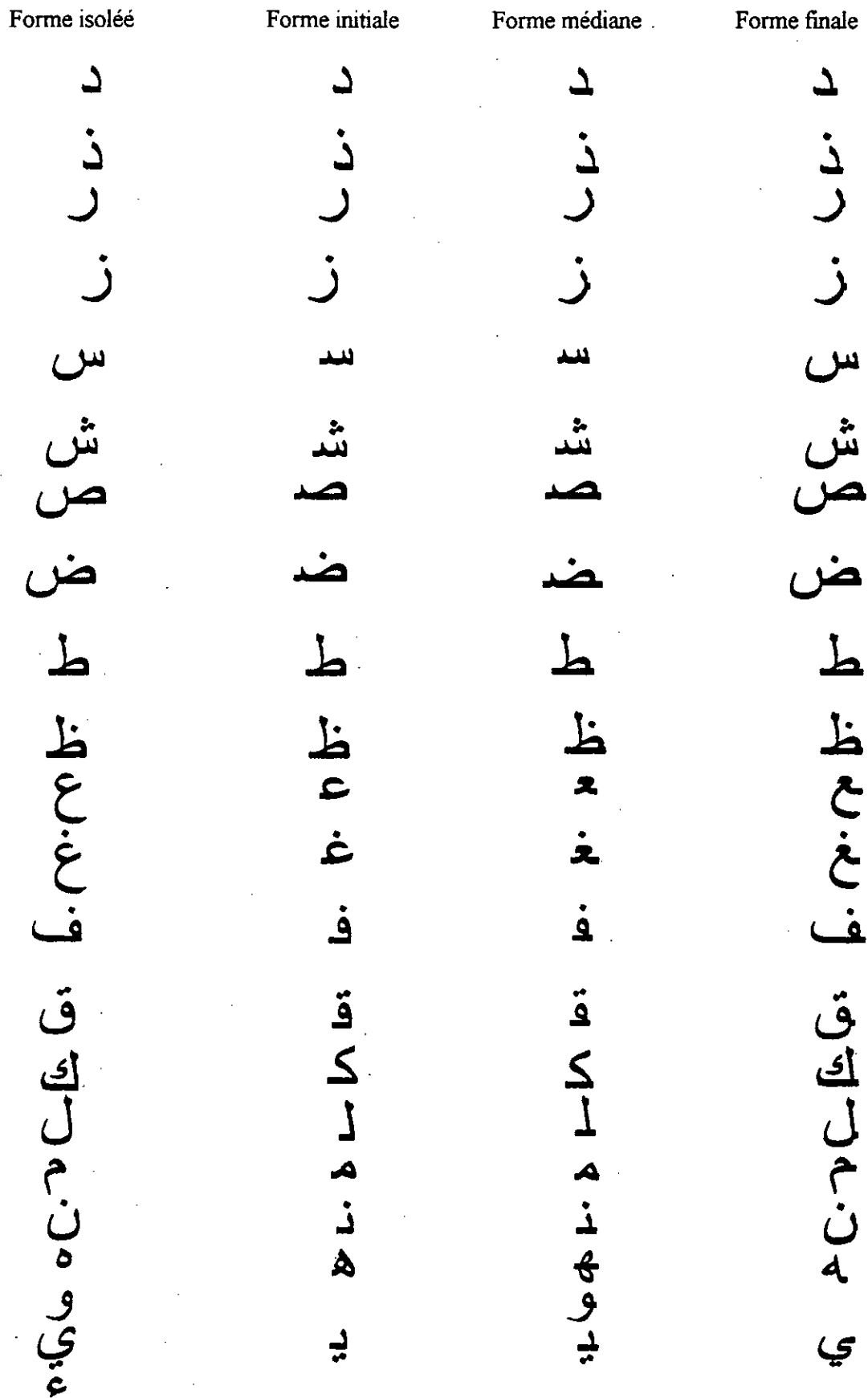


Fig. 37. Différentes formes des caractères dans un mot.

Six caractères parmi les 28 ont seulement deux formes distinctes initiale et finale: alif, dal, dhal, zin, ra et waw.

ز ا و ر ذ د

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.




| Tracé | Caractère |
|-------|-----------|
| ح | ج ح خ |
| س | ش س |
| ع | ع غ |
| ر | ز ر |
| ب | ب ث ت |




Fig. 38. Caractères ayant même tracé principal.

II.2. Les voyelles

Les voyelles arabes les plus utilisées sont : Fetha, Kesra et Damma. Ces voyelles sont symbolisées par des accents. On notera aussi que trois caractères peuvent se comporter comme des voyelles longues dans certaines de leur position dans le mot (alif, waw et ya).

Dans l'imprimerie les voyelles citées ne sont pas utilisées sauf dans les livres scolaires de primaire ou ceux des contes pour enfants afin de leur apprendre à lire correctement.

1. Fetha : 
2. kesra : 
3. Damma : 

4. Wasla : 
5. Mada : 
6. Tanwin : 

III. Reconnaissance des caractères arabes

III.1. Choix de la méthode

Le but de notre travail est de reconnaître des caractères arabes de certaines fontes préalablement apprises (système multifonte). Pour cela, nous avons choisi une méthode structurelle qui ne nécessite pas de squelettisation et les algorithmes classiques de suivi du contour. Elle consiste à analyser les frontières des caractères pour extraire leurs caractéristiques géométriques et topologiques.

Le système de reconnaissance réalisé est basé sur deux niveaux, le premier niveau permet de grouper les caractères ayant les mêmes caractéristiques principales dans une même classe et le second appelé identification a pour rôle de comparer un caractère inconnu à un ensemble de prototypes stockés en mémoire et appelé *dictionnaire*.

III.2. Acquisition des caractères

L'acquisition des caractères est obtenue par un scanner à plat de type HP ScanJet IIP de la compagnie Hewlett Packard. Le stockage de l'image se fait par un logiciel appelé Deskscan IIP qui travaille sous windows.

Grâce à ce logiciel, l'image est directement sauvegardée en code binaire: chaque pixel est codé sur un bit (1 ou 0). La résolution du scanner est fixée à 300 dpi (points par pouce) afin d'assurer une bonne qualité de l'image acquise. Cette dernière est transformée en code ASCII sous un fichier graphique de type TIFF. La lecture de ce fichier nécessite la connaissance de sa structure.

Comme notre image est à deux niveaux de gris (en noir et blanc), nous allons lire seulement les champs qui nous intéressent, à savoir l'adresse de l'IFD (entête du fichier), la longueur de l'image (champs 257), sa largeur (champs 256), l'adresse du premier pixel de cette image (champs 273), et enfin la couleur attribuée aux pixels (champs 262).

III.3. Filtrage

L'image scannée présente souvent du bruit dû essentiellement au scanner lui-même, à la qualité du papier et surtout au type d'imprimante utilisée.

Au début, nous avons essayé les algorithmes des filtres morphologiques et celui de Tuckey. Nous avons constaté que les premiers filtres suppriment effectivement le bruit mais ils déforment complètement le caractère comme nous l'avons mentionné dans de la figure 19 et 20 du chapitre 3. Le second est le filtre médian de Tuckey qui donne de bons résultats en général mais laisse néanmoins certains points non utiles dans l'image. Enfin nous avons choisi un filtrage réalisé par Guendouz [16] vu son efficacité et sa simplicité et dont l'algorithme est énoncé ci dessous.

Algorithme du filtrage**Début**

Faire

Mise à zéro des bords .

Si $image[i,j]=0$ alors:

Faire

Si le nombre des huit pixels voisins allumés est ≥ 4 alors $image[i,j] = 1$.
Sinon $image[i,j]=image[i,j]$.

Fsi

Si $image[i,j]=1$ alors:

Faire

Si le nombre des huit pixels voisins éteints est > 4 alors $image[i,j] = 0$.
Sinon $image[i,j]= image[i,j]$.

Fsi

Fin.

a) Image bruitée.



b) Filtre de Tuckey



c) Filtre choisi.

Fig.39. Filtrage de l'image.

III.4. Extraction des primitives

Nous avons vu précédemment que l'écriture arabe et plus précisément le style Naskhi est cursif et rond, c'est à dire qu'il présente des courbures plus ou moins importantes. Par conséquent, nous avons choisi le nombre de concavités et des boucles comme caractéristiques principales. Ces concavités sont détectées dans les quatre directions : à gauche , à droite, en haut et enfin en bas pour le caractère considéré.

L'extraction des concavités se fait en calculant les quatre profils du caractère. Ces derniers sont obtenus en comptant le nombre de pixels blancs situés entre le bord du cadre circonscrit au caractère et le premier pixel allumé. Ce nombre est calculé sur plusieurs hauteurs pour donner le profil recherché [6].

Exemple:

Calcul du profil gauche :

Algorithme



Fig. 40. Cadre circonscrit au caractère.

Début

Pour i allant de y_h à y_b

faire

$j = x_g$.

$P[i] = 0$.

Tant que $Im[i,j] \neq 1$

Faire

$P[i] = P[i] + 1$.

$j = j + 1$.

Fin de tant que.

Fp

Fin.

III.4.1. Lissage des profils

Pour lisser les profils et éliminer les perturbations ponctuelles, nous utiliserons un filtre moyen défini comme suit:

$$P[yh] = P[yh]$$

$$P[i] = \frac{P[i-1] + P[i] + P[i+1]}{3}$$

$$P[yb] = P[yb]$$



profil haut



profil gauche



profil droit



profil bas

Fig.41. Extraction des profils.

III.4.2. Détection des concavités

a) Définition d'une concavité

Une fonction f est dite concave dans un intervalle si et seulement si son graphe entre deux points de cet intervalle reste toujours au dessus ou sur la corde joignant les deux points.



Fig.42. Exemple de fonction concave entre a et b.

b) Détection des concavités

Pour extraire les concavités présentes dans chaque caractère, nous étudierons la variation de la courbe du profil $P[i]$ en fonction de i , ce qui revient donc à détecter les extremums locaux. Cette méthode est très rapide et facile à programmer et ne tient compte que des concavités profondes. Les fausses concavités dues au bruits sont éliminées en utilisant un seuil de profondeur. Ce seuil est déterminé expérimentalement en introduisant plusieurs caractères.

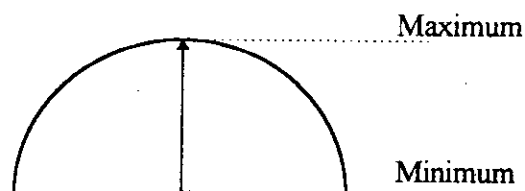


Fig.43. Profondeur d'une concavité.

Algorithme de détection des concavitésExemple: concavités gauchesDébut

Soient $P[i]$ le profil gauche du caractère à la ligne i , y_h la première ligne de la matrice contenant le caractère privé de sa ponctuation, et y_b la dernière ligne de cette matrice.

$N_c = 0$ (N_c : nombre de concavités).

Pour $i \leq y_b$ Faire:

Si $P[i+1] > P[i]$ alors

Faire

$i = i+1$.

Si $P[i+1] < P[i]$ alors

Faire

$\max = P[i]$.

$i = i + 1$.

Si $P[i+1] > P[i]$ alors

Faire

$\min = P[i]$.

Si $(\max - \min) \geq 12$ pixels

alors $N_c = N_c + 1$

Sinon $i = i+1$.

Sinon $i = i+1$.

Sinon $i = i+1$.

Fp.

Fin.

III.4.3. Détection des trous ou des boucles

Le nombre de trous ou des boucles constitue une des principales caractéristiques qui permet de distinguer les caractères entre eux. L'extraction de cette primitive nécessite souvent une squelettisation ou un suivi du contour. Dans ce dernier cas (suivi de contour), le nombre de trous dans une image est donnée par le nombre de contours fermés moins un [29].

$$N_b = N_{cf} - 1.$$

N_b : nombre de boucles.

N_{cf} : nombre de contours fermés.

Comme nous n'avons utilisé aucun prétraitement de type squelettisation ou suivi du contour, nous avons opté pour la méthode décrite par A. AMIN [5]. Cette opération se déroule suivant les étapes suivantes.

a) Recherche des transitions

Une première étape consiste à parcourir verticalement l'image et compter le nombre d'intersections d'une tranche verticale avec le caractère dépourvu de ses points isolés. Si ce nombre est supérieur ou égale à quatre, on peut alors prévoir l'existence d'une ou plusieurs boucles.

b) Détection des trous

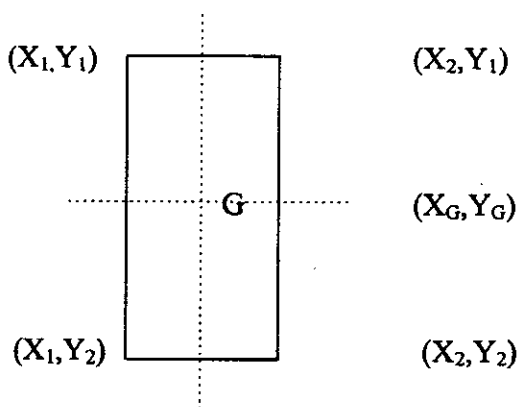
Une fois le nombre de transitions est connu, l'étape suivante consiste à déterminer le centre G de la zone blanche ou cavité telle que la position de ce point est donnée par les coordonnées X_G et Y_G .

$$X_G = X_1 + \frac{X_2 - X_1}{2}$$

X_1 et X_2 désignent les points de l'axe des abscisses limitant la tranche verticale.

Soient P_1 et P_2 deux points appartenant à la droite verticale $X = X_G$ et correspondant aux transitions 1-0 et 0-1 alors Y_G est calculé comme suit :

$$Y_G = Y_1 + \frac{Y_2 - Y_1}{2}$$



De cette façon, on cherche les demi-droites partant du nord, du sud, de l'est et de l'ouest du pixel blanc G et qui finissent par rencontrer un pixel noir du tracé principal du caractère.

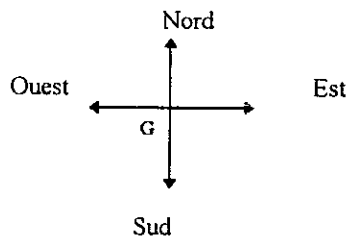


Fig.44. Les quatre demi- droites .

Exemple

Fig.45. Détection de deux trous.

Algorithme de détection des trousDébut

Pour extraire tous les trous existant dans l'image, on suit les étapes suivantes:

- $tr = 0$ (nombre de trous).
- Compter le nombre de transitions 1-0 ou 0-1 dans chaque colonne de la matrice image.
- Si ce nombre est supérieur ou égal à 4 alors:

Faire

1- Déterminer la largeur de la séquence dont le nombre de transitions est ≥ 4 . Si cette largeur est plus grande que quatre alors

Faire

- j_1 = début de la séquence.
- j_2 = fin de la séquence.

$$c) j_{moy} = j_1 + \frac{j_2 - j_1}{2}$$

d) i_1 : position ligne de la $n^{ème}$ transition.

e) i_2 : position ligne de la $n+1$ transition.

$$f) i_{moy} = i_1 + \frac{i_2 - i_1}{2}$$

g) A partir du point (i_{moy}, j_{moy})

Faire

$$j = j_{moy}$$

Répéter

$$j = j - 1.$$

Si $image[i_{moy}, j] = 1$ alors $j = j_{min}$ et
Ouest = true.

Jusqu'à $j = X_g$

$$j = j_{moy}$$

Répéter

$$j = j + 1.$$

Si $image[i_{moy}, j] = 1$ alors $j = j_{max}$ et
Est = true.

jusqu'à $j = X_d$

Si (ouest) and (est) = true alors $tr = tr + 1$.

Refaire les mêmes étapes pour toutes les séquences existantes.

Fin.

III.5. 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:

$$CI = 4^0 * Ntr + 4^1 * Ncg + 4^2 * Ncd + 4^3 * Nch + 4^4 * Ncb.$$

Ntr: nombre de trous

Ncg: nombre de concavités gauches.

Ncd: nombre de concavités droites.

Nch: nombre de concavités hautes.

Ncb: nombre de concavités basses.

Cette classification a été utilisée par **M.B. EI KURDY** [20]. 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.

D'après la figure ci dessous, on remarque pour les caractères arabes que les concavités basses sont négligeables ou n'existent pas, donc le dernier terme (Ncb) est toujours nul.

| Classe | Caractères |
|--------|--------------------|
| 0 | ا ت ا ث |
| 1 | م D O 0 ض ص م ه م |
| 2 | ه 8 B |
| 4 | ح ع ع C G F |
| 5 | ظ 6 |
| 6 | ه |
| 8 | ع غ ع ع E |
| 16 | ز ج ذ ح ر 7 ل ن ر |
| 17 | ق ف و م 9 |
| 20 | س Z 25 ج غ ج ح ك ح |
| 21 | ة ظ ه ط ط |
| 24 | ح ج |
| 64 | ي ن ب ي س U Y V |
| 65 | ن ف ق م ص |
| 68 | س ي ي K |
| 80 | ش د ك ل ل ر د |
| 81 | ف ض ل ا و م |
| 84 | x 1 |
| 128 | W |

Fig.46. Classification des caractères.

IV. Caractéristiques secondaires

Les caractéristiques principales que nous venons de voir servent essentiellement à définir les différentes classes possibles du dictionnaire. Mais dans la plupart des cas, nous trouverons dans la même classe plusieurs prototypes de caractères différents. Pour les distinguer, nous allons utiliser d'autres caractéristiques que nous appellerons *secondaires*. Ce sont des informations supplémentaires qui permettent de lever les ambiguïtés quand plusieurs caractères d'une même classe sont candidats. Ces caractéristiques sont : position des groupes de points, nombre de ces points, allongement du cadre contenant le corps principal et enfin la distributions des pixels dans les quatre angles du rectangle circonscrit au caractère.

IV.1. Position des points isolés

Certains caractères sont accompagnés d'un groupe de points variant de un jusqu'à trois éléments. Ces groupes de points jouent un rôle essentiel pour discriminer les caractères appartenant à la même famille de forme. Comme ils peuvent apparaître au dessus, au dessous ou au milieu de la ligne d'écriture, ils sont associés à un attribut de position (figure 47).



Trois points au dessus (code=1) un point au milieu (code=2) Deux points au dessous (code=3)

Fig. 47. Position des points .

La détermination de la position du groupe des points d'un caractère délimite le tracé de celui-ci par deux lignes : ligne₁ et ligne₂ comme l'illustre la figure 48.

Le cadre obtenu définit trois zones pouvant contenir ces points dont la position est égale à 1, 2, ou 3 selon que le groupe de points se situe au dessus, au milieu, ou au dessous du tracé principal. Elle vaut 0 dans le cas où le caractère est dépourvu des points. L'importance de cette caractéristique réside dans le cas où plusieurs caractères ont la même allure et le même nombre de points.

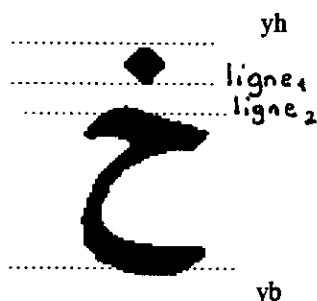


Fig. 48. Séparation des groupes des points.

Les caractères djim (ج), dhad (ظ) et kaf (ك) sont traités à part car la détection de leurs points ou de la partie connexe pour le kaf est très complexe. Pour cela, nous avons utilisé une projection partielle et verticale afin de pouvoir segmenter les deux parties à savoir le corps principal et le groupe de points ou el chara pour le caractère Kaf (forme isolée et finale).

L'axe de projection est choisi expérimentalement et après plusieurs essais sur plusieurs fontes. Nous notons enfin que pour les caractères djim et dhad, le cumul des pixels se fait de droite à gauche et dans le sens contraire pour le kaf.

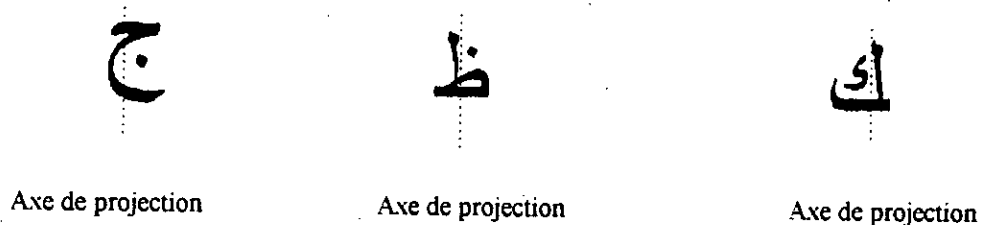


Fig.49. Détection des points milieu.

Algorithme de détection des points

Début

- Calculer la surface S_1 formée par les pixels allumés comprise entre y_h et ligne₁. La ligne₁ est la première ligne trouvée qui contient au plus 2 pixels allumés.
- Chercher la ligne₂ qui possède au moins 3 pixels allumés .
- Calculer la surface S_2 comprise entre ligne₂ et y_b .
- Si $S_1 < S_2$, $S_1 \geq 2$ et $S_2 > 2$ alors la position des points est au dessus, sinon elle est au dessous.
- Fin.

IV.2. Nombre des points isolés

Quinze caractères arabes parmi les vingt et huit de l'alphabet comptent de 1 à 3 points . Cette caractéristique est intéressante dans le cas où le nombre de caractères d'une classe ayant la même position des points est supérieur à un comme pour le ta et tha.

Le nombre de ces points est déterminé par le calcul de la surface formant le groupe de ces points et la forme du cadre englobant ces derniers (fig.50). Des seuils sont fixés pour trouver dans chaque cas le nombre de points de chaque caractère lorsque la variable logique ex-pt (existence de points) est vraie.



Fig. 50. Forme du cadre des points recherchés.

Algorithme

Début

Soient S_1 et S_2 les surfaces calculées précédemment.

- On calcule le rapport des surfaces $r = \frac{S_2}{S_1}$.
- On encadre les points détectés par un rectangle de côtés dx et dy .
- Si la variable position vaut 3 (au dessous), le nombre de points (Nb_pt) est soit 1 soit 2.
- Si la forme du rectangle contenant ces points est presque carrée alors le $Nb_pt = 1$, sinon il vaut 2.
- Si la variable position est 1 alors si le cadre contenant les points est rectangulaire alors $Nb_pt = 2$ sinon on obtient les conditions suivantes sur le rapport r
 - Si $r > 3$ (pour les caractères petits) alors $Nb_pt = 1$ sinon $Nb_pt = 3$.
 - Si $r > 6$ (pour les grands caractères) alors $Nb_pt = 1$ sinon $Nb_pt = 3$.

Fin

Remarque

Les caractères petits ont une taille inférieure à 56 pixels. Alors que les grands caractères ont une taille supérieure à 55 pixels.

IV.3. Allongement du cadre circonscrit

Dans l'alphabet arabe, plusieurs caractères ont les mêmes caractéristiques structurales telles que: Ra (isolé) \curvearrowright et dal (isolé) \curvearrowleft . Pour les différencier, nous utiliserons un test portant sur la forme du cadre contenant le tracé principal défini comme le rapport de L_y sur L_x (figure 51). Il en découle trois types de formes : Verticale, Horizontale et Carrée.



Fig. 51. Forme du tracé des caractères.

IV.4. Distribution des pixels dans les quatre coins du cadre

Cette caractéristique permet de savoir si les quatre sommets du rectangle contenant le corps principal du caractère sont pleins ou non. Pour cela, nous calculerons le nombre de pixels allumés situés dans une surface rectangulaire de longueur et largeur fixes ($L_x/6$ et $L_y/6$) [20].



Fig. 52. Distribution des points.

Dans chaque zone, on affecte un poids qui permet de codifier la position des régions pleines ou vides. Les poids 1, 2, 4 et 8 sont des puissances de 2.

Si le nombre de pixel allumés est supérieur à un certain seuil, on considère alors que la région est pleine et on définit dans ce cas une variable s qui vaut 1. Sinon la zone est supposée vide et s prend la valeur 0. Le poids total est calculé comme suit:

$$P = s_1 \cdot 2^0 + s_2 \cdot 2^1 + s_3 \cdot 2^2 + s_4 \cdot 2^3$$

$$P = \sum_{i=1}^4 s_i \cdot 2^{i-1}$$

Avec i désignant la position du coin considéré (figure 52).

Cette propriété sert à séparer les caractères ayant les même formes géométriques tels que le chiffre 5 et le caractère kaf (position début).

V. Dictionnaire

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

$$d_1(c, x) = |pp_c - pp_x|$$

$$d_2(c, x) = |Np_c - Np_x|$$

$$d_3(c, x) = |All_c - All_x|$$

$$d_4(c, x) = |P_c - P_x|$$

x: caractère à reconnaître

c: caractère candidat

pp: position des points

Np: nombre de points

P: poids du caractère.

All: allongement du caractère.

Nous déterminons alors la liste des candidats satisfaisant les conditions suivantes:

$$d_1(c, x) = 0$$

$$d_2(c, x) \leq \text{seuil}_1$$

$$d_3(c, x) \leq \text{seuil}_2$$

$$d_4(c, x) \leq \text{seuil}_3$$

Les seuil_1 , seuil_2 , seuil_3 sont déterminés expérimentalement. A ce stade du traitement, l'algorithme de reconnaissance a trois possibilités:

- Si la liste est vide, le système décide que le caractère est rejeté.
- Si la liste contient un seul élément, le système identifie le caractère immédiatement.
- Si la liste contient plus d'un élément, nous passerons à la troisième étape qui détermine le bon caractère. Cette étape consiste à extraire d'autres caractéristiques annexes pour éliminer les ambiguïtés telles que la position des concavités et les segments de droites.

Certains caractères sont difficiles à séparer tels que **2**, **Z**, **S** et **5** de la classe 20. La première caractéristique supplémentaire détecte la position des concavités comme par exemple le chiffre **2** qui contient une concavité gauche supérieure et une concavité droite inférieure tandis que **5** possède une concavité droite supérieure et une concavité gauche inférieure.

La deuxième caractéristique permet de séparer les caractères **2** et **Z** ainsi que **5** et **S**. La lettre **Z** possède deux segments de droites horizontaux par contre le chiffre **2** ne contient qu'un seul.

Enfin si la liste contient toujours plus d'un candidats nous déciderons que le caractère est ambigu.

Algorithme de détection des segments de droite horizontaux

Début

- Pour chaque ligne de la matrice image, nous comptons le nombre de pixels allumés.
- Si ce nombre est proche de la largeur de l'image et si ce nombre se trouve sur au moins quatre lignes successives alors nous obtenons un segments horizontal.
- Fin.

Conclusion

Les résultats obtenus par notre méthode sont les suivants:

- Taux de reconnaissance: 98,28%.
- Taux de confusion : 1,7%.
- Taux de rejet: 0%

Ces résultats peuvent être améliorés en introduisant d'autres caractéristiques sans alourdir le système. La confusion entre fai isolée et dad position début peut être annulée en déclarant une variable logique "lié" qui doit vraie lorsque le caractère est lié et fausse dans le cas isolé.

Conclusion

Plusieurs méthodes (statistiques et structurelles) parmi les plus usuelles en reconnaissance de caractères ont été utilisées au laboratoire de recherches de l'ENP. Ces méthodes présentaient souvent un bon taux de reconnaissance mais les prétraitements utilisés (squelettisation et détection des contours) sont lents et par conséquent donnent une vitesse de reconnaissance non satisfaisante. Ces prétraitements ont été testés au début de notre étude. Ils présentent très fréquemment des déformations de la forme et en plus ils ne permettent pas de réduire réellement l'épaisseur à un pixel. La squelettisation nécessite souvent un autre traitement supplémentaire pour pouvoir suivre correctement le tracé du caractère. Généralement les travaux effectués récemment en OCR utilisent l'approximation par des polygones après amincissement [2].

Les méthodes statistiques ne peuvent traiter qu'une seule fonte à la fois donc le logiciel se trouve limité. Pour pallier à toutes ces limitations une autre approche a été adoptée; son but est de traiter plusieurs fontes différentes et plusieurs styles et aussi d'améliorer la vitesse de reconnaissance. Cette approche consiste à déterminer les frontières du caractère ou ses quatre profils : gauche, droit, haut et bas [6] pour l'extraction des concavités dans les quatre directions possibles (hautes, basses, gauches et droites) et la recherche des cavités pour obtenir les trous ou les boucles. D'autres caractéristiques secondaires sont nécessaires pour diminuer les ambiguïtés qui peuvent exister.

Le système ainsi réalisé est basé sur deux niveaux. Le 1^{er} niveau permet la classification de tous les caractères ayant les mêmes caractéristiques principales dans un même groupe et le second niveau est l'étape de reconnaissance et de décision. Cette façon de procéder accélère considérablement le processus de reconnaissance.

Les résultats obtenus par notre système sont jugés satisfaisants puisque on atteint un pourcentage d'identification de 98,28 %, la rejection est de 0% tandis que le taux de confusion est de 1.7%. Les caractères confus sont Y et V pour le latin et dad position début avec le fa isolé. Le temps de reconnaissance dépend de la complexité de la classe adressée et de la manière d'organiser le dictionnaire. Le temps moyen obtenu avec notre méthode est très satisfaisant et il ne dépasse pas les 8 centièmes de seconde sur un micro-ordinateur de type 486 DX 4. Ce système peut être amélioré en introduisant d'autres fontes pour s'approcher d'un système omnifonte.

Bibliographie

- [1] T.L. Abégnoli, " Quatre phases de la reconnaissance des caractères ",
Electronique, n°29, pp 38- 40, juin 93.
- [2] I.S.I. Abuhaiba, "Processing of off line handwritten text: Polygonal approximation
and enforcement of temporal information", Graphical models and image processing,
vol 56, n°4, pp 324-335, juillet 94.
- [3] I.S.I. Abuhaiba, " Recognition of handwritten cursive arabic characters", IEEE
transaction on P.A.M.I, vol 16, n°6, 9p, juin 94.
- [4] H.Al-Yousefi et S.S.Udpa, " Recognition of arabic characters", IEEE transaction on
P.A.M.I, vol 14, n°8, 5p, Août 92.
- [5] Adnan. amin, " IRAC: un système pour la reconnaissance et la compréhension de
l'arabe écrit et imprimé", Thèse de doctorat d'état es-sciences, université de
Nancy1, Chapitre 5, pp104 - 144, 1985.
- [6] Abdel. Belaid et Y.Belaid, Reconnaissance des formes, Inter-Edition, Paris, 1992,
412p.
- [7] Abdelouaheb. Belaid, "A syntactic approach for handwritten mathematical formula
recognition", IEEE transaction on P.A.M.I, vol 6, n°1, pp 105 - 111, janvier 84.
- [8] 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.
- [9] Sophie Bercu, " Segmentation et reconnaissance en ligne de mots manuscrits",
IRISA, INRIA, université de Rennes 1, France, 32p, février 1993.
- [10] J.F. Duvier, " Structure des fichiers BMP, décoder les fichiers bitmap", Language
et système info-PC, n°81.
- [11] J.M. Chassery " Segmentation cytologique", Reconnaissance des formes et
Intelligence artificielle, Tome 1, AFCET 4^{ème} congrés, janvier 84.
- [12] Bruno. Ferre, " Reconnaissance ", Sciences et vie micro, n°71, pp 148 - 156, Avril
1990.
- [13] Daniel. Goldwaser, " Lecture optique, lecture de pages ", Technique de l'ingénieur,
Traité informatique, H 1410, 7p, 1988.
- [14] Habib. Goraine, " Off-Line arabic character recognition", Computer, 4p, 1992.
- [15] Djamilia. Guamaz, " Reconnaissance de caractères arabe isolés manuscrits par la
méthode de masque ", Thèse de Magister, CDTA, 1993.

- [16] M. Guendouz, " Reconnaissance de bulletins scolaires", mémoire d'ingénieur, institut d'informatique, USTHB, 1994.
- [17] F. haj-Hassan et W. Haj-Ali, " Printed arabic text recognition", The arabian journal for science and engineering, vol 16 n°4, pp 511 -518, Octobre 1991.
- [18] Claudie Faure et Laurent Miclet, " Intelligence artificielle et reconnaissance des formes", Techniques de l'ingénieur, Traité informatique, H 1920, 15p, 1991.
- [19] Kok-Lay. Heng, " Analyse d'images de lignes: des pixels aux primitives", Thèse de doctorat d'état, Paris 6. 1988, 90p.
- [20] M.B.kurdy, " Multifont arabic/ latin optical character recognition system, Langue arabe et technologies informatiques avancées, pp 245 - 255, 1993.
- [21] F.Matougui, " Conception et réalisation d'un système interactif de reconnaissance de caractères arabes manuscrits: SIRCAM ", Thèse de Magister, CDTA, Juin 90.
- [22] L.Miclet, " Méthodes structurelles pour la reconnaissance des formes", Edition Eyrolles, 1984.
- [23] Shunji.Mori, " Historical review of OCR research and development, Proceedings of IEEE, vol 80, n°7, Juillet 92.
- [24] Dj. Ould - Brahim, " Etude d'un nouveau modèle de primitives pour la description des images numériques: Le macro-pixel ", Thèse de docteur ingénieur, université des sciences et techniques de Lille, Flandres Artois, 1988.
- [25] L.Personnaz, " Réseaux de neurones et reconnaissance des formes", Ecole de printemps, NSI 91, Chapitre 6, 31p, 1991.
- [26] A.Poor, " Looking at the TIFF specification from the inside ", PC magazine, Décembre 1991.
- [27] L.Saâdaoui, " Techniques de traitement numérique d'images en vue de la RF ", Thèse de Magister, ENP, 1991.
- [28] C.Souchard, " Les formats des fichiers image sur PC ", PC expert, pp141 - 144, Juillet-Août 92.
- [29] J.J.Toumazet, Traitement de l'image sur micro-ordinateur, France, 1987.

Annexe

Caractères utilisés dans le dictionnaire

Isolée

א ב ג ד ה ו ז ח ט י כ ל מ נ ס ע פ צ ק ר ש ת יו יז יח יט

Début

א. ב. ג. ד. ה. ו. ז. ח. ט. י. כ. ל. מ. נ. ס. ע. פ. צ. ק. ר. ש. ת. יו. יז. יח. יט.

Milieu

א. ב. ג. ד. ה. ו. ז. ח. ט. י. כ. ל. מ. נ. ס. ע. פ. צ. ק. ר. ש. ת. יו. יז. יח. יט.

Finale

א. ב. ג. ד. ה. ו. ז. ח. ט. י. כ. ל. מ. נ. ס. ע. פ. צ. ק. ר. ש. ת. יו. יז. יח. יט.

ق.ر.و.ج.ف.ك.ل.م.ن.ه.و.ي.ح.ط.ز

ق.ر.و.ج.ف.ك.ل.م.ن.ه.و.ي.ح.ط.ز

ق.ر.و.ج.ف.ك.ل.م.ن.ه.و.ي.ح.ط.ز

ق.ر.و.ج.ف.ك.ل.م.ن.ه.و.ي.ح.ط.ز

Autres formes de caractères non séparables

ق ر و

Chiffres arabes

0 1 2 3 4 5 6 7 8 9

Caractères latins majuscule (Times New Roman)

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Quelques caractères arabes de différentes fontes

ا ب ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ف ق
ك ل م ن . و ي . ة ؤ ي

ا ب ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ
ع غ ف ق ك ل م ن ه و
ي . ؤ

ا ب ت ث ج خ د ذ ر ز س ش ص ض ط ظ ع
غ ف ق ك ل م ن ه و ي

Fichier dl.dic (Caractères sans points)

xxx: case vide.

| N°= | car | pos_pt | nb_pt | poids | all | taille |
|-----|-------|--------|-------|-------|-----|--------|
| 0 | alif | 0 | 0 | 15 | 1 | 57 |
| 1 | hamza | 0 | 0 | 8 | 1 | 17 |
| 2 | alif | 0 | 0 | 5 | 1 | 42 |
| 3 | hamza | 0 | 0 | 10 | 2 | 14 |
| 4 | alif | 0 | 0 | 5 | 1 | 30 |
| 5 | alifi | 0 | 0 | 5 | 1 | 75 |
| 6 | alifm | 0 | 0 | 5 | 1 | 76 |
| 7 | I | 0 | 0 | 15 | 1 | 42 |
| 8 | T | 0 | 0 | 9 | 2 | 41 |
| 9 | sini | 0 | 0 | 10 | 3 | 21 |
| 10 | mimi | 0 | 0 | 10 | 1 | 32 |
| 11 | hamza | 0 | 0 | 2 | 1 | 13 |
| 12 | L | 0 | 0 | 7 | 1 | 42 |
| 13 | hai | 0 | 0 | 10 | 2 | 16 |
| 14 | xxx | 0 | 0 | 0 | 0 | 0 |
| 15 | xxx | 0 | 0 | 0 | 0 | 0 |
| 16 | xxx | 0 | 0 | 0 | 0 | 0 |
| 17 | xxx | 0 | 0 | 0 | 0 | 0 |
| 18 | xxx | 0 | 0 | 0 | 0 | 0 |
| 19 | xxx | 0 | 0 | 0 | 0 | 0 |
| 20 | mimi | 0 | 0 | 10 | 1 | 63 |
| 21 | hai | 0 | 0 | 14 | 2 | 33 |
| 22 | mimi | 0 | 0 | 10 | 1 | 42 |
| 23 | hai | 0 | 0 | 6 | 2 | 22 |
| 24 | hai | 0 | 0 | 10 | 2 | 16 |
| 25 | hai | 0 | 0 | 2 | 1 | 37 |
| 26 | mimd | 0 | 0 | 6 | 3 | 34 |
| 27 | şadd | 0 | 0 | 14 | 3 | 35 |
| 28 | D | 0 | 0 | 11 | 2 | 41 |
| 29 | O | 0 | 0 | 15 | 2 | 44 |
| 30 | P | 0 | 0 | 11 | 1 | 42 |
| 31 | R | 0 | 0 | 15 | 2 | 42 |
| 32 | 0 | 0 | 0 | 15 | 1 | 47 |
| 33 | 4 | 0 | 0 | 0 | 1 | 41 |
| 34 | Q | 0 | 0 | 5 | 2 | 46 |
| 35 | xxx | 0 | 0 | 0 | 0 | 0 |
| 36 | xxx | 0 | 0 | 0 | 0 | 0 |
| 37 | xxx | 0 | 0 | 0 | 0 | 0 |
| 38 | xxx | 0 | 0 | 0 | 0 | 0 |
| 39 | xxx | 0 | 0 | 0 | 0 | 0 |
| 40 | had | 0 | 0 | 6 | 1 | 53 |
| 41 | B | 0 | 0 | 15 | 1 | 41 |
| 42 | 8 | 0 | 0 | 15 | 1 | 48 |
| 43 | xxx | 0 | 0 | 0 | 0 | 0 |
| 44 | xxx | 0 | 0 | 0 | 0 | 0 |
| 45 | xxx | 0 | 0 | 0 | 0 | 0 |
| 46 | xxx | 0 | 0 | 0 | 0 | 0 |
| 47 | xxx | 0 | 0 | 0 | 0 | 0 |
| 48 | xxx | 0 | 0 | 0 | 0 | 0 |
| 49 | xxx | 0 | 0 | 0 | 0 | 0 |
| 50 | xxx | 0 | 0 | 0 | 0 | 0 |
| 51 | xxx | 0 | 0 | 0 | 0 | 0 |
| 52 | xxx | 0 | 0 | 0 | 0 | 0 |
| 53 | xxx | 0 | 0 | 0 | 0 | 0 |
| 54 | xxx | 0 | 0 | 0 | 0 | 0 |
| 55 | xxx | 0 | 0 | 0 | 0 | 0 |

| | | | | | | |
|-----|-------|---|---|----|---|----|
| 56 | xxx | 0 | 0 | 0 | 0 | 0 |
| 57 | xxx | 0 | 0 | 0 | 0 | 0 |
| 58 | xxx | 0 | 0 | 0 | 0 | 0 |
| 59 | xxx | 0 | 0 | 0 | 0 | 0 |
| 60 | hhai | 0 | 0 | 15 | 1 | 42 |
| 61 | aini | 0 | 0 | 6 | 1 | 50 |
| 62 | hhai | 0 | 0 | 15 | 1 | 32 |
| 63 | aini | 0 | 0 | 6 | 1 | 38 |
| 64 | hamza | 0 | 0 | 2 | 2 | 40 |
| 65 | aind | 0 | 0 | 6 | 3 | 42 |
| 66 | C | 0 | 0 | 15 | 2 | 43 |
| 67 | F | 0 | 0 | 11 | 1 | 42 |
| 68 | G | 0 | 0 | 15 | 2 | 44 |
| 69 | L | 0 | 0 | 7 | 1 | 42 |
| 70 | taai | 0 | 0 | 6 | 3 | 29 |
| 71 | Q | 0 | 0 | 5 | 2 | 46 |
| 72 | xxx | 0 | 0 | 0 | 0 | 0 |
| 73 | xxx | 0 | 0 | 0 | 0 | 0 |
| 74 | xxx | 0 | 0 | 0 | 0 | 0 |
| 75 | xxx | 0 | 0 | 0 | 0 | 0 |
| 76 | xxx | 0 | 0 | 0 | 0 | 0 |
| 77 | xxx | 0 | 0 | 0 | 0 | 0 |
| 78 | xxx | 0 | 0 | 0 | 0 | 0 |
| 79 | xxx | 0 | 0 | 0 | 0 | 0 |
| 80 | 6 | 0 | 0 | 15 | 1 | 48 |
| 81 | taai | 0 | 0 | 6 | 3 | 29 |
| 82 | xxx | 0 | 0 | 0 | 0 | 0 |
| 83 | xxx | 0 | 0 | 0 | 0 | 0 |
| 84 | xxx | 0 | 0 | 0 | 0 | 0 |
| 85 | xxx | 0 | 0 | 0 | 0 | 0 |
| 86 | xxx | 0 | 0 | 0 | 0 | 0 |
| 87 | xxx | 0 | 0 | 0 | 0 | 0 |
| 88 | xxx | 0 | 0 | 0 | 0 | 0 |
| 89 | xxx | 0 | 0 | 0 | 0 | 0 |
| 90 | xxx | 0 | 0 | 0 | 0 | 0 |
| 91 | xxx | 0 | 0 | 0 | 0 | 0 |
| 92 | xxx | 0 | 0 | 0 | 0 | 0 |
| 93 | xxx | 0 | 0 | 0 | 0 | 0 |
| 94 | xxx | 0 | 0 | 0 | 0 | 0 |
| 95 | xxx | 0 | 0 | 0 | 0 | 0 |
| 96 | xxx | 0 | 0 | 0 | 0 | 0 |
| 97 | xxx | 0 | 0 | 0 | 0 | 0 |
| 98 | xxx | 0 | 0 | 0 | 0 | 0 |
| 99 | xxx | 0 | 0 | 0 | 0 | 0 |
| 100 | ham | 0 | 0 | 0 | 1 | 61 |
| 101 | xxx | 0 | 0 | 0 | 0 | 0 |
| 102 | xxx | 0 | 0 | 0 | 0 | 0 |
| 103 | xxx | 0 | 0 | 0 | 0 | 0 |
| 104 | xxx | 0 | 0 | 0 | 0 | 0 |
| 105 | xxx | 0 | 0 | 0 | 0 | 0 |
| 106 | xxx | 0 | 0 | 0 | 0 | 0 |
| 107 | xxx | 0 | 0 | 0 | 0 | 0 |
| 108 | xxx | 0 | 0 | 0 | 0 | 0 |
| 109 | xxx | 0 | 0 | 0 | 0 | 0 |
| 110 | xxx | 0 | 0 | 0 | 0 | 0 |
| 111 | xxx | 0 | 0 | 0 | 0 | 0 |
| 112 | xxx | 0 | 0 | 0 | 0 | 0 |
| 113 | xxx | 0 | 0 | 0 | 0 | 0 |
| 114 | xxx | 0 | 0 | 0 | 0 | 0 |
| 115 | xxx | 0 | 0 | 0 | 0 | 0 |
| 116 | xxx | 0 | 0 | 0 | 0 | 0 |
| 117 | xxx | 0 | 0 | 0 | 0 | 0 |
| 118 | xxx | 0 | 0 | 0 | 0 | 0 |
| 119 | xxx | 0 | 0 | 0 | 0 | 0 |

| | | | | | | |
|-----|------|---|---|----|---|----|
| 120 | aini | 0 | 0 | 7 | 1 | 98 |
| 121 | E | 0 | 0 | 15 | 1 | 42 |
| 122 | aini | 0 | 0 | 6 | 1 | 42 |
| 123 | xxx | 0 | 0 | 0 | 0 | 0 |
| 124 | xxx | 0 | 0 | 0 | 0 | 0 |
| 125 | xxx | 0 | 0 | 0 | 0 | 0 |
| 126 | xxx | 0 | 0 | 0 | 0 | 0 |
| 127 | xxx | 0 | 0 | 0 | 0 | 0 |
| 128 | xxx | 0 | 0 | 0 | 0 | 0 |
| 129 | xxx | 0 | 0 | 0 | 0 | 0 |
| 130 | xxx | 0 | 0 | 0 | 0 | 0 |
| 131 | xxx | 0 | 0 | 0 | 0 | 0 |
| 132 | xxx | 0 | 0 | 0 | 0 | 0 |
| 133 | xxx | 0 | 0 | 0 | 0 | 0 |
| 134 | xxx | 0 | 0 | 0 | 0 | 0 |
| 135 | xxx | 0 | 0 | 0 | 0 | 0 |
| 136 | xxx | 0 | 0 | 0 | 0 | 0 |
| 137 | xxx | 0 | 0 | 0 | 0 | 0 |
| 138 | xxx | 0 | 0 | 0 | 0 | 0 |
| 139 | xxx | 0 | 0 | 0 | 0 | 0 |
| 140 | rai | 0 | 0 | 10 | 2 | 49 |
| 141 | kafi | 0 | 0 | 14 | 2 | 59 |
| 142 | dali | 0 | 0 | 6 | 2 | 20 |
| 143 | hhad | 0 | 0 | 6 | 3 | 36 |
| 144 | lamd | 0 | 0 | 14 | 1 | 79 |
| 145 | kaff | 0 | 0 | 14 | 3 | 79 |
| 146 | 3 | 0 | 0 | 15 | 1 | 47 |
| 147 | 7 | 0 | 0 | 9 | 1 | 46 |
| 148 | rai | 0 | 0 | 10 | 2 | 24 |
| 149 | dali | 0 | 0 | 6 | 2 | 28 |
| 150 | ram | 0 | 0 | 2 | 3 | 63 |
| 151 | xxx | 0 | 0 | 0 | 0 | 0 |
| 152 | xxx | 0 | 0 | 0 | 0 | 0 |
| 153 | xxx | 0 | 0 | 0 | 0 | 0 |
| 154 | xxx | 0 | 0 | 0 | 0 | 0 |
| 155 | xxx | 0 | 0 | 0 | 0 | 0 |
| 156 | xxx | 0 | 0 | 0 | 0 | 0 |
| 157 | xxx | 0 | 0 | 0 | 0 | 0 |
| 158 | xxx | 0 | 0 | 0 | 0 | 0 |
| 159 | xxx | 0 | 0 | 0 | 0 | 0 |
| 160 | 9 | 0 | 0 | 15 | 1 | 48 |
| 161 | mimi | 0 | 0 | 10 | 1 | 71 |
| 162 | mimf | 0 | 0 | 2 | 1 | 72 |
| 163 | wawi | 0 | 0 | 10 | 2 | 51 |
| 164 | wawi | 0 | 0 | 10 | 1 | 28 |
| 165 | xxx | 0 | 0 | 0 | 0 | 0 |
| 166 | xxx | 0 | 0 | 0 | 0 | 0 |
| 167 | xxx | 0 | 0 | 0 | 0 | 0 |
| 168 | xxx | 0 | 0 | 0 | 0 | 0 |
| 169 | xxx | 0 | 0 | 0 | 0 | 0 |
| 170 | xxx | 0 | 0 | 0 | 0 | 0 |
| 171 | xxx | 0 | 0 | 0 | 0 | 0 |
| 172 | xxx | 0 | 0 | 0 | 0 | 0 |
| 173 | xxx | 0 | 0 | 0 | 0 | 0 |
| 174 | xxx | 0 | 0 | 0 | 0 | 0 |
| 175 | xxx | 0 | 0 | 0 | 0 | 0 |
| 176 | xxx | 0 | 0 | 0 | 0 | 0 |
| 177 | xxx | 0 | 0 | 0 | 0 | 0 |
| 178 | xxx | 0 | 0 | 0 | 0 | 0 |
| 179 | xxx | 0 | 0 | 0 | 0 | 0 |
| 180 | S | 0 | 0 | 15 | 1 | 44 |
| 181 | Z | 0 | 0 | 15 | 1 | 42 |
| 182 | 2 | 0 | 0 | 15 | 1 | 46 |
| 183 | 5 | 0 | 0 | 15 | 1 | 47 |

d2.dic (caractères avec points)

| N°= | car | pos_pt | nb_pt | poids | all | taille |
|-----|--------|--------|-------|-------|-----|--------|
| 0 | thai | 1 | 3 | 14 | 3 | 32 |
| 1 | tai | 1 | 2 | 15 | 3 | 18 |
| 2 | thai | 1 | 3 | 15 | 3 | 18 |
| 3 | tai | 1 | 2 | 14 | 3 | 14 |
| 4 | thai | 1 | 3 | 14 | 3 | 15 |
| 5 | nouni | 1 | 1 | 14 | 2 | 61 |
| 6 | taf | 1 | 2 | 15 | 3 | 41 |
| 7 | chini | 1 | 3 | 10 | 3 | 20 |
| 8 | thaf | 1 | 3 | 7 | 3 | 41 |
| 9 | xxx | 0 | 0 | 0 | 0 | 0 |
| 10 | xxx | 0 | 0 | 0 | 0 | 0 |
| 11 | xxx | 0 | 0 | 0 | 0 | 0 |
| 12 | xxx | 0 | 0 | 0 | 0 | 0 |
| 13 | xxx | 0 | 0 | 0 | 0 | 0 |
| 14 | xxx | 0 | 0 | 0 | 0 | 0 |
| 15 | xxx | 0 | 0 | 0 | 0 | 0 |
| 16 | xxx | 0 | 0 | 0 | 0 | 0 |
| 17 | xxx | 0 | 0 | 0 | 0 | 0 |
| 18 | xxx | 0 | 0 | 0 | 0 | 0 |
| 19 | xxx | 0 | 0 | 0 | 0 | 0 |
| 20 | kai | 1 | 2 | 10 | 2 | 51 |
| 21 | fai | 1 | 1 | 15 | 3 | 41 |
| 22 | dadd | 1 | 1 | 14 | 3 | 35 |
| 23 | dadf | 1 | 1 | 10 | 3 | 62 |
| 24 | tami | 1 | 2 | 4 | 2 | 37 |
| 25 | tamf | 1 | 2 | 4 | 2 | 57 |
| 26 | xxx | 0 | 0 | 0 | 0 | 0 |
| 27 | xxx | 0 | 0 | 0 | 0 | 0 |
| 28 | xxx | 0 | 0 | 0 | 0 | 0 |
| 29 | xxx | 0 | 0 | 0 | 0 | 0 |
| 30 | xxx | 0 | 0 | 0 | 0 | 0 |
| 31 | xxx | 0 | 0 | 0 | 0 | 0 |
| 32 | xxx | 0 | 0 | 0 | 0 | 0 |
| 33 | xxx | 0 | 0 | 0 | 0 | 0 |
| 34 | xxx | 0 | 0 | 0 | 0 | 0 |
| 35 | xxx | 0 | 0 | 0 | 0 | 0 |
| 36 | xxx | 0 | 0 | 0 | 0 | 0 |
| 37 | xxx | 0 | 0 | 0 | 0 | 0 |
| 38 | xxx | 0 | 0 | 0 | 0 | 0 |
| 39 | xxx | 0 | 0 | 0 | 0 | 0 |
| 40 | djimi | 2 | 1 | 15 | 1 | 43 |
| 41 | khai | 1 | 1 | 15 | 1 | 43 |
| 42 | ghaini | 1 | 1 | 6 | 1 | 50 |
| 43 | djimi | 2 | 1 | 15 | 1 | 32 |
| 44 | khai | 1 | 1 | 15 | 1 | 31 |
| 45 | ghaini | 1 | 1 | 7 | 1 | 37 |
| 46 | yai | 3 | 2 | 10 | 3 | 64 |
| 47 | yaf | 3 | 2 | 11 | 3 | 45 |
| 48 | dhadi | 2 | 1 | 6 | 3 | 28 |
| 49 | baf | 3 | 1 | 7 | 3 | 39 |
| 50 | xxx | 0 | 0 | 0 | 0 | 0 |
| 51 | xxx | 0 | 0 | 0 | 0 | 0 |

| | | | | | | |
|-----|--------|---|---|----|---|----|
| 52 | xxx | 0 | 0 | 0 | 0 | 0 |
| 53 | xxx | 0 | 0 | 0 | 0 | 0 |
| 54 | xxx | 0 | 0 | 0 | 0 | 0 |
| 55 | xxx | 0 | 0 | 0 | 0 | 0 |
| 56 | xxx | 0 | 0 | 0 | 0 | 0 |
| 57 | xxx | 0 | 0 | 0 | 0 | 0 |
| 58 | xxx | 0 | 0 | 0 | 0 | 0 |
| 59 | xxx | 0 | 0 | 0 | 0 | 0 |
| 60 | dhadi | 2 | 1 | 6 | 2 | 57 |
| 61 | dhadi | 2 | 1 | 6 | 3 | 28 |
| 62 | xxx | 0 | 0 | 0 | 0 | 0 |
| 63 | xxx | 0 | 0 | 0 | 0 | 0 |
| 64 | xxx | 0 | 0 | 0 | 0 | 0 |
| 65 | xxx | 0 | 0 | 0 | 0 | 0 |
| 66 | xxx | 0 | 0 | 0 | 0 | 0 |
| 67 | xxx | 0 | 0 | 0 | 0 | 0 |
| 68 | xxx | 0 | 0 | 0 | 0 | 0 |
| 69 | xxx | 0 | 0 | 0 | 0 | 0 |
| 70 | xxx | 0 | 0 | 0 | 0 | 0 |
| 71 | xxx | 0 | 0 | 0 | 0 | 0 |
| 72 | xxx | 0 | 0 | 0 | 0 | 0 |
| 73 | xxx | 0 | 0 | 0 | 0 | 0 |
| 74 | xxx | 0 | 0 | 0 | 0 | 0 |
| 75 | xxx | 0 | 0 | 0 | 0 | 0 |
| 76 | xxx | 0 | 0 | 0 | 0 | 0 |
| 77 | xxx | 0 | 0 | 0 | 0 | 0 |
| 78 | xxx | 0 | 0 | 0 | 0 | 0 |
| 79 | xxx | 0 | 0 | 0 | 0 | 0 |
| 80 | ghaini | 1 | 1 | 7 | 1 | 97 |
| 81 | ghaini | 1 | 1 | 6 | 1 | 74 |
| 82 | xxx | 0 | 0 | 0 | 0 | 0 |
| 83 | xxx | 0 | 0 | 0 | 0 | 0 |
| 84 | xxx | 0 | 0 | 0 | 0 | 0 |
| 85 | xxx | 0 | 0 | 0 | 0 | 0 |
| 86 | xxx | 0 | 0 | 0 | 0 | 0 |
| 87 | xxx | 0 | 0 | 0 | 0 | 0 |
| 88 | xxx | 0 | 0 | 0 | 0 | 0 |
| 89 | xxx | 0 | 0 | 0 | 0 | 0 |
| 90 | xxx | 0 | 0 | 0 | 0 | 0 |
| 91 | xxx | 0 | 0 | 0 | 0 | 0 |
| 92 | xxx | 0 | 0 | 0 | 0 | 0 |
| 93 | xxx | 0 | 0 | 0 | 0 | 0 |
| 94 | xxx | 0 | 0 | 0 | 0 | 0 |
| 95 | xxx | 0 | 0 | 0 | 0 | 0 |
| 96 | xxx | 0 | 0 | 0 | 0 | 0 |
| 97 | xxx | 0 | 0 | 0 | 0 | 0 |
| 98 | xxx | 0 | 0 | 0 | 0 | 0 |
| 99 | xxx | 0 | 0 | 0 | 0 | 0 |
| 100 | dhali | 1 | 1 | 6 | 2 | 39 |
| 101 | zini | 1 | 1 | 10 | 2 | 34 |
| 102 | dhali | 1 | 1 | 6 | 2 | 21 |
| 103 | djimd | 3 | 1 | 7 | 3 | 36 |
| 104 | khad | 1 | 1 | 7 | 3 | 37 |
| 105 | yad | 3 | 2 | 14 | 2 | 40 |
| 106 | bam | 3 | 1 | 6 | 2 | 40 |
| 107 | tam | 1 | 2 | 14 | 2 | 40 |

| | | | | | | |
|-----|---------|---|---|----|---|----|
| 220 | xxx | 0 | 0 | 0 | 0 | 0 |
| 221 | xxx | 0 | 0 | 0 | 0 | 0 |
| 222 | xxx | 0 | 0 | 0 | 0 | 0 |
| 223 | xxx | 0 | 0 | 0 | 0 | 0 |
| 224 | xxx | 0 | 0 | 0 | 0 | 0 |
| 225 | xxx | 0 | 0 | 0 | 0 | 0 |
| 226 | xxx | 0 | 0 | 0 | 0 | 0 |
| 227 | xxx | 0 | 0 | 0 | 0 | 0 |
| 228 | xxx | 0 | 0 | 0 | 0 | 0 |
| 229 | xxx | 0 | 0 | 0 | 0 | 0 |
| 230 | yai | 3 | 2 | 10 | 3 | 53 |
| 231 | ya_hamf | 1 | 2 | 11 | 3 | 43 |
| 232 | xxx | 0 | 0 | 0 | 0 | 0 |
| 233 | xxx | 0 | 0 | 0 | 0 | 0 |
| 234 | xxx | 0 | 0 | 0 | 0 | 0 |
| 235 | xxx | 0 | 0 | 0 | 0 | 0 |
| 236 | xxx | 0 | 0 | 0 | 0 | 0 |
| 237 | xxx | 0 | 0 | 0 | 0 | 0 |
| 238 | xxx | 0 | 0 | 0 | 0 | 0 |
| 239 | xxx | 0 | 0 | 0 | 0 | 0 |
| 240 | xxx | 0 | 0 | 0 | 0 | 0 |
| 241 | xxx | 0 | 0 | 0 | 0 | 0 |
| 242 | xxx | 0 | 0 | 0 | 0 | 0 |
| 243 | xxx | 0 | 0 | 0 | 0 | 0 |
| 244 | xxx | 0 | 0 | 0 | 0 | 0 |
| 245 | xxx | 0 | 0 | 0 | 0 | 0 |
| 246 | xxx | 0 | 0 | 0 | 0 | 0 |
| 247 | xxx | 0 | 0 | 0 | 0 | 0 |
| 248 | xxx | 0 | 0 | 0 | 0 | 0 |
| 249 | xxx | 0 | 0 | 0 | 0 | 0 |
| 250 | dhali | 1 | 1 | 6 | 1 | 52 |
| 251 | zini | 1 | 1 | 10 | 2 | 66 |
| 252 | bad | 3 | 1 | 14 | 3 | 40 |
| 253 | tad | 1 | 2 | 14 | 2 | 40 |
| 254 | thad | 1 | 3 | 14 | 2 | 39 |
| 255 | nound | 1 | 1 | 14 | 2 | 40 |
| 256 | djimm | 3 | 1 | 6 | 3 | 37 |
| 257 | dhalm | 1 | 1 | 6 | 2 | 53 |
| 258 | zinm | 1 | 1 | 2 | 3 | 65 |
| 259 | nounm | 1 | 1 | 6 | 2 | 39 |
| 260 | zini | 1 | 1 | 10 | 2 | 47 |
| 261 | nouni | 1 | 1 | 14 | 2 | 45 |
| 262 | dhali | 1 | 1 | 6 | 2 | 27 |
| 263 | zini | 1 | 1 | 10 | 2 | 24 |
| 264 | kham | 1 | 1 | 7 | 3 | 35 |
| 265 | lam_all | 1 | 2 | 9 | 1 | 81 |
| 266 | lam_al2 | 3 | 1 | 9 | 1 | 83 |
| 267 | xxx | 0 | 0 | 0 | 0 | 0 |
| 268 | xxx | 0 | 0 | 0 | 0 | 0 |
| 269 | xxx | 0 | 0 | 0 | 0 | 0 |
| 270 | dadm | 1 | 1 | 14 | 3 | 35 |
| 271 | fam | 1 | 1 | 6 | 2 | 44 |
| 272 | xxx | 0 | 0 | 0 | 0 | 0 |
| 273 | xxx | 0 | 0 | 0 | 0 | 0 |
| 274 | xxx | 0 | 0 | 0 | 0 | 0 |
| 275 | xxx | 0 | 0 | 0 | 0 | 0 |