



HAL
open science

**Procédé de reconnaissance syntaxique des caractères
alphanumériques manuscrits : réalisation
micro-informatique**

Alain Kinder

► **To cite this version:**

Alain Kinder. Procédé de reconnaissance syntaxique des caractères alphanumériques manuscrits : réalisation micro-informatique. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1981. Français. NNT: . tel-00298254

HAL Id: tel-00298254

<https://theses.hal.science/tel-00298254>

Submitted on 16 Jul 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

T H È S E

présentée à

l'INSTITUT NATIONAL POLYTECHNIQUE de GRENOBLE

pour obtenir le

TITRE de DOCTEUR - INGENIEUR

par

ALAIN KINDER

INSTITUT INAG
INSTITUT NATIONAL POLYTECHNIQUE
CNR 9 - INP 9 - 100 M 9
MEDIATHEQUE
B.P. 53 X
38041 GRENOBLE CEDEX
FRANCE
Tel. 78.31.48.33

★

PROCÉDÉ DE RECONNAISSANCE SYNTAXIQUE
DES
CARACTÈRES ALPHANUMÉRIQUES MANUSCRITS

RÉALISATION MICRO - INFORMATIQUE

★

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

ANNEE UNIVERSITAIRE

— 1979 - 1980 —

P R E S I D E N T : M. Philippe TRAYNARD

V I C E - P R E S I D E N T S : M. Georges LESPINARD
M. René PAUTHENET

PROFESSEURS DES UNIVERSITES

MM.	ANCEAU François	Informatique fondamentale et appliquée
	BENOIT Jean	Radioélectricité
	BESSON Jean	Chimie minérale
	BLIMAN Samuel	Electronique
	BLOCH Daniel	Physique du solide, Cristallographie
	BOIS Philippe	Mécanique
	BONNETAIN Lucien	Génie chimique
	BONNIER Etienne	Métallurgie
	BOUVARD Maurice	Génie mécanique
	BRISSONNEAU Pierre	Physique des matériaux
	BUYLE-BODIN Maurice	Electronique
	CHARTIER Germain	Electronique
	CHERADAME Hervé	Chimie physique macromoléculaire
Mme	CHERUY Arlette	Automatique
MM.	CHIAVERINA Jean	Biologie, Biochimie, Agronomie
	COHEN Joseph	Electronique
	COUMES André	Electronique
	DURAND Francis	Métallurgie
	DURAND Jean-Louis	Physique nucléaire et corpusculaire
	FELICI Noël	Electrotechnique
	FOULARD Claude	Automatique
	GUYOT Pierre	Métallurgie physique
	IVANES Marcel	Electrotechnique
	JOUBERT Jean-Claude	Physique du solide, Cristallographie
	LACOUME Jean-Louis	Géophysique, Traitement du signal
	LANCIA Roland	Electronique, Automatique
	LESIEUR Marcel	Mécanique
	LESPINARD Georges	Mécanique
	LONGQUEUE Jean-Pierre	Physique nucléaire corpusculaire
	MOREAU René	Mécanique
	MORET Roger	Physique nucléaire corpusculaire
	PARIAUD Jean-Charles	Chimie physique
	PAUTHENET René	Physique du solide, Cristallographie
	PERRET René	Automatique
	PERRET Robert	Electrotechnique
	PIAU Jean-Michel	Mécanique
	PIERRARD Jean-Marie	Mécanique
	POLOUJADOFF Michel	Electrotechnique
	POUPOT Christian	Electronique, Automatique
	RAMEAU Jean-Jacques	
	ROBERT André	Chimie appliquée et des matériaux
	ROBERT François	Analyse numérique
	SABONNADIÈRE Jean-Claude	Electrotechnique
Mme	SAUCIER Gabrielle	Informatique fondamentale et appliquée
M.	SOHM Jean-Claude	Chimie physique
Mme	SCHLENKER Claire	Physique du solide, Cristallographie
MM.	TRAYNARD Philippe	Chimie physique
	VEILLON Gérard	Informatique fondamentale et appliquée
	ZADWORYN François	Electronique

CHERCHEURS DU CNRS

Directeur de Recherche :

M. FRUCHART Robert

Maîtres de Recherche :

MM. ANSARA Ibrahim
BRONOEL Guy
CARRÉ René
DAVID René
DRIOLE Jean

MM. KAMARINOS Georges
KLEITZ Michel
LANDAU Ioan-Doré
Mermet Jean
MUNIER Jacques

PERSONNALITES HABILITEES A DIRIGER DES TRAVAUX DE RECHERCHE
(décision du Conseil Scientifique)

E.N.S.E.E.G. :

MM. ALLIBERT Michel
BERNARD Claude
CAILLET Marcel
Mme CHATILLON Catherine
MM. COULON Michel
HAMMOU Abdelkader
JOUD Jean-Charles

MM. RAVAINÉ Denis
SAINFORT
SARRAZIN Pierre
SOUQUET Jean-Louis
TOUZAIN Philippe
URBAIN Georges

E.N.S.M.S.E. :

MM. BISCONDI Michel
BOOS Jean-Yves
DRIVER Julian
GUILHOT Bernard
KOBILANSKI André
LALAUZE René
LANCELOT Francis

MM. LE COZE Jean
LESBATS Pierre
RIEU Jean
SOUSTELLE Michel
THEVENOT François
THOMAS Gérard
TRAN MINH Canh

E.N.S.E.R.G. :

MM. BOREL Joseph
CHEHIKIAN Alain

M. VIKTOROVITCH Pierre

E.N.S.I.E.G. :

MM. BORNARD Guy
DESCHIZEAUX Pierre
GLANGEAUD François
JAUSSAUD Pierre

Mme JOURDAIN Geneviève
MM. LEJEUNE Gérard
PERARD Jacques

E.N.S.H.G. :

M. DELHAYE Jean-Marc

E.N.S.I.M.A.G. :

MM. COURTIN Jacques
LATOMBE Jean-Claude

MM. LUCAS Michel
VERDILLON André

☆ ☆

★

SOMMAIRE

Pages :

1. INTRODUCTION	1
1.1. Reconnaissance statistique	2
1.2. Reconnaissance structurelle	5
1.3. Notre proposition	7
2. ACQUISITION D'UN CARACTERE	13
2.1. Analyse du texte	14
2.2. Digitalisation et prétraitement	15
3. LE TRAITEMENT	19
3.1. Génération de chaînes de caractères	20
3.2. Simplification de la chaîne de caractères	25
3.3. Normalisation des chaînes de caractères	30
3.4. Essai de l'organe de traitement	33
3.5. Détermination du seuil d'acquisition optimal	38
3.6. Proposition d'une recherche rapide du seuil d'acquisition	41
4. LE CLASSIFIEUR	47
4.1. Description induite	49
4.1.1. Description de la structure globale du caractère	49
4.1.2. Fonctions descriptives	52
4.2. Stratégie d'apprentissage et résultats	55
4.2.1. Le dictionnaire	57
4.2.2. Recherche de g_i' et l_i''	61
4.2.3. Test du classifieur	68
4.3. Second niveau du classifieur	77
5. CONCLUSION	87
BIBLIOGRAPHIE	91

Ce travail a été effectué à l'Ecole Nationale Supérieure d'Electronique et de Radioélectrique de Grenoble, grâce à l'accueil bienveillant de son Directeur, Monsieur le Professeur M. BUYLE-BODIN qui me fait l'honneur d'accepter la présidence de ce jury, je le prie de bien vouloir trouver ici ma profonde reconnaissance.

Je remercie Monsieur le Professeur L. BOLLIET et Monsieur I.D. LANDAU, Maître de recherche, d'avoir accepté d'examiner mes travaux et de participer au jury. Je leur sais gré de cette marque d'intérêt.

Qu'il me soit permis de remercier de façon toute particulière Monsieur le Professeur A. CHEHIKIAN, Directeur du laboratoire du Signal Bioélectrique et de la Reconnaissance des Formes de l'E.N.S.E.R.G, qui m'a accueilli dans son laboratoire. Ses conseils éclairés et son aide efficace m'ont permis de structurer et de mener à bien ce travail.

Monsieur Gérard BOUVIER a suivi mes activités de recherche avec beaucoup d'attention. Ses travaux précédents et ses conseils pertinents sont à l'origine de cette thèse. Qu'il trouve ici le témoignage de ma profonde gratitude.

Que tout le personnel du Service Informatique et en particulier son chef, Monsieur P. EVEN soit remercié pour leur aide.

Je n'oublierai pas de remercier tous mes collègues du laboratoire, le personnel technique et administratif de l'Ecole qui m'ont permis d'effectuer ce travail de recherche dans une excellente ambiance. Je remercie en particulier Monsieur M. CREPIN-JOURDAN, Mademoiselle O. MERKLEN pour l'important travail qu'elle a fourni lors de la mise en page de cette thèse et ainsi que Messieurs CARECCHIO et DANIEL pour l'excellente reprographie de ce document.

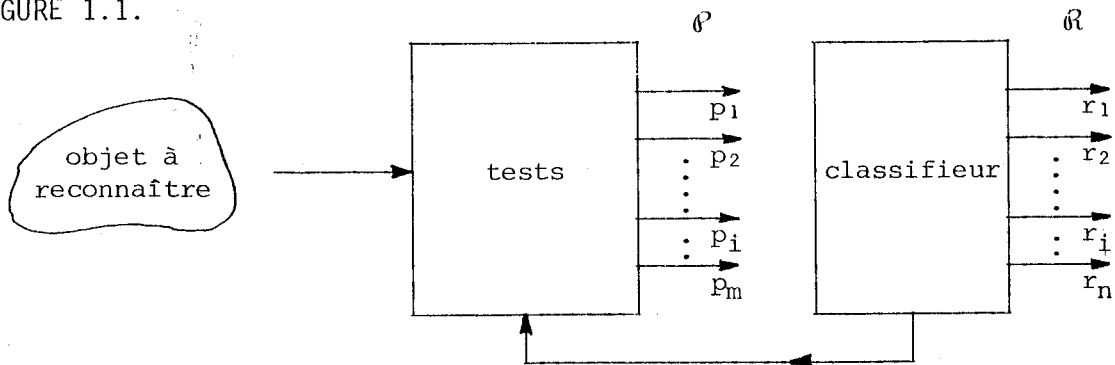
Enfin, qu'il me soit permis de remercier Sylvaine, ma famille et mes amis qui m'ont encouragé et aidé pendant ce travail de recherche.

I - INTRODUCTION

Le but de notre travail consiste à concevoir et réaliser un appareil de reconnaissance de caractères alphanumériques latins dactylographiés et manuscrits.

La structure générale d'un tel appareil sera la suivante [1,2,3,4,5,6,7]:

FIGURE 1.1.



- les tests : on effectue sur l'objet à reconnaître, qui se présente en général sous la forme d'une matrice binaire (les points blancs seront codés "0" et les noirs par des "1", des tests qui ont pour but d'extraire, de l'image bidimensionnelle, un ensemble \mathcal{P} de propriétés p_i adaptées au type du problème posé par la reconnaissance.
- le classifieur exploite les propriétés p_i afin de déterminer la classe d'appartenance de l'objet et donne ainsi une réponse $r_i \in \mathcal{R}$. En reconnaissance de caractères alphanumériques, l'objet à reconnaître est constitué par le caractère échantillonné et digitalisé. Une réponse r_i représente une classe parmi les 36 possibles (26 lettres + 10 chiffres).

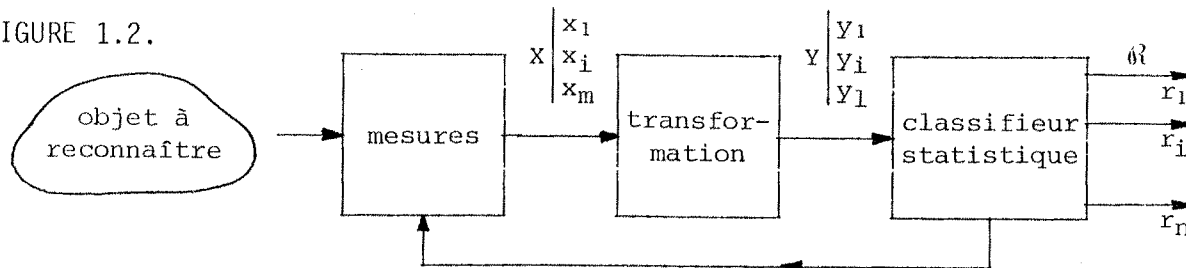
En cas d'indécision, réponse multiple ($r_i \cup r_j$), le classifieur a la possibilité de demander de nouveaux tests, spécifiques à l'ambiguïté à lever. En cas de rejet, aucune réponse ne correspond aux propriétés p_i ; le classifieur demandera une nouvelle acquisition de l'objet.

Suivant le type des tests effectués et le classifieur utilisé, on distingue classiquement deux procédés de reconnaissance : reconnaissance statistique et reconnaissance structurale [8].

1.1. RECONNAISSANCE STATISTIQUE

Dans ce cas, l'ensemble des tests est constitué par des mesures effectuées sur l'objet. Les propriétés p_i sont alors représentées par un vecteur de mesure X défini dans un espace à n dimensions. On peut représenter le système de reconnaissance de la manière suivante :

FIGURE 1.2.



Les mesures effectuées sur l'objet, ici le caractère alphanumérique, sont réalisées dans une métrique supposée adaptée au problème. On sera amené à faire subir au vecteur X une transformation en vue d'obtenir une meilleure séparation des classes dans l'espace de mesure. Le caractère est alors représenté par un vecteur Y souvent moins redondant que le vecteur X et dont la qualité conditionnera les résultats du classifieur [9,10,11].

Supposons que le vecteur Y ne comprenne que deux composantes et soit représentable dans le plan \mathbb{R}^2 (figure 1.3, page 3). Le classifieur statistique, faisant appel à la théorie de la décision bayésienne, a comme but, dans une première phase d'apprentissage, la recherche d'une droite D satisfaisant à la relation du type : "Tous les points appris du caractère n° 1 se trouvent d'un côté de la droite D , tous ceux du caractère n° 2 se trouvent de l'autre côté". L'ensemble des points regroupés dans une même région du plan constitue une classe. Les mesures et les transformations sont d'autant meilleures que les points appartenant à une même classe sont regroupés (distance intraclasse faible) et que les classes ainsi formées sont éloignées les unes des autres (distance interclasse grande).

FIGURE 1.3. Apprentissage.

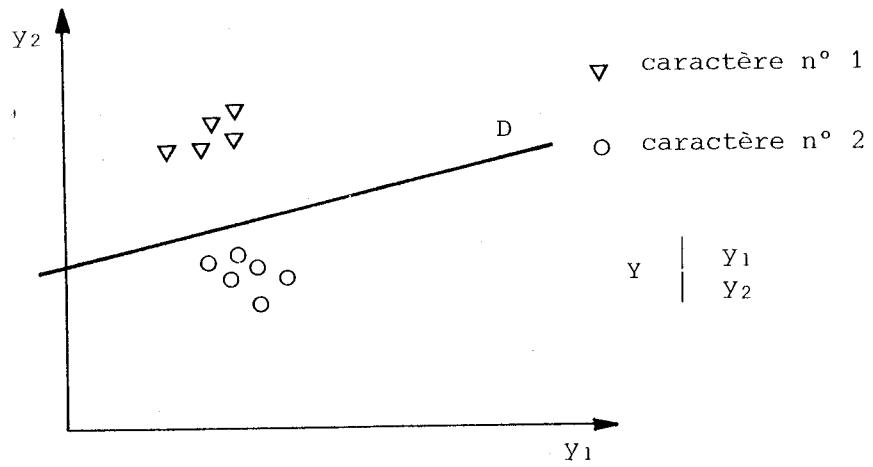
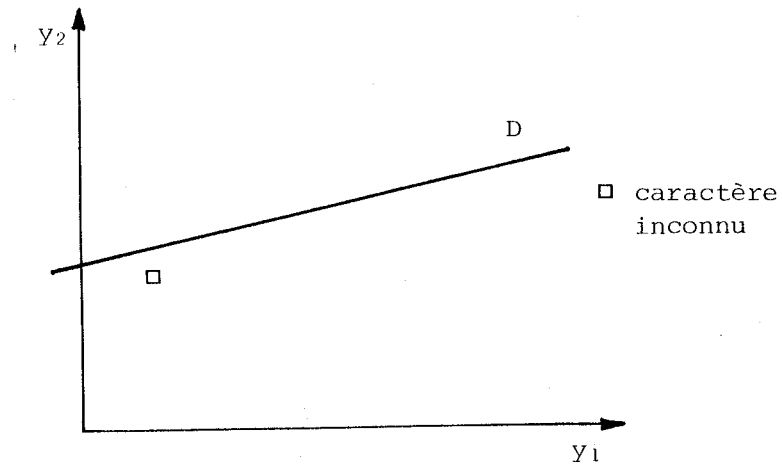


FIGURE 1.4. Reconnaissance.



Pendant la seconde phase, un caractère inconnu (figure 1.4) va être présenté au classifieur sous forme d'un vecteur Y ; l'organe de décision qui s'appuie sur les résultats de l'apprentissage va classer ce caractère : "le caractère inconnu est situé 'sous' la droite D ; il convient de le reconnaître comme appartenant à la classe du caractère n° 2".

La reconnaissance statistique est adaptée si l'on peut répondre aux trois questions :

☆ Les échantillons d'apprentissage sont-ils significatifs de l'ensemble étudié ?

En général, cette question n'a pas de réponse, sauf si des données théoriques permettent de connaître les lois de distribution des éléments.

☆ Les classes sont-elles toujours linéairement séparables ?

En reconnaissance multiforme, on peut considérer qu'elles ne le sont pas (on peut visualiser ce problème dans le plan \mathbb{R}^2 ; figure 1.5). A défaut, il est possible de rendre ces classes linéairement séparables par parties, en créant des sous-classes a priori inconnues, fonctions de la métrique utilisée, des tests et des transformations (figure 1.6). Ces sous-classes sont déterminées par apprentissage non supervisé.

FIGURE 1.5. Classes non linéairement séparables.

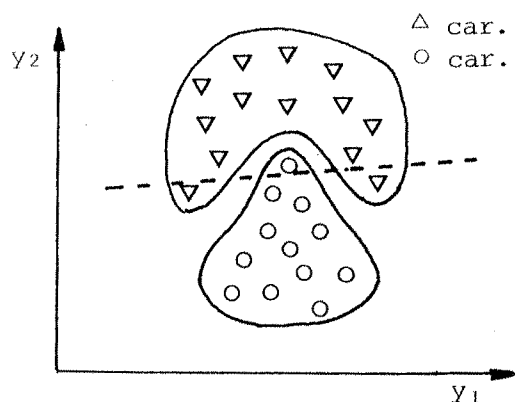
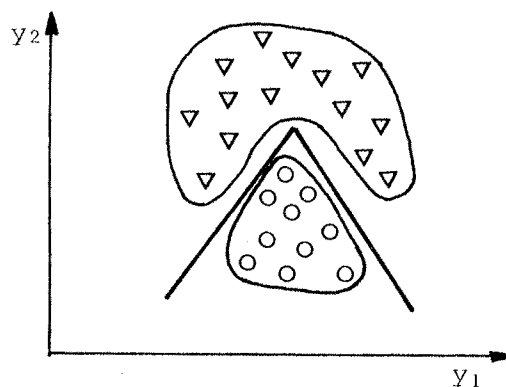


FIGURE 1.6.



☆ L'apprentissage non supervisé est trivial dans \mathbb{R}^2 . Malheureusement, le vecteur Y n'est pas défini dans un espace à deux dimensions, mais dans \mathbb{R}^l avec $l \gg 2$; qu'en est-il de l'apprentissage ?

Les calculs se compliquent singulièrement et toute représentation est impossible, d'autant qu'il devient inextricable de séparer 36 classes non linéairement séparables dans l'espace \mathbb{R}^l .

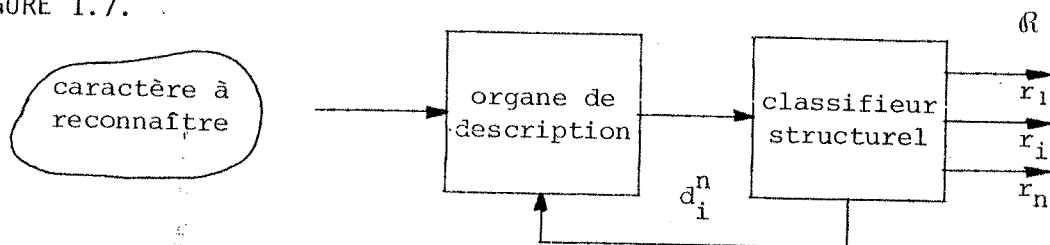
Nous pouvons conclure que ce procédé est parfaitement applicable à la lecture de caractères monofontes élaborés pour ce type de reconnaissance (E13B, OCRA, OCRB). Néanmoins, la reconnaissance statistique a toujours été un échec dans la lecture des caractères multiformes et, a fortiori, manuscrits qui, subissant de fortes distorsions, ne voient pas leurs propriétés métriques conservées.

1.2. RECONNAISSANCE STRUCTURELLE

Dans la reconnaissance structurelle, on fait appel, d'une part, à la théorie des graphes et le calcul des prédicats (on parle alors de méthode de reconnaissance descriptive) et, d'autre part, à la théorie des langages : méthodes syntaxiques. Nous ne développerons que ces dernières, car elles sont adaptées à la reconnaissance de caractères multifontes [8,12,13,14,15].

Dans un tel système (figure 1.7), l'ensemble de tests est constitué par un organe de description qui produit une ou plusieurs phrases de description.

FIGURE 1.7.



Les propriétés métriques sont abandonnées au profit des propriétés de structures, lesquelles sont dans une grande part insensibles aux distortions. Ce procédé convient donc à la reconnaissance des caractères manuscrits. GAILLAT a classé les méthodes d'extraction des traits caractéristiques selon la notion physique qui est mise en évidence [16]. Nous allons utiliser une méthode structurelle qui fait intervenir une synthèse des analyses sous forme d'une description globale du caractère [17,18,19,20].

Une phrase d_i^n décrit le graphisme à l'aide de primitives assemblées selon des règles établies. Les primitives peuvent être significatives de segments de droite, d'arcs de cercles, de boucles fermées, de vecteurs représentant une partie du caractère.

Comme en reconnaissance statistique, il faut traiter deux points dans le classifieur : l'apprentissage et la décision.

Le premier consiste à essayer de découvrir la structure commune d'une classe C_i , contenant un nombre n fini de caractères A_i^n décomposés en phrases d_i^n . Une classe C_i regroupera des caractères ayant des propriétés communes au sens du type de la description retenue, ce qui se traduit, en principe, par un regroupement des caractères de même nom ou des doublets classiques (8 et B par exemple).

Soit D_i l'ensemble des n phrases de description, rencontrées lors de l'apprentissage et relatives à la classe C_i . Il s'agit de découvrir le langage L_i propre à la classe C_i . Ce langage devra contenir l'ensemble D_i rencontré pendant l'apprentissage tel que : $D_i \subset L_i$ et $D_j \not\subset L_i$ ($i \neq j$).

Le langage L_i sera obtenu par l'intermédiaire de la grammaire G_i qui le génère, grammaire inférée à partir de D_i . Cependant l'inférence grammaticale, au sens strict, n'est pas toujours possible et demeure limitée à des grammaires de structure particulière, comme les grammaires régulières et les grammaires à structure contextuelle [21,22]. A défaut, on infèrera une grammaire g_i générant le langage $\ell_i = L(g_i)$ tel que : $D_i \subset \ell_i \subset L_i$ et $D_j \not\subset \ell_i$ ($i \neq j$),

ce qui implique que certains éléments de C_i ne pourront être reconnus, que les éléments de C_j appartenant à l'échantillon d'apprentissage ne seront pas faussement reconnus, mais que certains éléments de C_j , non rencontrés pendant l'apprentissage, pourront être faussement attribués à C_i .

Le second point se définit ainsi : un caractère inconnu X est décrit par d_x ; parmi tous les langages, il faut trouver i tel que $d_x \in L(g_i)$.

On dispose alors de deux types de reconnaissance :

- ★ la première, de type lexical, consiste à rechercher si d_x peut s'identifier à une des descriptions de ℓ_i . Cette méthode, de mise en oeuvre simple, ne peut convenir que si les langages ℓ_i sont peu étendus, car il faut stocker toutes les descriptions générées par g_i .
- ★ la seconde fait appel à l'analyse syntaxique. Dans un premier cas, on examinera si la description d_x peut être générée par la grammaire g_i ; on dit que l'analyse est haut-bas (top-down). Si la réponse est affirmative, le caractère inconnu appartiendra à C_i ; dans le cas contraire, le système renouvellera son essai avec la grammaire g_{i+1} .

Dans le second cas, on se demandera si d_x possède une structure telle qu'elle soit grammaticalement correcte par rapport à la grammaire g_i ; on effectue alors une analyse syntaxique bas-haut (bottom-up).

Le classifieur, après analyse syntaxique, donne une réponse parmi les quatre possibilités suivantes :

1. $\left. \begin{array}{l} d_x \in \ell_i \\ d_x \notin \ell_j \\ x = i \end{array} \right\} \forall j \quad j \neq i \Rightarrow$ il y a reconnaissance $x = i$
2. $\left. \begin{array}{l} d_x \in \ell_i \\ d_x \in \ell_j \\ x = i \end{array} \right\} j \neq i \Rightarrow$ il y a ambiguïté $x = i$ ou j
3. $d_x \notin \ell_i \quad \forall i \Rightarrow$ il y a rejet
4. $\left. \begin{array}{l} d_x \in \ell_i \\ d_x \notin \ell_j \\ x \neq i \end{array} \right\} \forall j \quad j \neq i \Rightarrow$ il y a fausse reconnaissance ou erreur.

Les trois derniers cas proviennent du fait que $\ell_i \subset L_i$; en effet, les éléments $L_i - \ell_i$ ne sont pas décrits par g_i , mais peuvent l'être par une des grammaires g_j ($i \neq j$). En fonction du mode d'inférence, une grammaire g_j peut même décrire l'ensemble $L_i - D_i$. Notre but pendant l'analyse syntaxique sera de tendre vers la situation ; $L_i - \ell_i \rightarrow \emptyset$ c'est à dire que $g_i \rightarrow G_i$.

Les réponses ont été rangées par ordre de gravité croissante. On espère que les cas 3. et 4. soient minoritaires, que le cas 1. soit plus fréquent que 2., mais on ne peut pas affirmer que le cas 1. soit très supérieur à 2. compte tenu du mode d'inférence et de la description choisie.

1.3. NOTRE PROPOSITION

Une description structurelle est constituée de phrases unidimensionnelles. Un procédé trivial pour passer d'une image bidimensionnelle à une "image" monodimensionnelle consiste à effectuer une projection sur une droite. On conservera l'information contenue dans l'image en procédant, par exemple, à une projection sur un ensemble de droites d'orientation variable (Transformation de Radon). Il est tentant de ne projeter que sur deux axes orthogonaux confondus avec les orientations de la maille d'échantillonnage mais on sait qu'il y aura alors une perte sensible d'information à moins que la fonction d'image $f(x,y)$ soit séparable, c'est à dire si $f(x,y) = f_1(x) \cdot f_2(y)$; dans tous les autres cas, il y aura perte de l'information. On peut cependant minimiser cette perte, tout en ne projetant que sur deux axes orthogonaux à la condition de décomposer au préalable en éléments d'ordre k , selon :

★ élément horizontal d'ordre k :

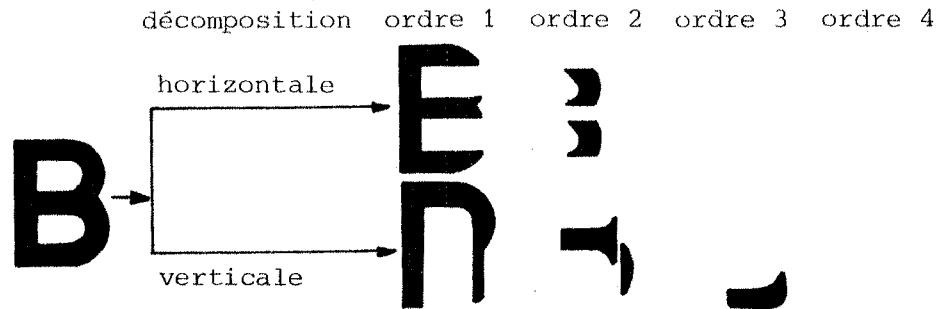
l'ensemble des points, de l'image échantillonnée et digitalisée, connexes selon une direction horizontale sont attribués à un même élément d'ordre k . Les points séparés des précédents par un espace suffisant seront attribués à un élément horizontal d'ordre $k+1$.

★ élément vertical d'ordre k :

la connexité est observée dans une direction verticale.

Cette méthode a été proposée initialement par G. BOUVIER [23]. Dans la suite, nous confondrons, par abus de langage, "élément d'ordre k " avec "ordre k ". A titre d'exemple, un caractère B sera décomposé selon le modèle de la figure 1.8, page 8.

FIGURE 1.8. Décomposition du B en éléments d'ordre k.



Chaque élément horizontal (ou vertical) est ensuite projeté horizontalement (ou verticalement). La figure 1.9 (page 9) explicite la projection horizontale de l'élément horizontal d'ordre 1 du caractère B, tandis que la figure 1.10 (page 9) montre la projection horizontale "brutale" du même caractère B.

Ces figures montrent à l'évidence que la présence des segments horizontaux apparaît clairement dans la projection de l'ordre 1 (figure 1.9) alors que, dans la projection brutale (figure 1.10), ces segments ne sont pas discernables.

De plus, les éléments d'ordre supérieur à 1, traités de façon similaire, apportent un complément d'information sur la structure de l'image ; cette information serait perdue si on ne considérait que, soit la projection d'ordre 1, soit, a fortiori, la projection brutale de tout le caractère.

Cette méthode présente en outre les avantages suivants :

- . sauf accident (taches, ...), les caractères alphanumériques latins possèdent au plus quatre éléments horizontaux ou quatre éléments verticaux (cas de M et W). Généralement, les éléments sont au nombre de 2 ou 3 par sens d'analyse, ce qui limite l'importance de la description.
- . au plan technologique, projeter un élément sur deux axes orthogonaux est une opération éminemment simple à réaliser et nécessite en général peu de temps.

On voit que les projections horizontales (ou verticales) des éléments horizontaux (verticaux) nous permettent de détecter simplement, par comparaison à un seuil, la présence de segments horizontaux (verticaux), d'éléments de liaisons entre de tels segments. En d'autres termes, ces projections nous permettent de donner une description structurelle de l'objet.

FIGURE 1.9. Projection horizontale de l'élément horizontal d'ordre 1.

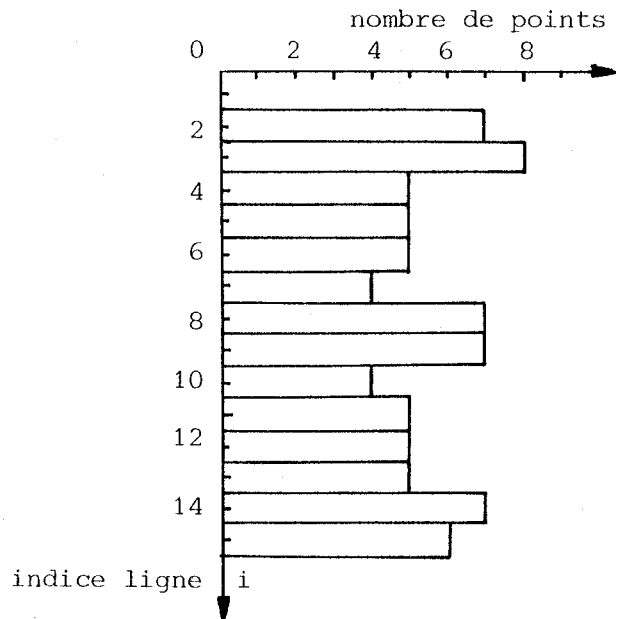
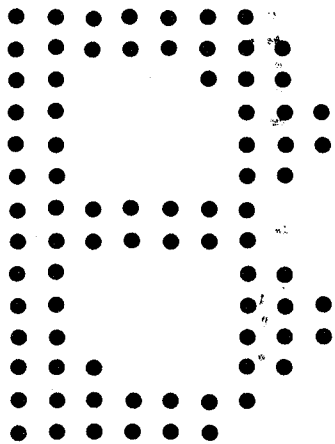
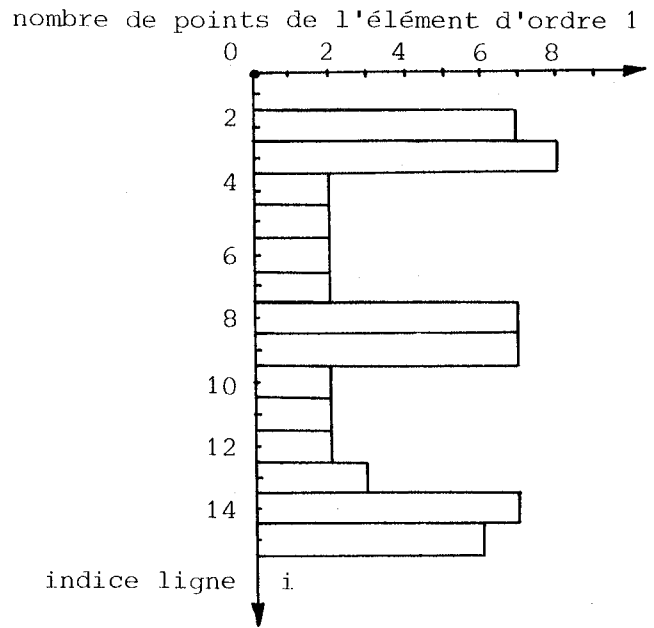
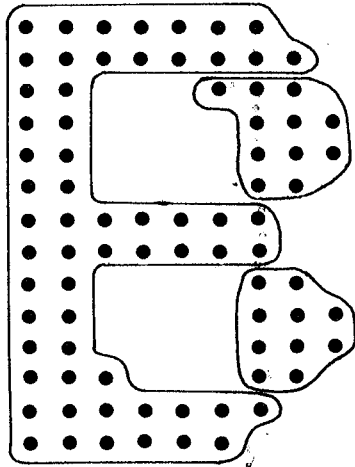


FIGURE 1.10. Projection horizontale de l'élément (sans la notion d'ordre).

Cette description est ensuite simplifiée afin d'éliminer l'effet du bruit et de certaines distorsions ; on opère ensuite une normalisation pour s'affranchir des affinités et réduire l'information que l'on présentera au classifieur [24,25].

En ce qui concerne le classifieur, il procédera par analyse syntaxique bas-haut. Comme nous l'avons vu précédemment, on peut penser que, compte tenu de l'imperfection au stade de l'apprentissage (\mathcal{L}_i au lieu de L_i), les cas d'ambiguïté apparaîtront. De plus, certains caractères alphanumériques possèdent des structures voisines (B,8 ; Z,2 ; D,O,Q ; ...) et ne seront pas discernés dans un premier temps. En effet, vouloir discriminer ces doublets nécessiterait une description très fine, comportant beaucoup d'informations inutiles pour séparer la majorité des autres caractères. Cette masse d'informations alourdirait inutilement la tâche du classifieur. Il est donc préférable de procéder en deux étapes [26].

La première, au cours de laquelle le classifieur détermine la ou les classes d'appartenance du caractère analysé ; en cas de non reconnaissance (ambiguïté), une seconde étape portant sur la reconnaissance de 2, 3, voire 4 caractères voisins, permet de lever l'ambiguïté et de séparer ces caractères en effectuant des tests spécifiques [27].

1.3.1. TECHNOLOGIE RETENUE

Les machines de reconnaissance de caractères furent, dans les années 70, essentiellement de deux types : électronique et informatique.

Les systèmes électroniques favorisent de par leur élaboration les traitements basés sur des propriétés métriques, ce qui impose l'utilisation d'un classifieur statistique ou "bayésien". Ces systèmes construits avec des circuits discrets ont des vitesses de reconnaissance élevées (qq 10 ms), mais se comportant mal en reconnaissance multifontes eu égard aux choix imposés par la technologie. On remarquera également que ces machines ont une configuration figée, peu apte à la recherche.

A l'opposé, on rencontre les systèmes purement informatiques qui introduisent la souplesse de la programmation. Les reconnaissances de type structurel ont été favorisées et, dans ce système, les caractères sont décrits par des phrases facilement traitées par l'informatique. Généralement, ces systèmes sont imposants et ont des temps de reconnaissance élevés.

L'avènement de la micro-informatique a permis d'allier la taille raisonnable des circuits électroniques, à la souplesse des programmes d'ordinateur. Nous avons adopté une configuration multiprocesseurs.

1.3.2. STRUCTURE PROPOSEE

L'organe d'acquisition (figure 1.11, page 12) est composé d'une caméra pilotée par un microprocesseur qui échantillonne, digitalise et lisse le caractère qu'il isole d'une page d'écriture [28].

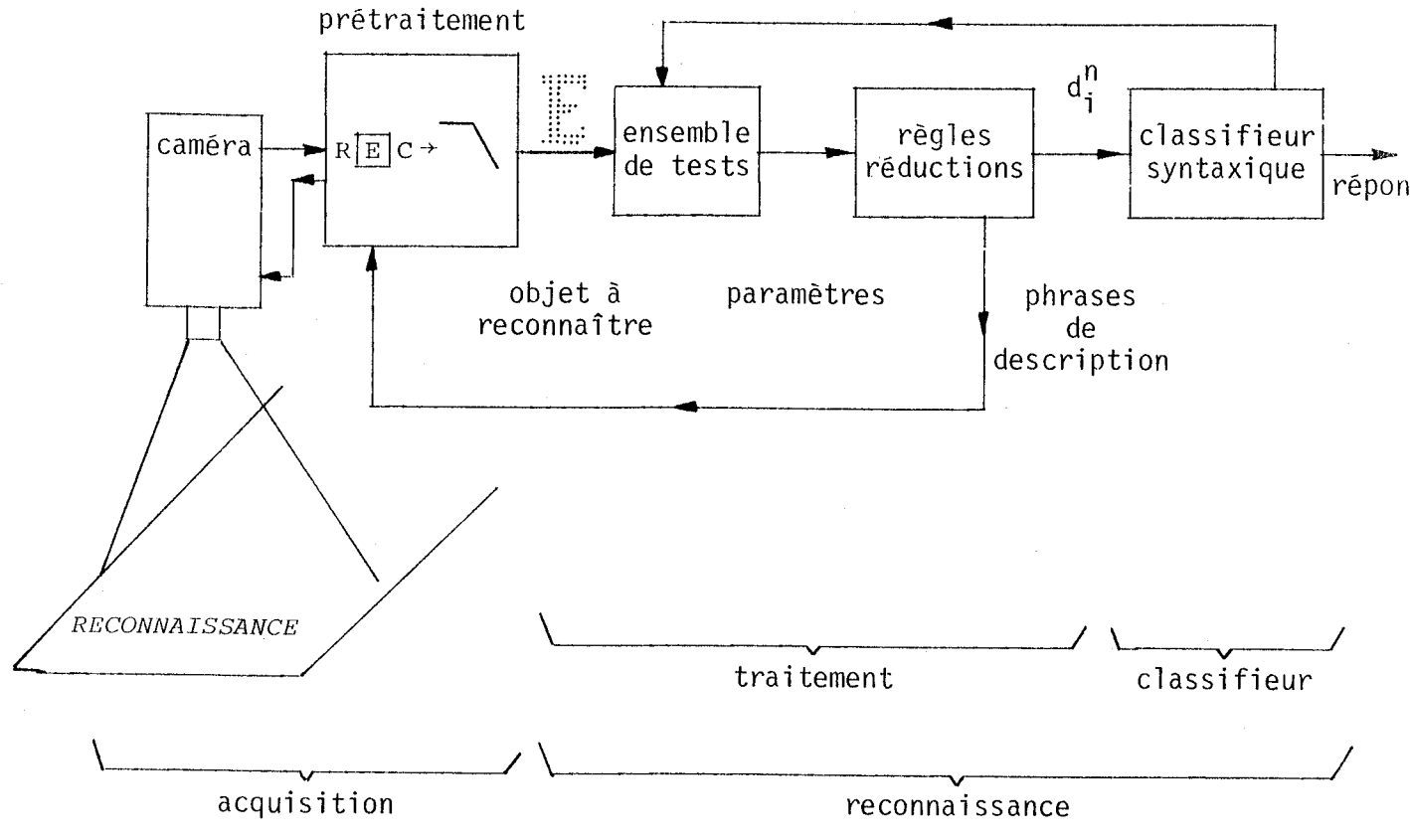
Le traitement élaboré autour d'un autre microprocesseur décompose et projette la matrice binaire issue de l'étage précédent. Dans une seconde étape, l'information est réduite afin d'obtenir une phrase décrivant le caractère. Cette phrase est ensuite analysée par le classifieur qui donne une réponse à l'issue d'une analyse syntaxique.

On remarquera dans cette structure qu'en fonction de la "qualité" de la description obtenue, l'organe de traitement peut effectuer un retour en arrière et demander une nouvelle acquisition, en modifiant les paramètres d'acquisition, afin de trouver une meilleure description. Au niveau du classifieur, une interaction avec l'ensemble de tests est également réalisée afin d'effectuer des tests particuliers aux ambiguïtés constatées (doublets).

En cas de rejet, le classifieur redemandera, par l'intermédiaire de l'ensemble de tests, une nouvelle acquisition et un nouveau traitement.

Chacun de ces éléments sera développé dans la suite.

FIGURE 1.11. Organes d'acquisition et de reconnaissance.



2 - ACQUISITION D'UN CARACTÈRE

Généralement considéré comme un élément trivial, l'organe d'acquisition et de prétraitement introduit des imperfections qui se retrouvent tout au long de la chaîne de traitement. La tâche de cette dernière sera compliquée par des imperfections de deux types :

- bruits d'impression propres au caractère

Un caractère imprimé présente de nombreux défauts, principalement des défauts de contour, mais également des défauts intérieurs. Ce bruit se traduit sur la matrice digitalisée par des points erratiques. On souhaitera que ces défauts soient sinon éliminés, du moins atténués par un prétraitement réalisé au niveau de l'acquisition.

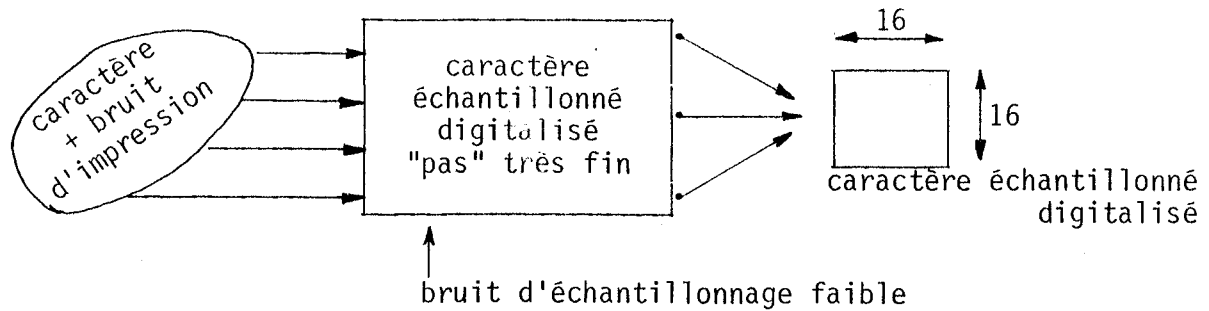
- bruit d'échantillonnage

Il se décompose en bruit de contour et bruit de phase. On notera que si le pas d'échantillonnage est important, la définition est grossière ce qui tend à atténuer le bruit de contour propre au caractère mais ajoute un bruit de contour propre à l'échantillonnage. De plus, la maille d'échantillonnage étant aléatoirement positionnée par rapport au caractère, la structure de celle-ci subira des déformations aléatoires après échantillonnage : bruit de phase.

On conçoit alors l'intérêt d'un prétraitement à qui on demandera de ne pas introduire d'imperfections sur la forme échantillonnée, d'atténuer celles existant sur l'image initiale, tout en limitant le nombre d'échantillons au strict nécessaire pour la reconnaissance [29].

L'expérience nous a montré, et la suite de ce travail le confirmera, qu'un caractère digitalisé dans une matrice 16×16 est reconnaissable. Cependant, le pas d'échantillonnage correspondant est trop grand pour éviter les problèmes que nous venons de citer. Nous procéderons alors à une acquisition selon le schéma de la figure 2.1, page 14.

FIGURE 2.1.



Le caractère est échantillonné avec un pas très petit, ce qui nous permet de diminuer l'influence du bruit d'échantillonnage. L'information, fortement redondante, est ensuite réduite grâce à un algorithme qui s'efforce de minimiser les bruits d'impression et d'échantillonnage présents sur le caractère. Nous avons confié ces tâches d'acquisition à un microprocesseur qui nous a permis d'effectuer un prétraitement (réduction + lissage) dont certains paramètres sont pilotables.

Le processeur spécialisé [28], construit autour du microprocesseur 6800 de MOTOROLA, assure les fonctions suivantes :

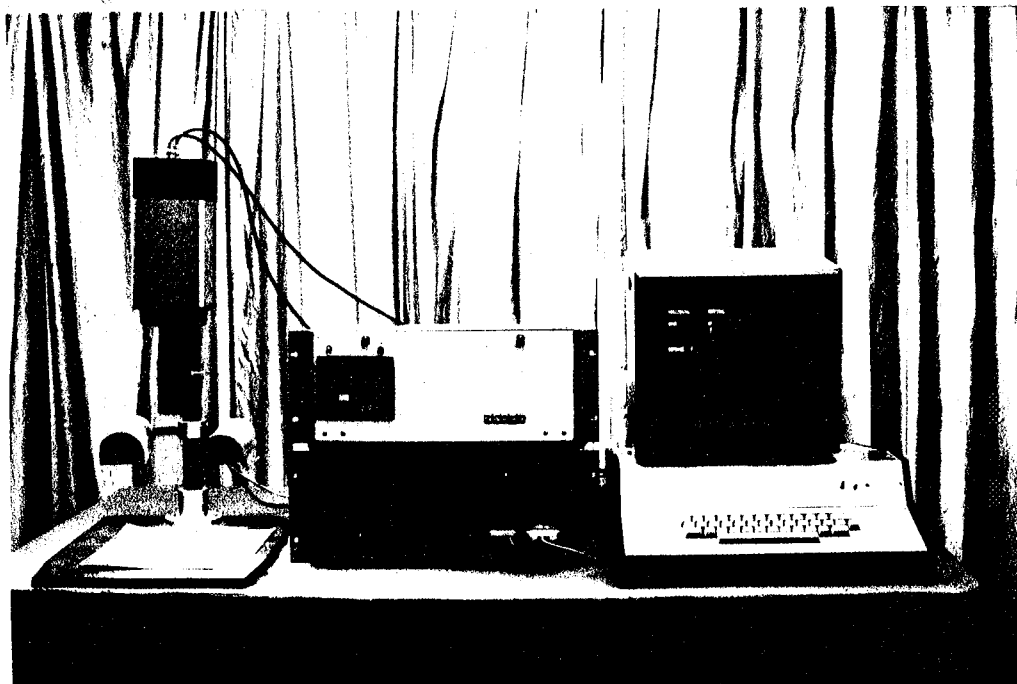
- analyse de texte . suivi de ligne,
 - . retour en début de ligne ou de page,
 - . cadrage des caractères,
- digitalisation d'un caractère,
- prétraitement de ce caractère.

2.1. ANALYSE DU TEXTE

Les caractères, écrits sur papier blanc, sont disposés sous une caméra T.V. standard équipée d'un tube vidicon dont la définition est de 1000 points/ligne et de 875 lignes entrelacées par image. Mémoriser toute une image T.V. nécessiterait une grande capacité mémoire, alors que notre but consiste en la reconnaissance de caractères indépendants les uns des autres.

Il nous est donc apparu plus judicieux de n'acquérir qu'une partie de cette image, matérialisée sur l'écran de contrôle par une fenêtre en surbrillance (figure 2.2, page 15). Le microprocesseur calcule les coordonnées de la fenêtre, la déplace afin d'effectuer le suivi d'une ligne de caractères, un retour ligne ainsi qu'un cadrage automatique des caractères.

FIGURE 2.2.



2.2. DIGITALISATION ET PRETRAITEMENT

Comme nous l'avons annoncé précédemment, nous avons attribué au pas d'échantillonnage une valeur très petite. La définition de la caméra étant de 1000 points/ligne, nous avons décidé d'acquérir 1024 échantillons/ligne. La caméra étant équipée d'un zoom, le pas d'échantillonnage varie de 0,015 mm à 0,12 mm suivant le grossissement. Avec un grossissement moyen, correspondant au cadrage dans la fenêtre de caractères dactylographiés, le pas est environ égal à 0,04 mm. La taille de la fenêtre d'acquisition est de 64 lignes \times 128 points, dans laquelle s'inscrivent en moyenne deux caractères. Pour la reconnaissance des caractères, deux niveaux de quantification sont suffisants : "0" pour le fond blanc et "1" pour le noir des caractères. La fenêtre d'acquisition est représentée par une matrice binaire : $F(i,j)$ avec : $0 \leq i < 64$; $0 \leq j < 128$.

La seconde partie : le prétraitement, consiste à réduire et lisser les deux caractères inscrits dans la matrice $F(i,j)$ de telle sorte que l'organe d'acquisition présente à l'étage suivant (organe de description) un caractère unique inscrit dans une matrice réduite g' de taille 16×16 .

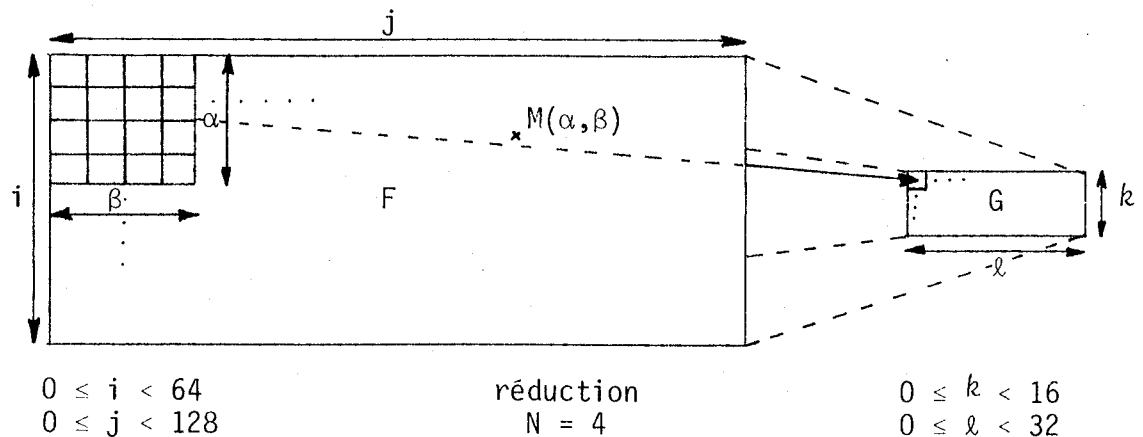
Soit N le facteur de réduction. Un carré de $N \times N$ points pris dans la matrice $F(i,j)$ donnera naissance à un point inscrit dans une nouvelle matrice $G(k,\ell)$ de taille $(64/N) \times (128/N)$ selon l'algorithme :

$$G(k,\ell) = \sum_{\alpha=0}^{N-1} \sum_{\beta=0}^{N-1} F(kN+\alpha, \ell N+\beta) \cdot M(\alpha,\beta)$$

avec : $0 \leq kN < 64$ ($k \in \mathbb{N}$) ;
 $0 \leq \ell N < 128$ ($\ell \in \mathbb{N}$) ;

$M(\alpha,\beta)$: matrice de pondération de taille $N \times N$.

FIGURE 2.3.



La figure ci-dessus illustre un exemple de réduction de facteur $N = 4$. La matrice $G(k,\ell)$ est donc de taille 16×32 ; chaque point de cette matrice représente la somme pondérée des 4×4 points correspondants de $F(i,j)$.

Pour la suite, nous conserverons $N = 4$ et la matrice de pondération $M(\alpha,\beta)$ aura tous ses éléments égaux à 1. La matrice $F(i,j)$ ne prenant que les valeurs 0 et 1, les éléments de $G(k,\ell)$ appartiendront à $[0, N^2]$. $G(k,\ell)$ est ensuite quantifié à deux niveaux 0 et 1 par une opération de seuillage :

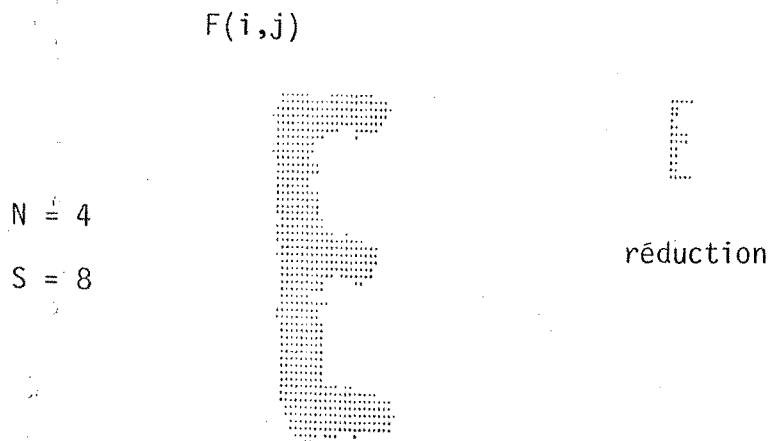
$$\begin{aligned} G_{kl} > S &\Rightarrow g_{kl} = 1 \\ G_{kl} \leq S &\Rightarrow g_{kl} = 0 \end{aligned} \quad S \in [0, N^2-1] .$$

La matrice ainsi obtenue, $g(k,\ell)$, est la matrice réduite contenant deux caractères. On transmet la matrice g' de taille 16×16 contenant un seul caractère isolé.

Le résultat de l'opération de réduction et lissage est illustré sur la figure 2.4, page 17. On observera à gauche un caractère brut inscrit dans la matrice $F(i,j)$, à droite le caractère réduit et lissé dans la matrice g' .

En choisissant un rapport de réduction $N = 4$, un trait de caractère dactylographique dont l'épaisseur moyenne est de l'ordre de 0,3 mm sera représenté, en général, par 2 pixels.

FIGURE 2.4. Acquisition d'un caractère E .



L'organe de tests pilote le seuil S qui a été fixé dans un premier temps à la valeur moyenne de $N^2/2$. L'organe de tests peut demander également : l'acquisition d'un caractère, un prétraitement nouveau, l'acquisition du caractère suivant, le retour de la fenêtre en haut de page.

Le dispositif d'acquisition digitalise deux caractères toutes les 60 ms. Dans ce temps, il faut compter 40 ms nécessaire à un balayage T.V. Si l'on désire augmenter la vitesse d'acquisition, il conviendra d'acquérir un plus grand nombre de caractères lors d'un balayage T.V. La communication entre l'organe d'acquisition et l'ensemble de tests s'effectue suivant un mode de communication série (1200 bauds). La transmission d'un caractère digitalisé prend, dans ces conditions, 300 ms. C'est dans un but de simplicité de mise en oeuvre que nous avons adopté ce type d'échange, qui peut être remplacé à tout moment par une communication de type parallèle qui abaisserait le temps de transmission d'un caractère digitalisé à 1 ms.

3 - LE TRAITEMENT

Le traitement d'un caractère digitalisé a pour but l'obtention d'une description utilisable par le classifieur, ce qui impose les opérations suivantes :

★ Il convient tout d'abord de transformer une image bidimensionnelle en une "image" monodimensionnelle dans laquelle on conservera l'information sur la structure du caractère.

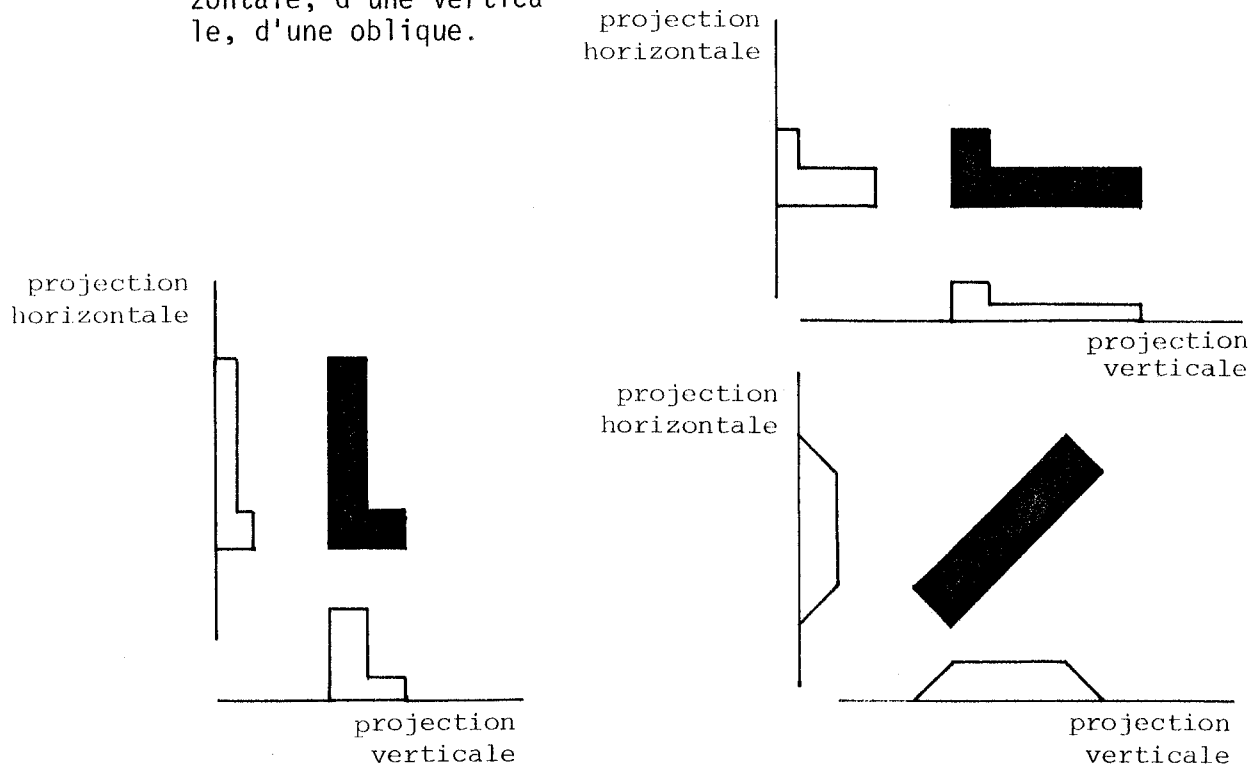
☆ La transformation que nous avons retenue consiste à projeter, sur des axes horizontaux et verticaux, un ensemble d'éléments d'ordre k issu du caractère à reconnaître.

☆ La présence de maximums sur les projections est caractéristique de segments horizontaux s'il s'agit d'une projection horizontale, de segments verticaux s'il s'agit d'une projection verticale. Réciproquement, la présence d'un segment horizontal (respectivement vertical) dans un caractère se traduira donc par la présence d'un maximum de points dans la projection correspondante et d'un "non maximum" dans la projection verticale (respectivement horizontale). Corrélativement, la présence d'un "non maximum" à la fois dans une projection horizontale et verticale traduit la présence d'un segment ni horizontal, ni vertical, c'est à dire : une oblique, un arc de cercle... (figure 3.1, page 20).

☆ Nous considérerons ainsi l'image d'un caractère comme étant l'association de primitives ; ces primitives se composent de segments horizontaux, de segments verticaux, d'éléments ni horizontaux et ni verticaux. De l'image monodimensionnelle, on extrait les informations concernant la structure du caractère, c'est à dire la présence et le mode d'assemblage des primitives.

Cette opération va nous conduire à traduire chacune des projections par une phrase qui se présente sous la forme d'une chaîne de caractères. Chaque caractère est significatif de la présence, dans la projection considérée, d'un maximum, d'un non maximum ou d'un vide.

FIGURE 3.1. Projections d'une horizontale, d'une verticale, d'une oblique.



★ La chaîne de caractères obtenue au niveau précédent est représentative de la structure de l'image à reconnaître mais également de certains paramètres (dimension, épaisseur des traits, ...). Nous appliquerons une transformation supplémentaire afin d'obtenir une chaîne de caractères significative de la classe en éliminant l'information sur la dimension, sur l'épaisseur des traits, en réduisant la part du bruit.

Le langage de description s'obtient donc à la suite de trois étapes [25] que nous grouperons ainsi :

- . génération de chaînes de caractères,
- . simplification de la chaîne de caractères,
- . normalisation de la chaîne.

3.1. GENERATION DE CHAÎNES DE CARACTERES

Le caractère que nous analysons est contenu dans une matrice binaire de 16×16 pixels.

3.1.1. DECOMPOSITION EN ELEMENTS D'ORDRE k

Rappelons que nous décomposons, tout d'abord, un caractère en éléments horizontaux (verticaux) d'ordre k .

Lors d'une analyse horizontale (de gauche à droite), le passage à un ordre supérieur ($k = k + 1$) dénote une discontinuité, ou espace suffisant, dans le caractère ; tous les points contigus, de l'image digitalisée, appar- tiendront à un même ordre.

De manière similaire, les éléments verticaux sont obtenus lors d'un balayage vertical de haut en bas.

La description d'un caractère sera constituée par l'ensemble des élé- ments horizontaux et verticaux d'ordre k soient 2×4 éléments au total pour les caractères alphanumériques latins qui possèdent au plus 4 ordres diffé- rents (pour la majorité des caractères, les éléments d'ordre supérieur à 2 seront vides).

On remarquera que le seul paramètre déterminant dans cette décomposi- tion est "l'espace suffisant", appelé ESPACE , nécessaire pour pouvoir détecter le passage à un ordre supérieur.

Si on donne à ce paramètre la valeur 1, une discontinuité d'un pixel dans l'image fait croître k lors de la décomposition. On détecte ainsi des ordres non significatifs. Dans l'analyse d'un caractère bruité (décomposition horizontale d'un R bruité; figure 3.2, page 22), un trou dans le caractère sera considéré, à tort, comme discontinuité du caractère.

Nous appellerons trou un pixel intérieur d'un trait digitalisé "0" (blanc) à tort. De même, on appellera excès un pixel codé "1" (noir) qui n'appartient pas au caractère.

Si au contraire le paramètre ESPACE prend une valeur élevée, par exem- ple $\text{ESPACE} > 3$ pixels, pratiquement aucun caractère ne pourra se prévaloir de posséder un espace suffisant pour passer à un ordre supérieur lors de la dé- composition. Ceci revient à négliger la notion d'ordre dans le caractère et à effectuer ultérieurement une projection brutale de celui-ci. Le figure 3.3 (page 22) illustre ce cas en décomposant horizontalement le même caractère R, mais en ayant adopté $\text{ESPACE} = 5$ pixels.

Compte tenu de la définition de 16×16 pixels, du fait que l'on peut rencontrer 4 ordres au maximum dans un caractère, nous prendrons le paramètre ESPACE égal à 2 points (figure 3.4, page 22) ; l'expérience confirmera ce choix.

Après décomposition, les éléments d'ordre k (que l'on appellera, par abus de langage, ordre k) sont projetés sur un axe vertical, pour les éléments horizontaux, et sur un axe horizontal pour les éléments verticaux.

FIGURE 3.2.
Décomposition horizontale
d'un R bruité.
ESPACE = 1 pixel.

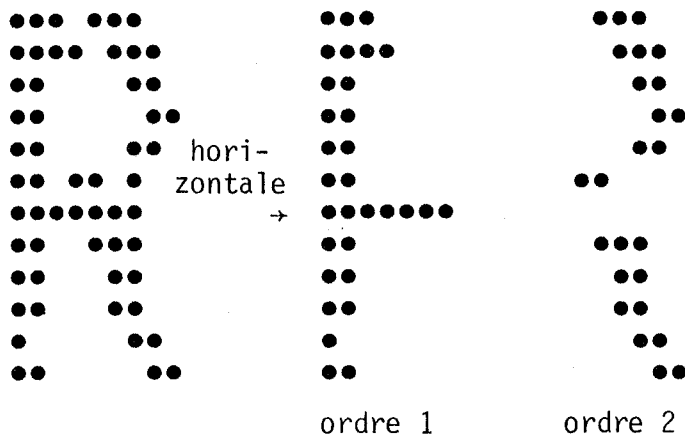


FIGURE 3.3.
Décomposition horizontale
d'un R bruité.
ESPACE = 5 pixels.

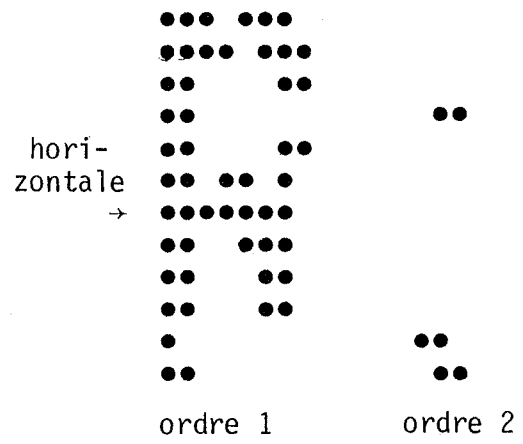
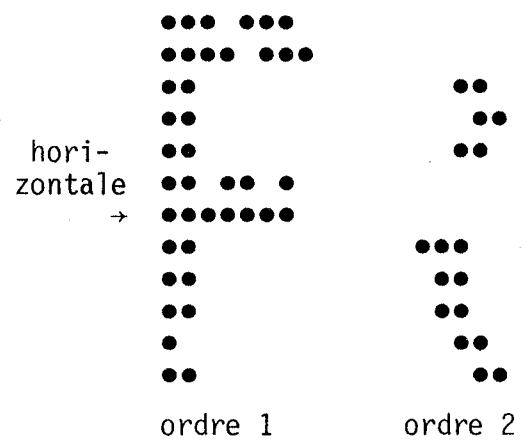


FIGURE 3.4.
Décomposition horizontale
d'un R bruité.
ESPACE = 2 pixels.



3.1.2. OBTENTION DE LA CHAÎNE DE CARACTÈRES

La projection orthogonale de l'élément horizontal d'ordre k se traduit par une suite L_{ik} représentant le nombre de "1" de la ligne i dans l'ordre k . Avec un élément vertical d'ordre k , on obtient la suite C_{jk} représentant le nombre de "1" de la colonne j dans l'ordre k (figure 3.5, page 24).

Les termes L_{ik} et C_{jk} seront traités, par la suite, de la même manière. Nous expliciterons les opérations effectuées sur les projections horizontales, c'est à dire les termes L_{ik} . Il va de soi que les termes C_{jk} issus des projections verticales subiront les mêmes opérations.

La présence d'un segment horizontal, donc d'une accumulation de "1" sur une ligne se traduit par une valeur élevée du terme L_{ik} correspondant. Il suffira donc de tester la valeur de L_{ik} en la comparant à un seuil S_k pour obtenir une information sur la présence d'un tel segment. Le seuil S_k tiendra compte de la dimension du caractère et de l'épaisseur des traits, par :

$$S_k = M_k + P_k ,$$

avec : $M_k = \frac{1}{N_k} \sum_{i=1}^{16} L_{ik}$: nombre moyen de points par ligne dans l'ordre k ,
 N_k : nombre de lignes dans l'ordre k possédant au moins 1 pixel,
 P_k : biais permettant d'assurer une bonne immunité au bruit ; l'expérience nous a conduit à choisir :
 $P_1 = 1$ pour l'ordre 1,
 $P_k = 2$ pour $k = 2, 3, 4$.

En reprenant l'exemple de la figure 3.5 (page 24), on obtient pour l'élément horizontal d'ordre 1 les valeurs suivantes : $N_1 = 14$; $M_1 = 4,1$; $S_1 = 6$.

A ce stade, la description de l'élément d'ordre k sera matérialisée par une chaîne de trois caractères M, F, Z, associés à L_{ik} selon les règles :

$$\begin{array}{l} L_{ik} \geq S_k \Rightarrow M : \text{maximum} \\ 0 < L_{ik} < S_k \Rightarrow F : \text{non maximum} : \text{forme} \\ L_{ik} = 0 \Rightarrow Z : \text{absence de points} : \text{zéro} . \end{array}$$

Les caractères ont pour signification :

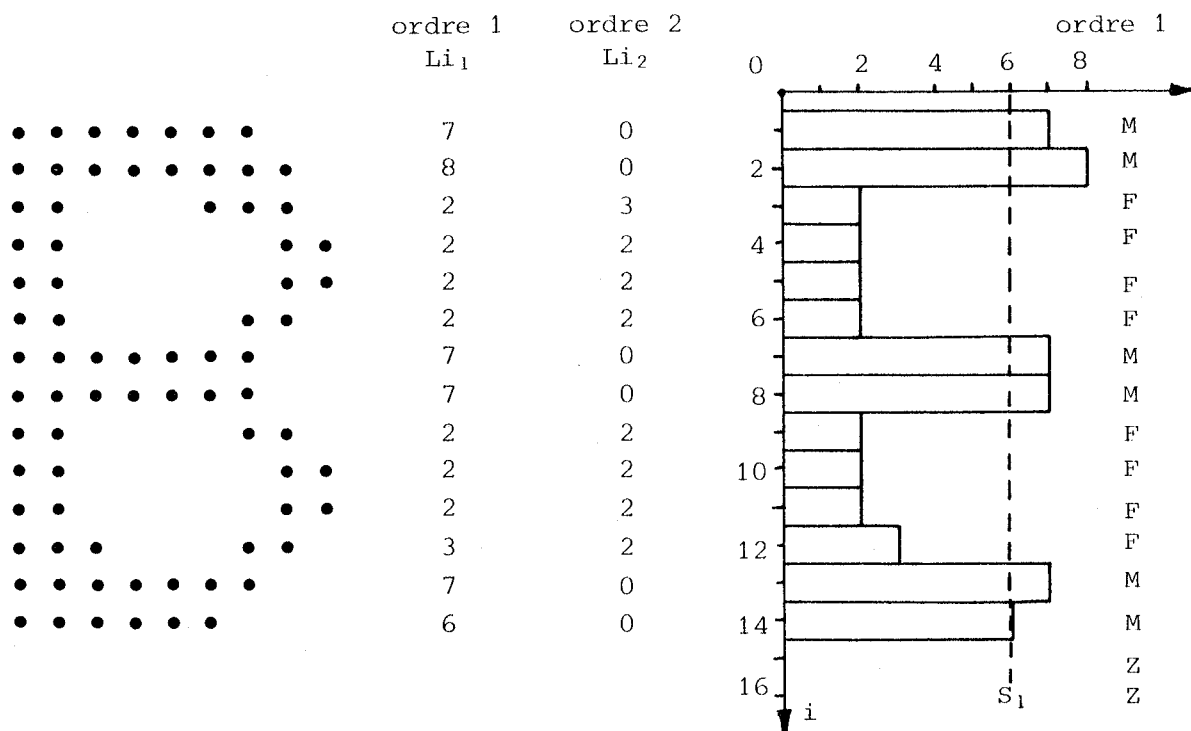
- M : présence d'un segment horizontal,
- F : présence d'un petit segment horizontal, d'un segment perpendiculaire au sens d'analyse, d'une oblique, d'un arc de cercle, de bruit non éliminé au préalable,
- Z : représente l'ensemble vide.

Une chaîne comportera ainsi 16 caractères par élément. En reprenant l'exemple de la figure 3.5, l'élément horizontal d'ordre 1 est décrit par la chaîne :

MMFFFFMMFFFFMMZZ

Un caractère est décrit par 8 chaînes de ce type (4 chaînes horizontales et 4 chaînes verticales).

FIGURE 3.5. Génération de la chaîne de caractère de l'élément horizontal d'ordre 1.



Les règles d'association sont, à ce niveau, très simples ; on distingue deux cas :

- ★ au sein d'une chaîne :
 - ☆ dans les ordres supérieurs (k = 2, 3, 4), toutes les combinaisons sont possibles,
 - ☆ à l'ordre 1, toutes les combinaisons sont également possibles ; cependant, la présence d'au moins 2 caractères Z consécutifs au milieu des groupements M,F témoigne d'une coupure du caractère. Dans ce cas, on provoque un rejet car, compte tenu du prétraitement, ceci est significatif d'une coupure très importante dans l'image originale.
- ★ sur l'ensemble des 4 chaînes horizontales et verticales :
 - une chaîne d'ordre k ne peut être égale à Z¹⁶ (chaîne vide) si une chaîne d'ordre k+1 contient un caractère différent de Z.

3.2. SIMPLIFICATION DE LA CHAÎNE DE CARACTERES

En pratique, les caractères ne sont pas d'aussi bonne qualité que celui de la figure 3.5. Le bruit, ayant échappé au prétraitement, n'ayant pas été éliminé par le seuil S_k lié à la moyenne M_k , a tendance à compliquer singulièrement la structure des phrases de description. Afin de réduire son influence, nous avons établi des règles de simplification. Ces règles peuvent se séparer en deux types :

- . simplification de chaque chaîne indépendamment des autres,
- . simplification d'une chaîne au vu de l'ensemble des chaînes d'un même sens d'analyse.

NOTATION :

la chaîne décrivant un élément horizontal d'ordre k sera notée H_k ,
la chaîne décrivant un élément vertical d'ordre k sera notée V_k .

3.2.1. SIMPLIFICATION D'UNE CHAÎNE

Une série de caractères présentant chacun un problème particulier marqué par "←" est présenté à la figure 3.6.

FIGURE 3.6. Caractères bruités.

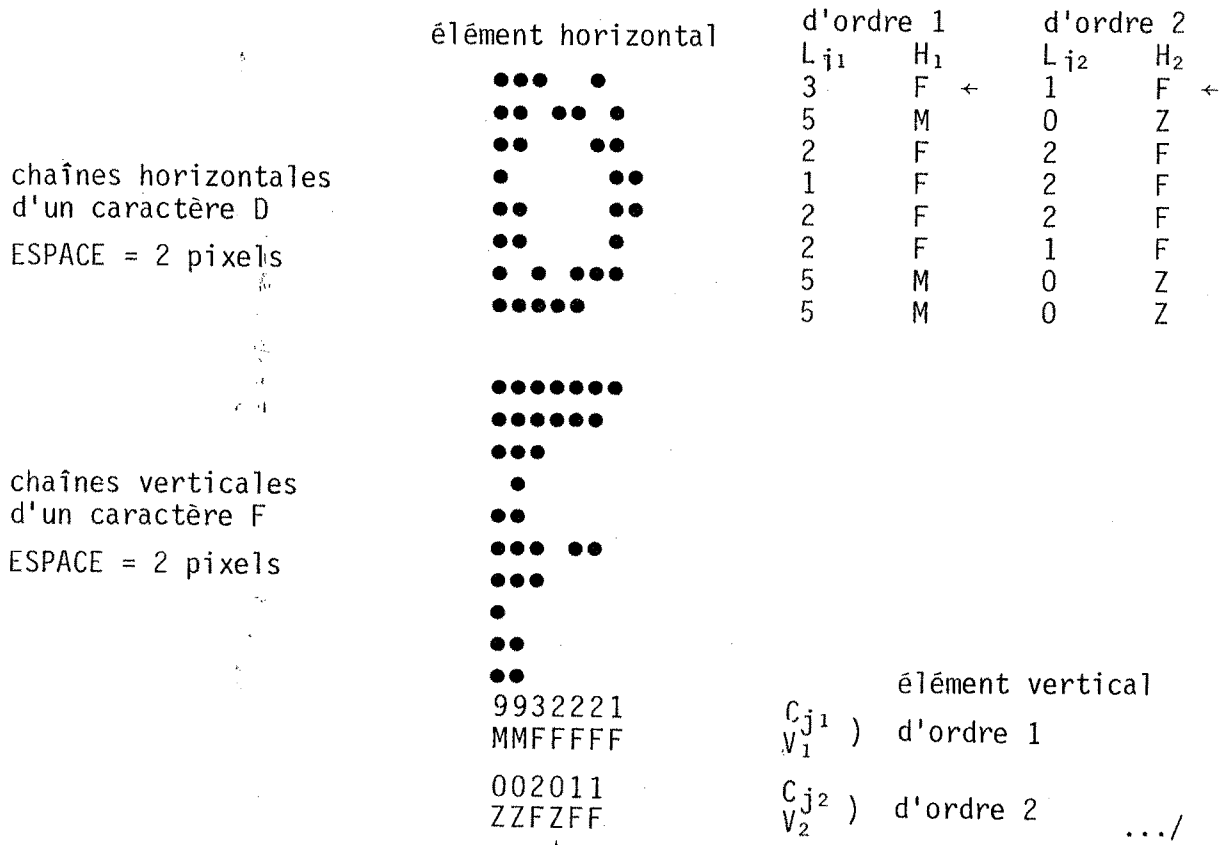
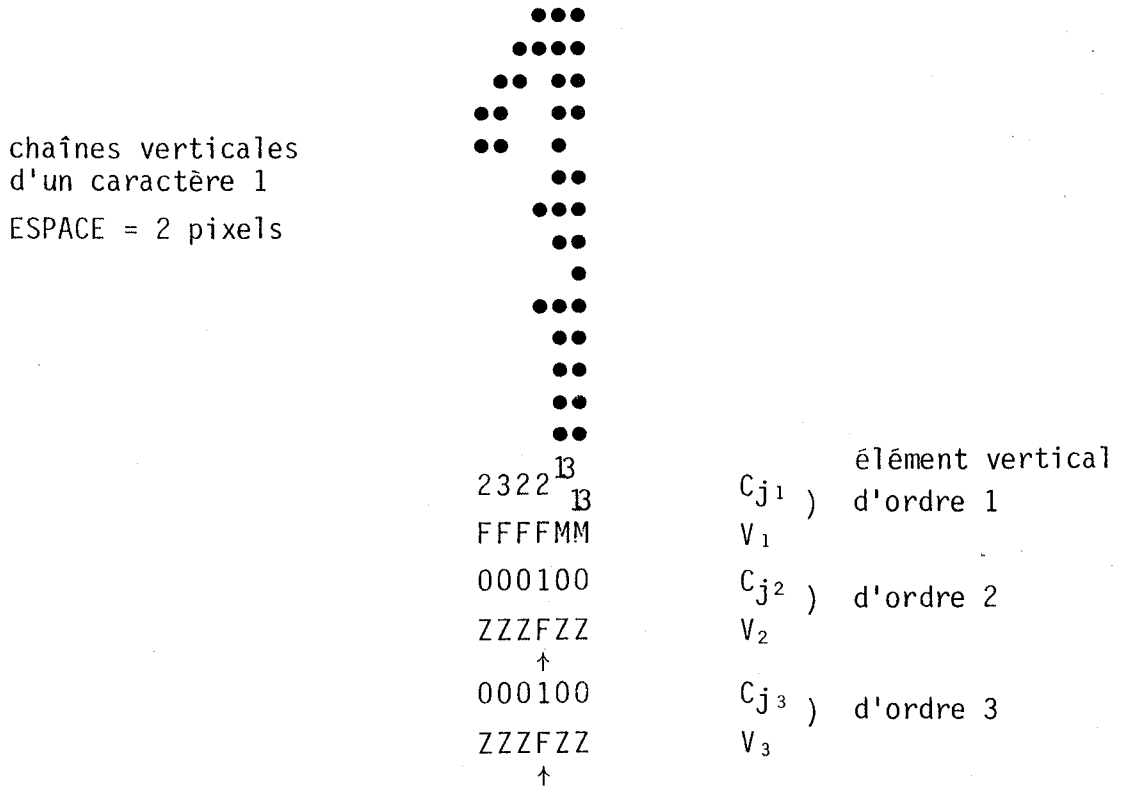


FIGURE 3.6 (suite).



Les simplifications vont s'effectuer tour à tour sur les huit chaînes. La nature (M, F ou Z) d'une ligne i (ou colonne j) d'une chaîne d'ordre k a été fixée en examinant le nombre L_{ik} (C_{jk}).

Pour une chaîne (k fixé), on corrigera éventuellement cette information en prenant en compte les lignes voisines qui ne sont pas indépendantes de la ligne i. Pour corriger la nature de cette ligne i, nous examinerons les deux lignes précédentes et les deux suivantes. Or, l'analyse des lignes s'effectuant pour des indices i croissants, les deux lignes i-1 et i-2 ont déjà subi une correction ; de ce fait, la ligne i-1 possède des informations fonctions des lignes précédentes. Pour corriger la ligne i, nous regarderons la nature des lignes i-1, i+1 et i+2. Les règles de simplification sont présentées dans le Tableau 1 (page 27).

En reprenant les exemples de la figure 3.6, l'application des règles du Tableau 1 conduit aux résultats suivants :

$$\begin{array}{l}
 \text{pour le D : } H_1 : Z^a F M F F F F M M Z^b \xrightarrow{(5)} Z^a M M F F F F M M Z^b \\
 \quad \quad \quad H_2 : Z^a F Z F F F F Z Z Z^b \xrightarrow{(14)} Z^a F F F F F F Z Z Z^b \\
 \text{pour le F : } V_2 : Z^c Z Z F Z F F Z^d \xrightarrow{(14)} Z^c Z Z F F F F Z^d \\
 \text{pour le 1 : et } \left. \begin{array}{l} V_2 \\ V_3 \end{array} \right\} Z^e Z Z Z F Z Z Z^f \xrightarrow{(1)} Z^e Z Z Z Z Z Z Z^f = Z^{16}
 \end{array}$$

TABLEAU 1. Simplification d'une chaîne de caractères.

$$C_{n-1} C_n C_{n+1} C_{n+2} \rightarrow C_{n-1} C'_n C_{n+1} C_{n+2}$$

- $C_n = M$
 C_n reste inchangé dans tous les cas.
- $C_n = F$

(1)	Z F Z Z	→	Z Z Z Z			
(2)	Z F Z M	→	Z Z Z M	si	$S_k > 3$	
(3)	Z F Z M	→	Z F Z M	si	$S_k \leq 3$	
(4)	Z F M X	→	Z Z M X	si	$L_{nk} \leq 2$	X = indifférent
(5)	Z F M X	→	Z M M X	si	$L_{nk} > 2$	
(6)	M F Z Z	→	M Z Z Z	si	$L_{nk} \leq 2$	
(7)	M F Z Z	→	M M Z Z	si	$L_{nk} > 2$	
(8)	M F Z M	→	M Z Z M	si	$S_k > 3$	
(9)	M F Z M	→	M F Z M	si	$S_k \leq 3$	
(10)	M F M X	→	M M M X			

C_n reste inchangé dans les autres cas.
- $C_n = Z$

(11)	M Z F X	→	M Z Z X			
(12)	M Z F X	→	M F F X	si	$S_k \leq 3$	
(13)	M Z M X	→	M M M X			
(14)	F Z F X	→	F F F X			
(15)	F Z M X	→	F F M X			
(16)	F Z Z M	→	F Z Z M	si	$S_k > 3$	
(17)	F Z Z M	→	F F Z M	si	$S_k \leq 3$	

C_n reste inchangé dans les autres cas.

Ces règles systématiques éliminent un nombre important de problèmes mais il en demeure car, en analysant les chaînes indépendamment les unes des autres, on perd l'information contenue dans les chaînes voisines. La prochaine étape consiste à simplifier un caractère d'une chaîne en considérant les autres ordres.

3.2.2. SIMPLIFICATION D'UN CARACTERE D'UNE CHAÎNE EN FONCTION DES ORDRES k ET k+1

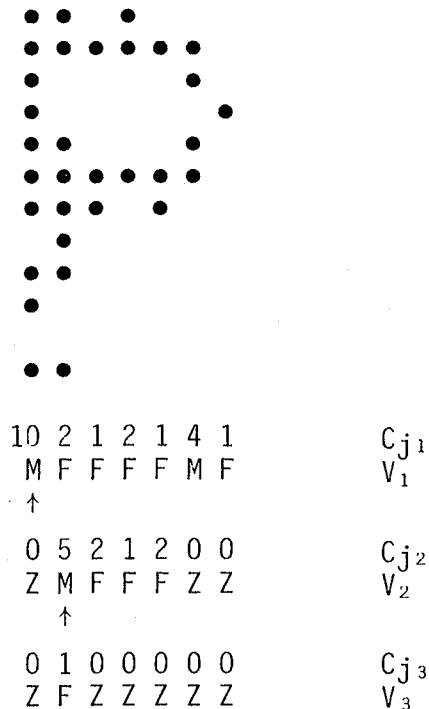
Après avoir effectué la simplification des 8 chaînes prises séparément, nous examinons un groupement de caractères d'une chaîne d'ordre k en fonction du groupement de même rang (même indice ligne i), mais de la chaîne d'ordre k+1.

En considérant que la plus grande partie de l'information se rencontre dans les chaînes d'ordre 1 et 2, les règles de simplification ne porteront que sur celle-ci. En appliquant ces règles, nous visons essentiellement deux buts :

- .améliorer la détection des segments de droites entachés de bruit,
- .corriger les "erreurs" commises lors de la simplification d'une chaîne (Tableau 1, page 27), au cours de laquelle on n'a pas pris en compte le sens d'analyse ordre 1 ↔ ordre 2.

Le caractère P de la figure 3.7 illustre (←) les problèmes qu'occasionne le bruit (trous) sur un segment de droite.

FIGURE 3.7. Traitement d'un caractère P bruité.



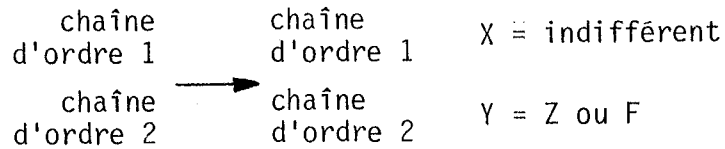
La présence de ce segment bruit se signale dans les chaînes V₁ et V₂. Ces problèmes vont être résolus par les simplifications du Tableau 2 (p. 29).

REMARQUES :

- . On note, en examinant le Tableau 2, que tous les cas ne sont pas explicités car certaines associations sont impossibles, eu égard aux simplifications que la chaîne a subies auparavant (Tableau 1).
- . Les cas n° 1, 4 peuvent sembler impossibles. On note par exemple dans le cas 1 la présence d'un caractère M dans une chaîne d'ordre 2 sous un caractère Z d'une chaîne d'ordre 1. Ce paradoxe provient de la règle (4) du tableau de simplification d'une chaîne, à savoir pour une chaîne d'ordre 1 :

$$Z F M X \xrightarrow{(4)} Z Z M X \quad \text{si } L_{nk} \leq 2 .$$

TABLEAU 2. Simplification de lignes (ou colonnes) en fonction des chaînes d'ordre 1 et 2.



1) ARRANGEMENT DES SEGMENTS DE DROITE

$$\begin{matrix} Z & Z & M \\ Z & M & Y \end{matrix} \longrightarrow \begin{matrix} Z & M & M \\ Z & Z & Z \end{matrix} \quad (1)$$

$$\begin{matrix} F & F & M \\ Y & M & X \end{matrix} \longrightarrow \begin{matrix} F & M & M \\ Y & Z & Z \end{matrix} \quad (2)$$

$$\begin{matrix} M & F & F \\ Y & M & X \end{matrix} \longrightarrow \begin{matrix} M & M & F \\ Z & Z & X \end{matrix} \quad (3)$$

$$\begin{matrix} M & Z & Z \\ Y & M & Z \end{matrix} \longrightarrow \begin{matrix} M & M & Z \\ Z & Z & Z \end{matrix} \quad (4)$$

2) CAS PARTICULIER

$$\begin{matrix} M & F & F & M \\ Y & M & X & X \end{matrix} \longrightarrow \begin{matrix} M & M & M & M \\ Z & Z & X & X \end{matrix}$$

$$\begin{matrix} M & F & F & M \\ X & X & M & Y \end{matrix} \longrightarrow \begin{matrix} M & M & M & M \\ X & X & Z & Z \end{matrix}$$

3. CORRECTION D'UNE SIMPLIFICATION DE CHAÎNE

$$\begin{matrix} F & M & F \\ F & Z & F \end{matrix} \xrightarrow[\text{de chaîne (14)}]{\text{simplification}} \begin{matrix} F & M & F \\ F & F & F \end{matrix} \longrightarrow \begin{matrix} F & M & F \\ F & Z & F \end{matrix}$$

initialement tab.1 tab.2

L'application des règles de simplification d'une chaîne (tab.1) puis de simplification d'une ligne (tab.2) va simplifier les chaînes V_k du caractère P de la figure 3.7 (page 28) :

$$\begin{aligned} \bullet V_1 & : Z^a M F F F F M F Z^b \xrightarrow{(6) \text{ tab.1}} Z^a M F F F F M Z Z^b \\ V_3 & : Z^a Z F Z Z Z Z Z Z^b \xrightarrow{(1) \text{ tab.1}} Z^a Z Z Z Z Z Z Z Z^b = Z^{16} \\ \bullet V_1 & : Z^a M F F F F M Z Z^b \xrightarrow{(3) \text{ tab.2}} Z^a M M F F F M Z Z^b \\ V_2 & : Z^a \underbrace{Z M F F F Z Z}_{} Z^b \xrightarrow{(3) \text{ tab.2}} Z^a Z Z F F F Z Z Z^b \end{aligned}$$

Nous avons appliqué aux associations de caractères des règles systématiques en considérant l'information qui existe entre les caractères au sein d'une chaîne, et l'information présente entre une chaîne d'ordre 1 et d'ordre 2. Nous n'avons pas exploité la mutuelle information qui existe entre

l'ensemble des quatre chaînes horizontales et l'ensemble des quatre chaînes verticales. Vouloir extraire cette information nous obligerait à entrer dans la structure du caractère, problème qui sera abordé au niveau du classifieur.

A la fin de cette étape, un caractère alphanumérique est décrit par :

- . 4 chaînes H_1, \dots, H_4 de 16 caractères M, F, Z relatives aux projections horizontales ;
- . 4 chaînes V_1, \dots, V_4 de 16 caractères M, F, Z relatives aux projections verticales.

Ces chaînes H_k (et V_k) sont indépendantes de la dimension horizontale (et verticale) puisque nous avons effectué une projection. Toutefois, elles possèdent encore une information sur la dimension verticale (et horizontale). Nous allons éliminer cette information en normalisant les chaînes.

3.3. NORMALISATION DES CHAÎNES DE CARACTÈRES

L'information présente dans les chaînes de caractères est redondante ; en effet, si l'on détecte par exemple un segment de droite, il va se traduire par une suite M^a avec $1 \leq a \leq 4$ en moyenne. Si l'on ne désire conserver que l'information de présence, et non la largeur, du segment de droite, il est possible de réduire la suite M^a en une primitive m . Cette réduction va conduire à définir trois primitives dont la signification est voisine de celle des caractères M, F, Z :

- .m : présence d'un segment de droite normalisé ;
- .f : présence d'un petit segment de droite normalisé parallèle au sens d'analyse, présence d'un segment normalisé perpendiculaire au sens d'analyse, présence d'une oblique, présence d'un arc de cercle, présence de bruit ;
- .z : ensemble vide.

Le rapport de réduction est choisi, automatiquement, en fonction de la composition des chaînes $H_1 ; H_2$ d'une part et $V_1 ; V_2$ d'autre part. On obtient des phrases de description $h_1 ; h_2$ et $v_1 ; v_2$ selon les règles présentées dans le Tableau 3 (page 31).

NOTATION :

- Φ_k représente indépendamment une chaîne H_k ou V_k (avant normalisation),
- X = M, F ou Z,
- Y = M, F ou Z,
- ϕ_k représente une phrase horizontale ou verticale (h_k ou v_k) (après normalisation).

TABLEAU 3. Normalisation des ordres 1 et 2.

★ \exists une suite $\langle M^a \rangle \in \Phi_1$

on examine dans Φ_2 l'intervalle "a" correspondant

☆ \exists une suite $\langle X^b M^c Y^d \rangle$ tel que $a = b + c + d$; X, Y caractères indifférents

alors $\langle M^a \rangle \in \Phi_1 \rightarrow m \in \varphi_1$

$\langle X^b M^c Y^d \rangle \in \Phi_2 \rightarrow m \in \varphi_2$

☆ \exists une suite $\langle F^b Z^{a-b} \rangle$ ou $\langle Z^{a-b} F^b \rangle$

alors $\langle M^a \rangle \in \Phi_1 \rightarrow m \in \varphi_1$

$$\langle F^b Z^{a-b} \rangle \in \Phi_2 \left\{ \begin{array}{l} \rightarrow f \text{ si } b > a-b \\ \rightarrow z \text{ si } b < a-b \\ \text{si } b = a-b \\ \rightarrow z \text{ pour } v_k \\ \rightarrow f \text{ pour } h_k \end{array} \right.$$

★ \exists une suite $\langle F^a \rangle \in \Phi_1$

on examine dans Φ_2 l'intervalle "a" correspondant

☆ \exists une suite $\langle M^b \rangle$

on crée un nouvel intervalle de largeur b') $\left. \begin{array}{l} \text{si } a-b \neq 1 \rightarrow b' = b \\ \text{si } a-b = 1 \rightarrow b' = a \end{array} \right\}$

$\langle F^{b'} \rangle \in \Phi_1 \rightarrow f \in \varphi_1$

$\langle M^b \rangle \in \Phi_2 \rightarrow m \in \varphi_2$

☆ \exists une suite $\langle F^c \rangle$

on crée un nouvel intervalle de largeur c') $\left. \begin{array}{l} \text{si } a-c \neq 1 \rightarrow c' = c \\ \text{si } a-c = 1 \rightarrow c' = a \end{array} \right\}$

$\langle F^{c'} \rangle \in \Phi_1 \rightarrow f \in \varphi_1$

$\langle F^c \rangle \in \Phi_2 \rightarrow f \in \varphi_2$

☆ \exists une suite $\langle Z^d \rangle$

on crée un nouvel intervalle de largeur d') $\left. \begin{array}{l} \text{si } a-d \neq 1 \rightarrow d' = d \\ \text{si } a-d = 1 \rightarrow d' = a \end{array} \right\}$

$\langle F^{d'} \rangle \in \Phi_1 \rightarrow f \in \varphi_1$

$\langle Z^d \rangle \in \Phi_2 \rightarrow z \in \varphi_2$

★ \exists une suite $\langle Z^a \rangle \in \Phi_1$

soient $A = M$ ou F ; $B = M$ ou F

☆ en bordure de la chaîne de caractères M, F :

$\langle Z^a \rangle \in \Phi_1 \rightarrow z \in \varphi_1$

en général, dans Φ_2 , on doit avoir : $\langle Z^a \rangle \in \Phi_2 \rightarrow z \in \varphi_2$

☆ si on trouve $\langle A^a Z^b B^c \rangle \in \Phi_1$, avec $b \geq 2$, c'est significatif d'une coupure du caractère \rightarrow rejet.

Les chaînes d'ordre 3 et 4, qui contiennent moins d'information, sont réduites selon le Tableau 4. On remarquera que, pour les chaînes d'ordre 3 et 4 qui n'interviennent pas dans le choix du rapport de réduction, la décision est à caractère majoritaire entre Z et F.

TABLEAU 4. Normalisation des ordres 3 et 4.

Soit "a" la largeur d'un intervalle de la phrase φ_1 défini selon les règles du Tableau 3 (page 31).

*	\exists	une suite	$\langle X^b M^c X^d \rangle$	$\in \Phi_3$ ou Φ_4	\rightarrow	$m \in \varphi_3$ ou φ_4	$a = b + c + d$
			$\langle F^b Z^{a-b} \rangle$	$\left. \vphantom{\langle F^b Z^{a-b} \rangle} \right\} \in \Phi_3$ ou Φ_4	$\left\{ \begin{array}{l} \rightarrow \\ \rightarrow \\ \rightarrow \end{array} \right.$	$f \in \varphi_3$ ou φ_4	si $b \geq a - b$
			$\langle Z^{a-b} F^b \rangle$			$z \in \varphi_3$ ou φ_4	si $b < a - b$.

La figure 3.8 explicite la décomposition horizontale, la simplification, la normalisation des éléments horizontaux d'un caractère G.

FIGURE 3.8. Etude de la normalisation d'un caractère G.

	L_{i1}		H_1	h_1		L_{i2}	H_2	h_2
●●●●	4	F	M	} m		0	Z	Z
●●● ●●●●	7	M	M			0	Z	Z
●●● ●●●●	3	F	F	} f		3	F	F
●●● ●●●●	3	F	F			1	F	F
●●● ●●●●	3	F	F	} f		0	Z	Z
●●● ●●●●	3	F	F			0	Z	Z
●●● ●●●●	2	F	F	: f		5	M	M
●●● ●●●●	3	F	F	} f		3	F	F
●●● ●●●●	3	F	F			1	F	F
●●● ●●●●	3	F	F	} f		2	F	F
●●● ●●●●	3	F	F			2	F	F
●●●●●●●●	7	M	M	} m		0	Z	Z
●●● ●●●	5	M	M			0	Z	Z

Les phrases de description horizontale de ce caractère seront :

- (z) m f f f f m (z)
- (z) z f z m f z (z)

Nous n'avons pas représenté les phrases z^6 d'ordre 3 et 4. De même, selon la dimension et le cadrage du caractère dans la matrice 16×16 , ces phrases seront encadrées ou non par des "z". Si l'on fait abstraction de ces primitives, la phrase d'ordre 1 sera de longueur supérieure ou égale à celles des autres ; une phrase d'ordre 1 ne peut pas contenir de primitives z parmi des primitives m, f. Un caractère sera décrit à l'aide de primitives m, f, z par huit phrases : 4 horizontales et 4 verticales. De par les règles de réduction et la structure des caractères alphanumériques, une phrase ne contiendra pas plus de 8 primitives. On utilisera donc au plus 64 primitives pour décrire un caractère ; mais on notera que les chaînes d'ordre 3 et 4 sont souvent vides.

Dans la suite, nous conviendrons de ne représenter que les phrases non vides, et de supprimer les primitives z encadrant les primitives m, f des phrases d'ordre 1. On supprimera également dans les phrases d'ordre 2, 3, 4 les primitives z débordant du cadre défini par la phrase d'ordre 1.

Un organigramme général (figure 3.9, page 34) résume les différentes opérations qui sont effectuées en vue de décrire un caractère inconnu par des primitives.

3.4. ESSAI DE L'ORGANE DE TRAITEMENT

Afin de tester les propriétés de l'organe de traitement, nous avons procédé à une série d'essais. Nous recherchons essentiellement à observer :

- . l'invariance de la description avec la translation du caractère,
- . l'insensibilité aux homothéties,
- . une tolérance vis à vis de petites rotations,
- . une relative invariance des descriptions devant appartenir à une même classe.

3.4.1. DESCRIPTION DE L'APPAREILLAGE

Les opérations décrites par l'organigramme précédent ont été implémentées sur un dispositif micro-informatique élaboré autour du microprocesseur 2650 de SIGNETICS. Le programme de traitement a nécessité une capacité mémoire de 2 Koctets. Nous avons d'autre part développé un système d'échange avec l'organe d'acquisition construit avec le microprocesseur 6800 de MOTOROLA. L'organe de traitement est maître et pilote l'appareil d'acquisition.

Afin de visualiser, d'exploiter les résultats des descriptions des caractères, nous avons développé un système de gestion permettant de contrôler les différentes étapes de traitement. Ce programme offre les possibilités suivantes :

- . visualisation du caractère digitalisé,
- . visualisation et modification des paramètres,
- . présentation des résultats L_{ik} , C_{jk} ,
- . visualisation des chaînes de caractères,
- . visualisation des phrases de description.

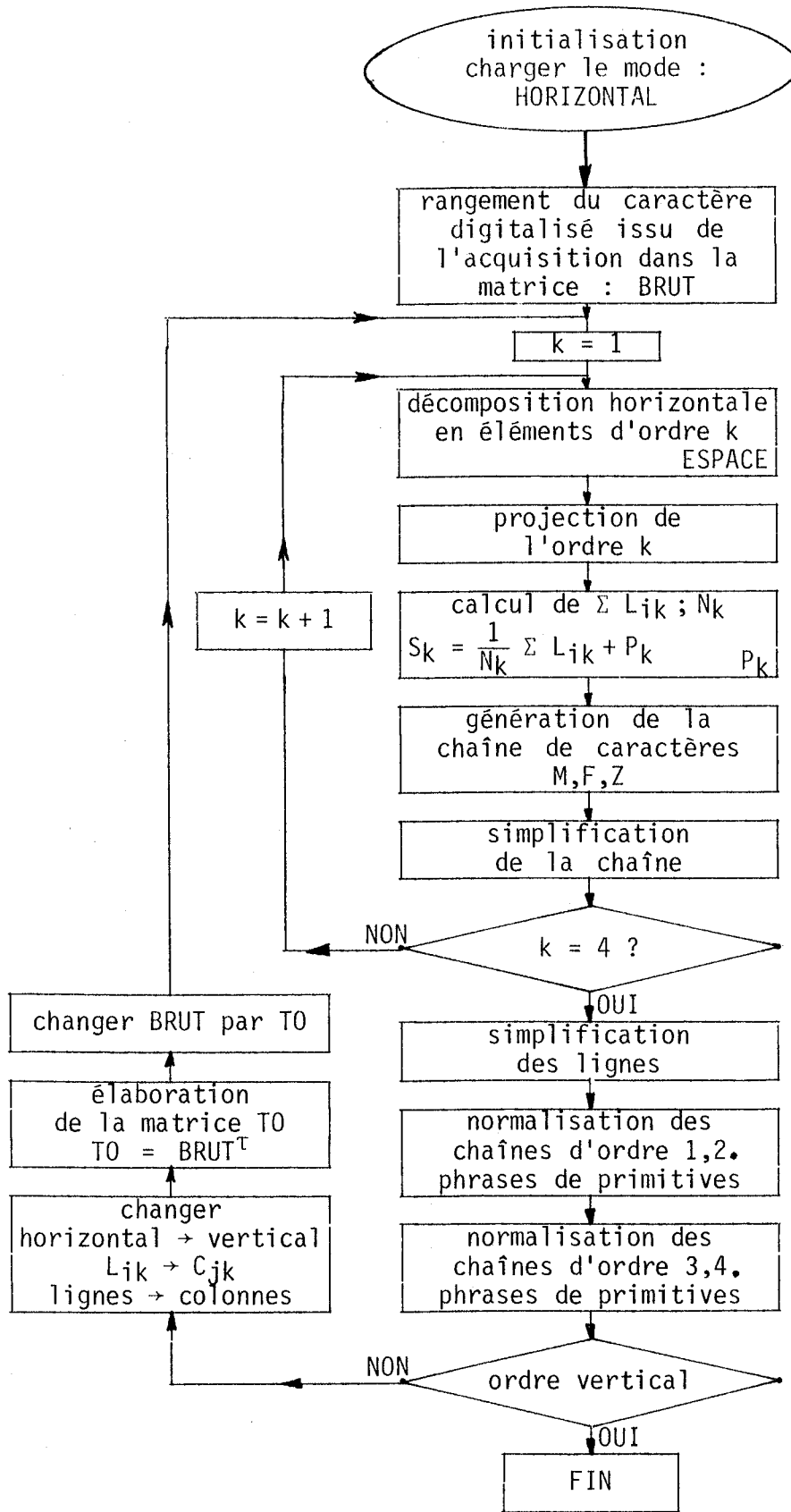


FIGURE 3.9. Organigramme général du traitement d'un caractère.

De plus, ce programme de gestion renseigne l'opérateur sur l'état du traitement en indiquant certaines erreurs : le nombre d'ordre k est > 4 , un caractère est coupé, il y a plus de 8 primitives dans une phrase...

Ce système de gestion occupe également 2 Koctets de mémoire.

3.4.2. DESCRIPTION DES CARACTERES

Nous avons soumis plusieurs alphabets au traitement : un alphabet de caractères "Letraset", de très bonne qualité, puis des caractères dactylographiés et manuscrits.

Sur des caractères de "bonne qualité" (définie a contrario par ce qui suit), on obtient une description quasi unique, quelle que soit la fonte dont est issu le caractère. La figure 3.10 (page 36) illustre la description de trois caractères D (Letraset, dactylographié, manuscrit) ; ces trois graphismes sont bien décrits par un ensemble de phrases identiques.

Nous constatons, par contre, des problèmes dans les cas suivants :

☆ caractères représentés avec des traits épais

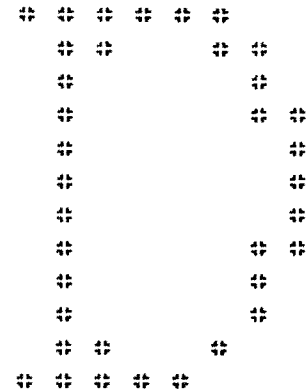
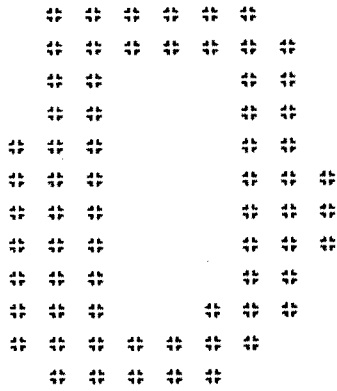
A l'ordre 1, les maximums sont bien détectés grâce au seuil S_k qui, tel qu'il est choisi, tient compte de l'épaisseur des traits. La description normalisée fait bien abstraction des propriétés dimensionnelles. Cependant, pour une dimension de caractère donné, l'épaississement des traits provoque une réduction de l'espace inter-ordres. Les éléments d'ordre supérieur à 1 peuvent se trouver éliminés (les pixels correspondants se trouvent inclus dans l'ordre 1). La description totale, comprenant les phrases relatives à ces ordres, se trouve donc fortement perturbée (figure 3.11, page 37).

☆ caractères représentés avec des traits fins

On trouve un problème de même nature que précédemment. Cette fois-ci, l'espace inter-ordre est accru ; en contre-partie, des coupures apparaissent et ceci provoque l'apparition inconsidérée d'élément d'ordre supérieur. Là encore, la description du caractère s'en trouve perturbée.

On peut résoudre ces deux types de problème en jouant par exemple sur le paramètre ESPACE pour le diminuer ou l'augmenter selon l'épaisseur des traits. Cependant, ce paramètre pour l'instant fixé à 2 pixels ne peut évoluer dans de grandes proportions. Rappelons que, compte tenu de la dimension maximale de 16×16 pixels et du nombre d'éléments d'ordre k pouvant atteindre 4, le paramètre ESPACE ne peut guère prendre une valeur supérieure à 4.

FIGURE 3.10.



HORIZONTAL
ZMMFFFFFFFFFMMZZZ
ZZZFFFFFFFFFZZZZZ
ZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZ

VERTICAL
ZZMMMFFMMZZZZZZZ
ZZZZZFFFZZZZZZZZZ
ZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZ

HORIZONTAL
ZMFFFFFFFFFMMZZZ
ZZFFFFFFFFFZZZZ
ZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZ

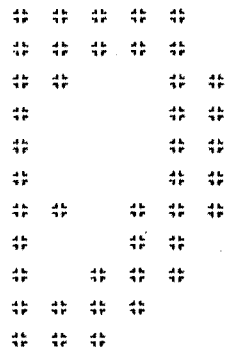
VERTICAL
ZZZMFFFFFFFFMZZZZZ
ZZZFZFFFFFFZZZZZZ
ZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZ

HORIZONTAL
MFM
ZFZ

VERTICAL
MFM
ZFZ

HORIZONTAL
MFM
ZFZ

VERTICAL
MFM
ZFZ



HORIZONTAL
ZZMMFFMMMMZZZ
ZZZZFFFFFFFFZZZZZ
ZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZ

VERTICAL
ZZZMFFMMZZZZZZZZZ
ZZZZFFFZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZ

HORIZONTAL
MFM
ZFZ

VERTICAL
MFM
ZFZ

FIGURE 3.11. Traitement d'un 8 ; émission d'élément d'ordre k.

```

      * * *
    * * * * *
  * *   * * *
 * *   * * *
 * *   * * *
 * * * * *
   * * * *
 * * * * *
 * *   * *
 * *   * *
 * *   * *
 * * * * *
   * * *
  
```

```

HORIZONTAL
FFFFFFFFMFFMZZZ
ZZZZZZZZZFFFZZZZ
ZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZ

```

```

VERTICAL
ZZZMMFFMZZZZZZZ
ZZZZZFFZFZZZZZZ
ZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZ

```

```

HORIZONTAL
FMFM
ZZFZ

```

```

VERTICAL
MFM
ZFZ

```

Une variation du paramètre ESPACE provoquera immédiatement une modification importante de la description, sans qu'il soit possible de contrôler aisément cette modification. De plus, le paramètre ESPACE ne peut résoudre les problèmes les plus aigus, à savoir : caractères très empatés (agglomérat de points) ou caractères coupés.

On peut également résoudre ces problèmes, à la base, en agissant sur le seuil d'acquisition et nous verrons que cette solution est réaliste.

3.5. DETERMINATION DU SEUIL D'ACQUISITION OPTIMAL

Rappelons que le système d'acquisition digitalise tous les points issus de la caméra T.V. et qu'il en mémorise une partie. Ces points inscrits dans une fenêtre de taille 64×128 sont rangés dans la matrice binaire $F(i,j)$. Pour réduire et lisser la matrice $F(i,j)$ (64×128), le système va réduire un groupe de 4×4 points en un seul point qui est un élément d'une matrice $G(k,l)$ de taille 16×32 . Un élément de $G(k,l)$ représente la somme des 16 points du carré de 4×4 points pris dans $F(i,j)$. Les informations de $F(i,j)$ étant binaires, les éléments de $G(k,l) \in [0,16]$. La matrice binaire $g(k,l)$ (16×32) est obtenue après seuillage :

$$\text{si } G(k,l) > S_{\text{acq}} \Rightarrow g(k,l) = 1$$

$$G(k,l) \leq S_{\text{acq}} \Rightarrow g(k,l) = 0$$

$$0 \leq S_{\text{acq}} < 16 .$$

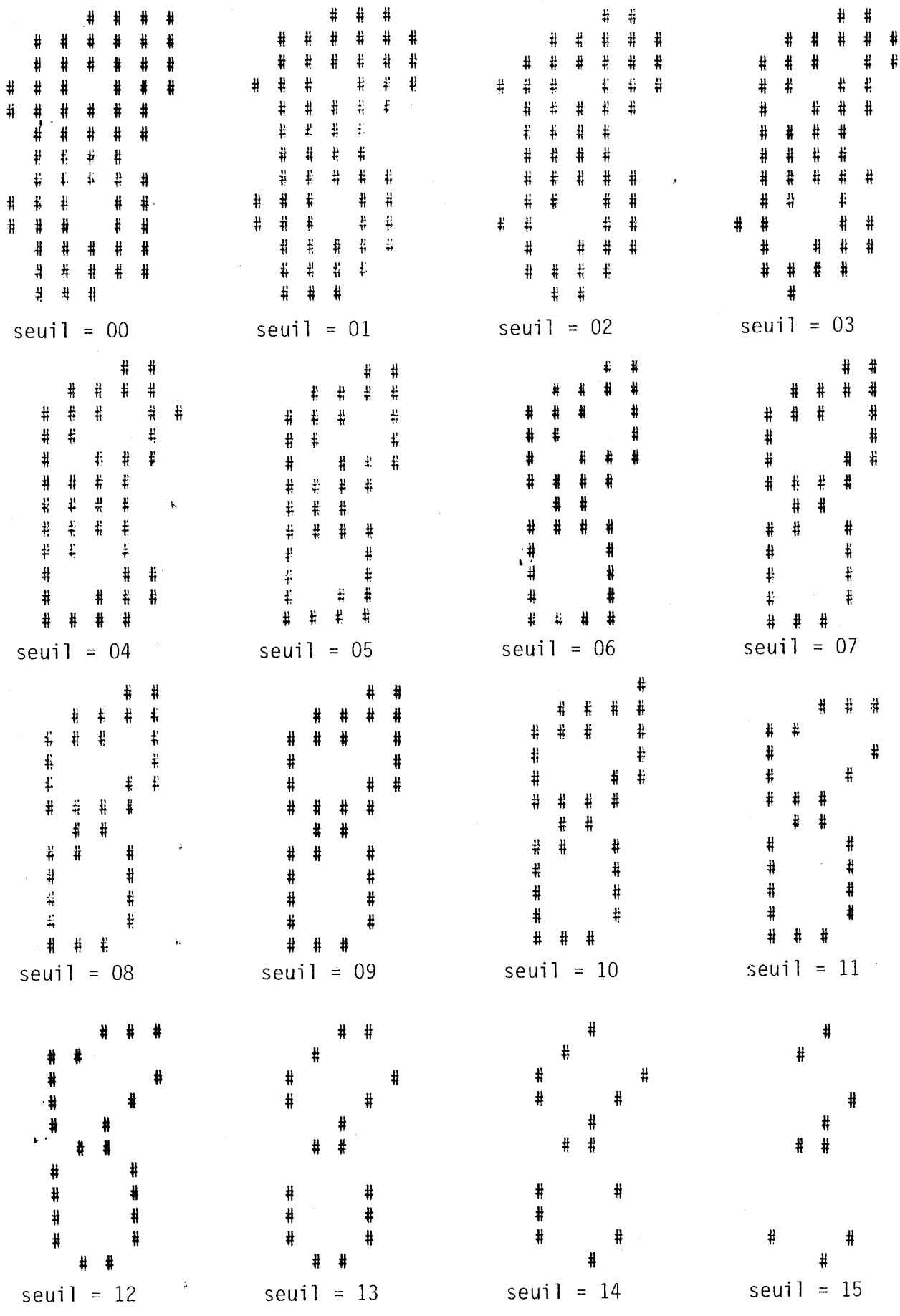
De la relation binaire $g(k,l)$, on crée la matrice $g'(k,l)$ (16×16) en isolant un caractère. Cette matrice g' est ensuite livrée à l'organe de traitement. En faisant varier le seuil S_{acq} , il est donc possible d'obtenir 16 matrices g' issues d'une même matrice F . Nous avons représenté à la figure 3.12 (page 39) l'évolution de la matrice g' en fonction des valeurs prises par le seuil S_{acq} pour un caractère 8.

Si le seuil est de faible valeur ($0 \leq S_{\text{acq}} < 3$), le caractère digitalisé a tendance à s'empâter, les segments de droite ont une largeur de 3 à 4 pixels. A l'opposé, si le seuil prend des valeurs élevées ($S_{\text{acq}} > 12$), le caractère se squelettise puis se coupe. Parmi les 16 valeurs de S_{acq} , il faut en choisir une qui donne, si possible, le caractère le plus adapté au traitement qu'il subit après l'acquisition.

De manière classique, si l'on doit choisir un seuil d'acquisition, donc une matrice g' dans laquelle est inscrit un caractère, l'on effectue au niveau de l'organe d'acquisition des tests permettant de sélectionner le caractère digitalisé le plus approprié au traitement qui suit. En ce qui concerne notre problème, ces tests consisteraient à examiner l'espace inter-ordre, l'épaisseur des traits, les coupures significatives. Vouloir réaliser ces tests de façon satisfaisante aboutirait à une recopie des algorithmes utilisés dans l'organe de description. C'est pourquoi nous pensons que le système le plus adapté à effectuer des tests sur l'image digitalisée est représenté par notre système de description lui-même, puisque ce sont les 8 phrases de description qui serviront à reconnaître le caractère au niveau du classifieur.

La recherche du bon seuil d'acquisition va donc se ramener à déterminer la meilleure des descriptions.

FIGURE 3.12. Variation du caractère digitalisé en fonction du seuil d'acquisition.



Cette technique a été rendue possible parce que :

- . la description que nous utilisons est de type structurel, de telle sorte que les primitives composant les phrases peuvent être facilement reliées à un élément de la structure du caractère. L'établissement d'un organe de choix de la meilleure description en sera facilité.
- . la technologie adoptée, à savoir deux microprocesseurs échangeant des informations, nous permet de réaliser une rétroaction de l'organe de traitement sur l'acquisition. De ce fait, en fonction des résultats obtenus au niveau des descriptions, l'organe de traitement pilotera le seuil d'acquisition.

Le système d'acquisition effectue une acquisition du caractère inconnu. A chaque matrice g' correspondant à un seuil ($0 \leq S_{acq} < 16$), nous déterminons sa description qui est alors stockée en mémoire. Pratiquement, cette opération de réduction du caractère débute avec un seuil d'acquisition de valeur égale à 0 (caractère épais). Le seuil est progressivement incrémenté jusqu'à la valeur maximale de 15, à moins que le caractère devenu trop squelettique ne se coupe, ce qui est détecté lors du traitement correspondant et provoque alors l'arrêt de cette phase, pour passer directement à la partie décision : choix de la "meilleure" description.

Afin de suivre le fil conducteur, ayant présidé à l'élaboration de l'organe de décision, comparons un instant notre système à un appareil photographique. Si, lors de la prise de vue, la mise au point n'est pas réalisée, l'image sera floue et empâtée ; le sujet photographié apparaîtra comme une tache aux contours incertains. Au fur et à mesure que l'on approche de la mise au point correcte, le sujet devient net, les arêtes vives, des détails apparaissent. Pour notre système de reconnaissance, il en sera de même ; plus nombreux seront les détails, plus vifs seront les segments, meilleure sera la description donc le seuil S_{acq} .

Les critères de décision qui permettent de sélectionner la description optimale sont classés par ordre décroissant d'importance. Parmi les 16 descriptions initiales, nous ne conserverons que celles ayant satisfait au critère n° 1, sommet de la hiérarchie. Cette opération se poursuit en éliminant à chaque étape une partie des descriptions rescapées de l'étape précédente. Nous distinguerons quatre critères :

1/ CRITERE DES SEGMENTS

Parmi toutes les descriptions, nous retenons celles qui possèdent le plus grand nombre de primitives "m" (ou segment de droite) dans les phrases d'ordre 1. On sélectionne ainsi les phrases qui possèdent le plus de détails possible.

2/ CRITERE PORTANT SUR LES ORDRES

On détecte ensuite, parmi les descriptions issues de l'étape précédente, celles qui possèdent le plus grand nombre de phrases d'ordre k non vides ($\varphi_k \neq z^a$). Le nombre de phrases non vides par description oscille entre 2 au minimum et 8 au maximum. Cette opération correspond à la détection du plus grand nombre d'"espaces suffisants" dans le caractère, ce qui théoriquement se réalise lorsque l'on atteint la limite entre le caractère aux segments le plus fin possible et le caractère z^a .

3/ CRITERE DES SEGMENTS D'ORDRE SUPERIEUR

Parmi les descriptions restantes, on retiendra celles qui possèdent le plus grand nombre de segments décrits avec la primitive m dans les ordres supérieurs à 1, c'est à dire 2, 3, 4.

4/ CRITERE MAJORITAIRE

Enfin, parmi les descriptions retenues à l'issue des trois étapes précédentes, nous comptabiliserons les descriptions identiques. Le groupe ainsi formé, dont le nombre sera le plus élevé, contiendra la description sélectionnée pour décrire le caractère. En cas de conflit, deux groupes possédant le même nombre de descriptions identiques, nous retiendrons la description dont le seuil d'acquisition est le plus faible.

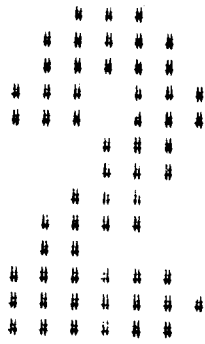
On pourra vérifier que cet algorithme de décision donne toujours une description unique. La figure 3.13 (page 42) détaille les différentes descriptions d'un 2 manuscrit en fonction du seuil d'acquisition. On indique également la description retenue, ainsi que le seuil correspondant.

Pour des caractères aux contours épais, Letraset par exemple, le seuil d'acquisition correspondant à la description sélectionnée prend des valeurs élevées (11, 12). Par contre, pour des caractères écrits avec une plume fine, le seuil retenu prend des valeurs faibles (0, 1, 2).

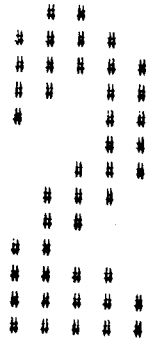
3.6. PROPOSITION D'UNE RECHERCHE RAPIDE DU SEUIL D'ACQUISITION

Si la méthode précédente de choix du seuil d'acquisition optimal, que nous avons adoptée pour la suite de notre travail, est très satisfaisante sur le plan de la description retenue, il n'en est pas de même au point de vue temps de calcul. En effet, une description optimale nécessite 16 acquisitions, 16 réductions et 16 traitements successifs, nous avons multiplié par 16 le temps de description d'un caractère.

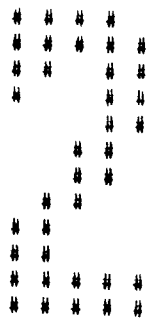
On peut certes concevoir la réalisation d'une machine, où 16 processeurs d'acquisition et 16 processeurs de traitement travailleraient en parallèle afin de déterminer, grâce à l'organe de décision, la description optimale, sans allonger le temps apparent du calcul. Mais on conçoit que cette solution d'architecture sophistiquée et onéreuse ne soit pas retenue pour des applications courantes. C'est pourquoi nous nous sommes appliqués à étudier une méthode de détermination du bon seuil S_{acq} , peut-être moins performante, mais plus rapide.



HORIZ. VERT.
fmfm fm
ff
SEUIL = 00



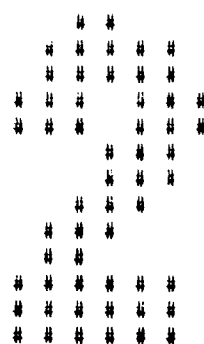
HORIZ. VERT.
fmfm fm
ff
SEUIL = 04



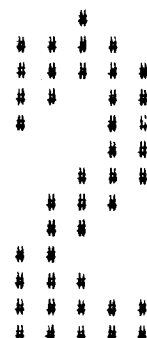
HORIZ. VERT.
mfm fm
ff
SEUIL = 08



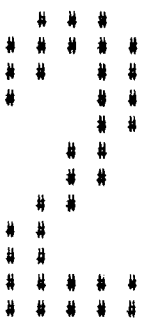
HORIZ. VERT.
mffm fm
f ff
SEUIL = 12



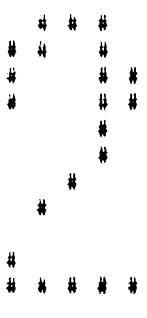
HORIZ. VERT.
fmfm fm
ff
SEUIL = 01



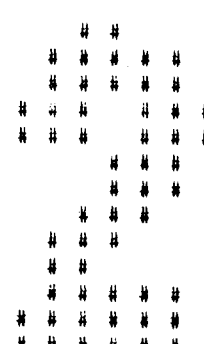
HORIZ. VERT.
fmfm fm
ff
SEUIL : 05



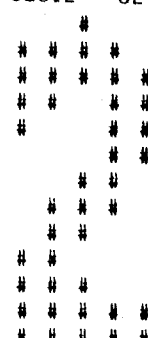
HORIZ. VERT.
mfm fm
ff
SEUIL = 09



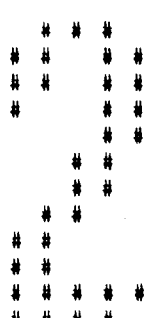
HORIZ. VERT.
mffm fm
f ff
SEUIL = 13



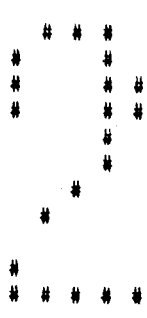
HORIZ. VERT.
fmfm fm
ff
SEUIL = 02



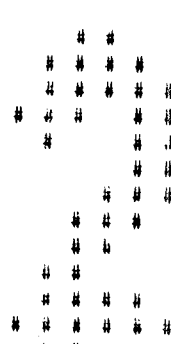
HORIZ. VERT.
mfm fm
ff
SEUIL = 06



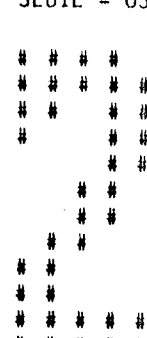
HORIZ. VERT.
mfm fm
ff
SEUIL = 10



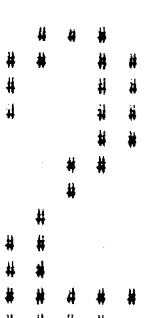
HORIZ. VERT.
mffm fm
f ff
SEUIL = 14



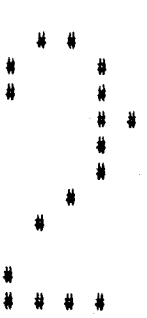
HORIZ. VERT.
fmfm fm
ff
SEUIL = 03



HORIZ. VERT.
mfm fm
ff
SEUIL = 07



HORIZ. VERT.
mffm fm
f ff
SEUIL = 11



HORIZ. VERT.
ffm fm
f ff
SEUIL = 15

FIGURE 13.
Description de la sélection...

HORIZ. VERT.
mffm fm
f ff
SEUIL = 13

3.6.1. DEFINITION DE LA DENSITE D'UN CARACTERE

Nous définissons la densité d'un caractère de la manière suivante :

$$\text{densité} = \frac{\text{nombre de points } \epsilon \text{ au caractère} \times 100}{\text{nombre total de points encadrant le caractère}}$$

- les points ϵ au caractère sont les points dont la valeur est "1" ;
- les points encadrant le caractère sont définis comme suit :
 - . lors d'un balayage ligne, le premier point est représenté par le premier point appartenant au caractère ; le dernier point est le dernier appartenant au caractère. Entre ces deux points, tous les pixels blancs ou noirs sont comptabilisés ;
 - . lors d'un balayage colonne, la définition est identiques à la précédente.

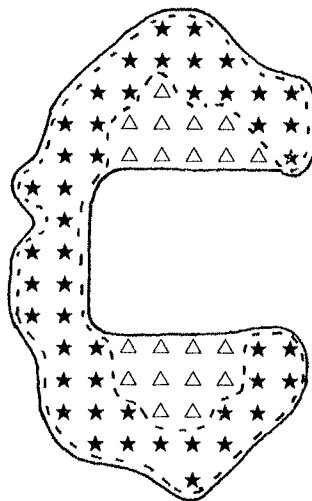
De ce fait, nous définissons une densité horizontale et une densité verticale que l'on notera DH et DV. :

$$\text{DH} = \frac{\text{nombre de points } \epsilon \text{ au caractère} \times 100}{\text{nombre total de points encadrant le caractère lors d'un balayage horizontal}}$$

$$\text{DV} = \frac{\text{nombre de points } \epsilon \text{ au caractère} \times 100}{\text{nombre total de points encadrant le caractère lors d'un balayage vertical}}$$

On remarquera que la densité mesure le rapport entre la taille du caractère et la taille de l'espace inter-ordre, ce qui constitue la notion la plus importante, au sens de notre traitement. Nous avons visualisé les points appartenant au caractère, \star , et les points encadrant le caractère, \star plus Δ , sur un caractère C représenté à la figure 3.14.

FIGURE 3.14. Calcul de la densité horizontale sur un caractère C.



— limite des points encadrant le caractère dans le sens horizontal
 - - - - points ϵ au caractère

Le calcul des densités de ce caractère nous donne :

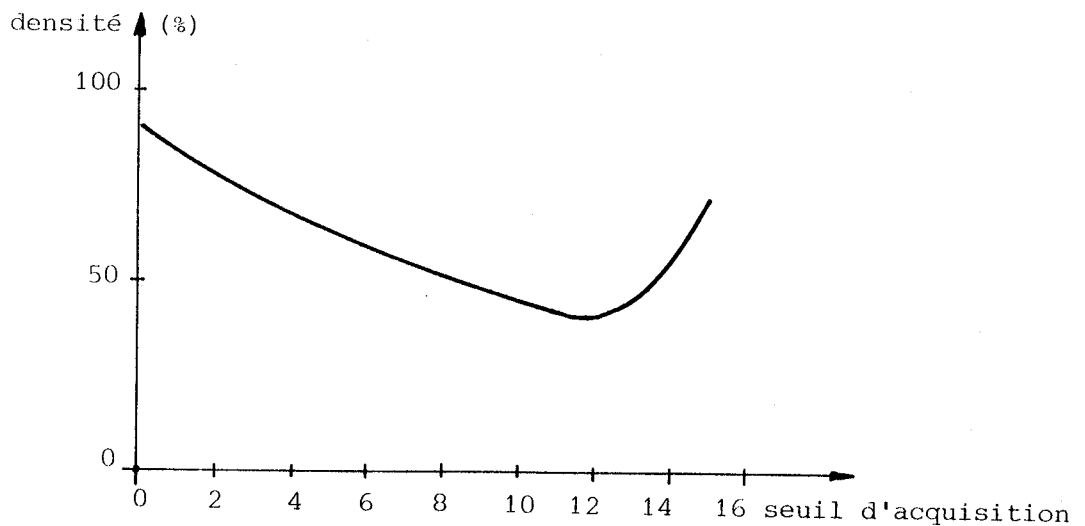
$$DH = \frac{47 \times 100}{67} = 70 \% \quad ; \quad DV = \frac{47 \times 100}{103} = 46 \% .$$

Un graphisme ayant de grands espaces inter-ordres aura une densité de faible valeur. A l'opposé un caractère ne possédant que des éléments d'ordre 1 aura une densité égale à 100 %.

3.6.2. VARIATION DE LA DENSITE EN FONCTION DU SEUIL D'ACQUISITION

Ayant implémenté l'algorithme de calcul de la densité, nous avons acquis un nombre important de caractères qui ont chacun été réduits avec les 16 seuils possibles. A chaque fois, nous avons calculé les densités horizontales et verticales. A l'aide de ces résultats, nous avons tracé l'allure de la courbe : densité = $f(S_{acq})$: figure 3.15.

FIGURE 3.15. Densité = $f(\text{seuil d'acquisition})$ en moyenne.



Cette courbe est translatable suivant l'axe des abscisses et des ordonnées suivant qu'il s'agit d'une densité horizontale ou verticale, suivant la qualité de l'impression, la fonte utilisée et le type du caractère. On constate par contre quasiment toujours l'existence d'un minimum que l'on explique en analysant la courbe décomposable en deux parties :

- partie à pente négative : lorsque le seuil d'acquisition augmente, le caractère se squelettise, de telle sorte que les espaces inter-ordres (s'il y en a) croissent ; la densité diminue.
- partie à pente positive : le caractère se squelettisant davantage, les boucles finissent par se couper par manque de points, si bien que la densité du caractère augmente.

3.6.3. CORRELATION ENTRE LE CHOIX DU SEUIL D'ACQUISITION ET LA DENSITE CALCULEE

Soient :

S_{acq} : le seuil sélectionné par l'algorithme de choix du seuil (§ 3.5),

S_{DH} : le seuil correspondant au minimum de la densité horizontale,

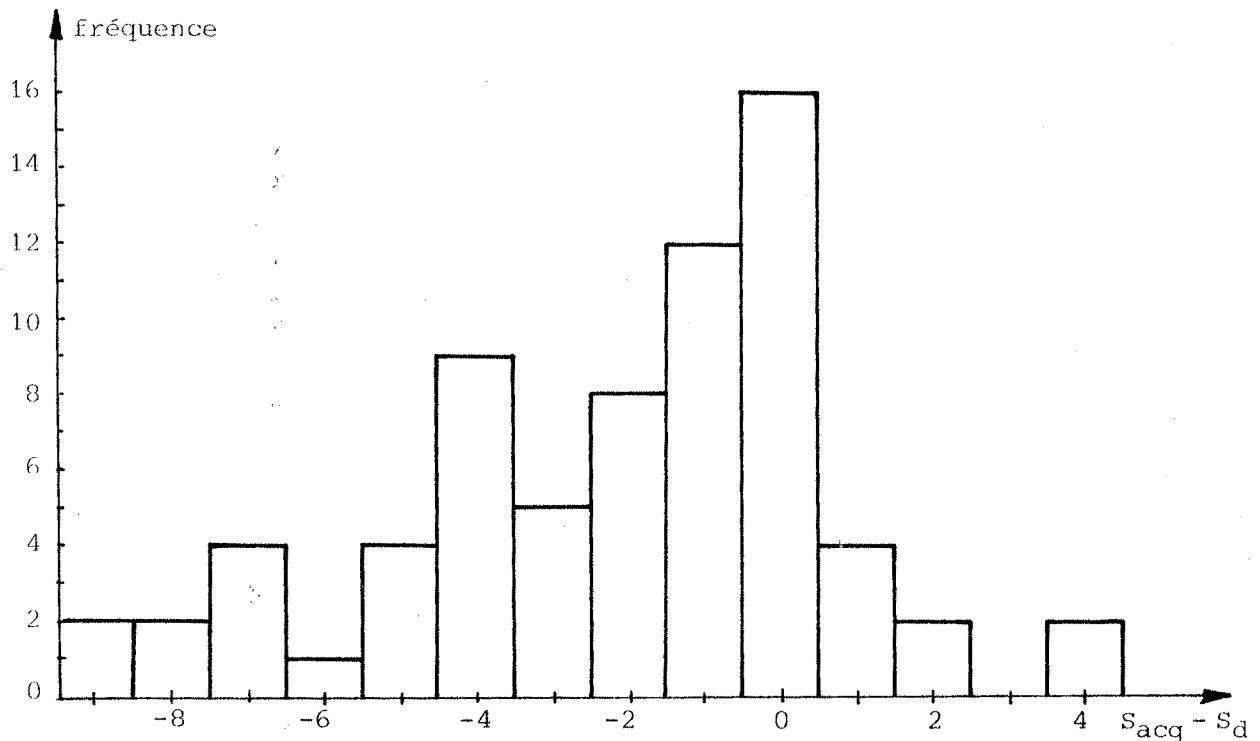
S_{HV} : le seuil correspondant au minimum de la densité verticale.

$$S_d = \min(S_{DH}, S_{DV})$$

S_d correspond à l'apparition de la première coupure du caractère.

Nous avons tracé l'histogramme de $S_{acq} - S_d$ pour des caractères "Letraset" (épais), des caractères dactylographiés et des caractères manuscrits (caractères plutôt fins, figure 3.16).

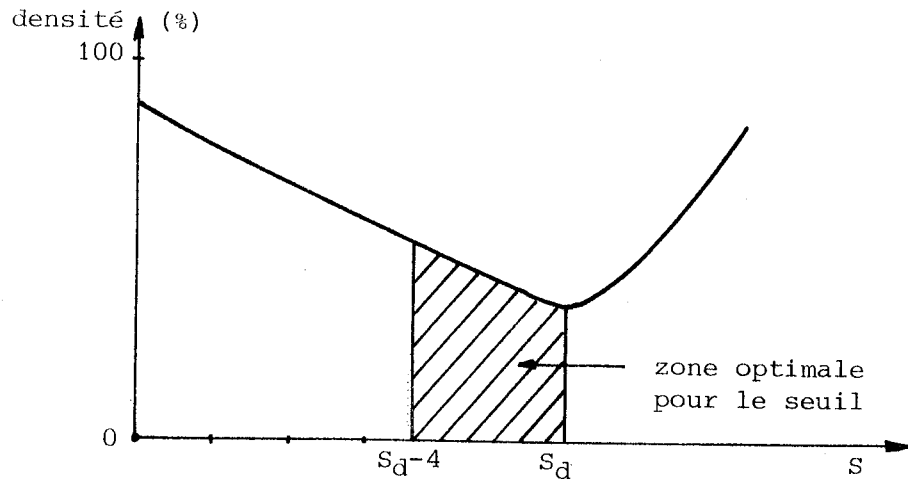
FIGURE 3.16. Histogramme de $S_{acq} - S_d$.



On constate que la majorité des seuils S_{acq} sont inférieurs au seuil S_d (dans 89 % des cas), c'est à dire que le seuil optimal se situe juste avant l'apparition de la première coupure.

On note également que 70 % des cas sont concentrés dans l'espace $S_{acq} = S_d$ et $S_{acq} = S_d - 4$. De ce fait, pour des seuils appartenant à la zone optimale $[S_d - 4, S_d]$, la description sera, dans la majorité des cas, satisfaisante au sens de notre traitement et des critères de choix (figure 3.17).

FIGURE 3.17. Détermination de la zone optimale pour le seuil.



Il est donc possible de réaliser un algorithme rapide de détermination du bon seuil. Pour ce faire, sur chaque matrice g' correspondant à un seuil d'acquisition, on calculera les densités horizontales et verticales. En débutant l'opération avec le seuil d'acquisition nul, on s'attachera à déterminer le premier minimum de densité, d'où connaissance du seuil S_d . On peut alors choisir par exemple le seuil $S_d - 2$, centre de la zone optimale et livrer la matrice g' correspondante à l'organe de traitement. On notera que, dans cette zone optimale, les descriptions correspondantes ne diffèrent pas les unes des autres dans de grandes proportions.

Cet algorithme de calcul de la densité et de détection du minimum est aisément implémentable sur l'organe d'acquisition qui, indépendamment de l'organe de traitement, déterminera le seuil adéquat. Le calcul des densités étant rapide, et pouvant être effectué pendant le traitement du caractère précédent, on conçoit que cette détermination du seuil, de qualité satisfaisante, ne pénalise pas l'ensemble de la reconnaissance.

Cependant, comme nous l'avons déjà signalé, nous avons conservé, pour notre étude, les critères de choix développés au paragraphe 3.5.

4 - LE CLASSIFIEUR

Après traitement, un caractère inconnu se présente donc sous forme d'une description composée de 8 phrases. On ne peut cependant espérer, compte tenu du mode de description utilisé, mais également des propriétés graphiques des caractères alphanumériques en particulier manuscrits, que cet octuplet soit : unique pour l'ensemble des caractères issus d'une même classe, différent pour tous les caractères issus de classes différentes. Par exemple :

- . en écriture manuscrite le caractère "quatre" s'écrit aussi bien :

4 , 4 , 4

Du point de vue de la description, ces graphismes se présentent avec des structures différentes. On demandera au classifieur de les associer dans une classe unique ;

- . en écriture manuscrite ou dactylographiée, les caractères :

D , O , Q

(par exemple) apparaîtront au contraire avec une description très voisine sinon identique. On demandera, dans ce cas, au classifieur de les séparer.

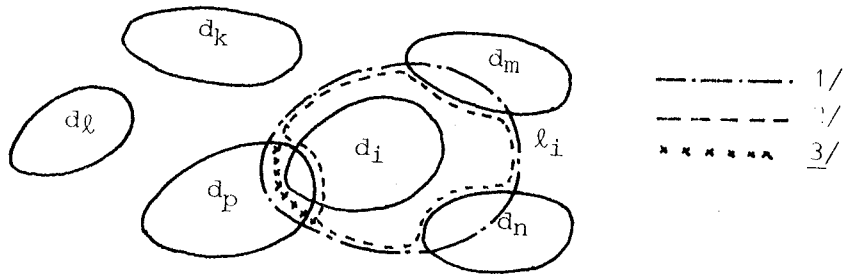
Le premier problème se résoudra au niveau de l'apprentissage par l'établissement d'un langage propre à chaque structure, le langage de la classe étant alors tout simplement l'union de chacun des langages élémentaires.

Le second problème, lui, nécessitera la mise en oeuvre d'une reconnaissance en deux étapes : on reconnaîtra, dans une première étape, au vu de la description, un groupement de classes ; la seconde étape permettra la séparation des classes, grâce à des tests supplémentaires adaptés au groupement considéré.

Au niveau de l'apprentissage, cette structure de décision à deux niveaux aura une conséquence intéressante : le langage d'une classe sera établi de façon à ce que, si possible, tous les éléments de cette classe soient reconnus, sans que l'on cherche à rendre ce langage absolument disjoint de celui des autres classes. Si une ambiguïté apparaît, due à des langages non disjoints, elle sera résolue par le deuxième niveau de décision. On se réserve donc la possibilité d'utiliser effectivement deux niveaux de décision pour l'ensemble des classes à reconnaître. Ces deux niveaux ne seront effectivement utilisés que si, en cours d'apprentissage, ils s'avèrent nécessaires.

La stratégie d'apprentissage que l'on a adoptée, pour le premier niveau, est la suivante (figure 4.1).

FIGURE 4.1.



Rappelons la signification des notations :

$\{d_i\}$: ensemble des descriptions du caractère i vues pendant l'apprentissage,
 l_i : langage de la classe C_i .

Le but de l'apprentissage consiste à découvrir l_i tel que :

$\{d_i\} \subset l_i$
 $\{d_j\} \not\subset l_i \quad \forall j \neq i$.

1/ On définit l_i tel que $\{d_i\} \subset l_i$.

2/ On teste les $\{d_j\}$ par rapport à l_i ($j = k, l, m, n, \dots$) :

si $\exists d_j \in l_i \Rightarrow$ on restreint l_i ,

si $\{d_j\} \not\subset l_i \Rightarrow$ on ne fait rien.

3/ Si $\exists d_i \notin l_i \Rightarrow$ on agrandit l_i tel que $\nexists d_i \notin l_i$.

En reprenant l'exemple de la figure 4.1, on constatera que les classes C_i et C_p ne sont pas séparables à ce niveau, et feront l'objet de tests spécifiques pris en compte par le second niveau.

4.1. DESCRIPTION INDUITE

Pour élaborer un classifieur, il faut inférer une grammaire à partir des descriptions fournies par l'organe de traitement. Ces descriptions sont représentées pour nous, par l'octuplet : $d_i = \{ h_1, h_2, h_3, h_4, v_1, v_2, v_3, v_4 \}$.

Chaque phrase φ_k composée de 8 primitives (m, f, z) contient l'information sur la présence et le mode d'assemblage des primitives d'un élément d'ordre k. L'examen des phrases h_k (ou v_k) permet également d'extraire une information sur le mode d'association des 4 éléments d'ordre k ; enfin, l'ensemble des phrases h_k n'est pas indépendant de l'ensemble v_k .

S'il nous est possible et même aisé d'inférer 8 grammaires à partir de chacune des phrases φ_k considérées isolément, ceci aurait conduit à une perte d'information trop importante. Quant à inférer une grammaire génératrice de 8 phrases non indépendantes, nous y avons renoncé. Il nous a paru préférable d'extraire de d_i une information sur la structure globale du caractère sous forme d'une description induite que nous explicitons.

4.1.1. DESCRIPTION DE LA STRUCTURE GLOBALE DU CARACTERE

Un certain nombre de propriétés du graphisme peuvent être dégagées à partir d'opérations élémentaires réalisées sur d_i .

a/ COMPTAGE DES ELEMENTS D'ORDRE k

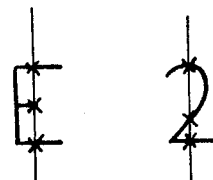
On compte le nombre de phrases non vides h_k et v_k de la description, c'est à dire différente de z^8 .

Ceci revient, pour les phrases v_k par exemple, à calculer le nombre maximal d'intersections avec une droite verticale (figure 4.2). Le caractère E et le caractère 2 possèdent ainsi trois intersections avec un axe vertical.

Nous utiliserons les notations :
OH pour les éléments horizontaux,
OV pour les éléments verticaux.

FIGURE 4.2.

Les caractères alphanumériques latins possèdent au minimum une phrase h_k et v_k non vide et, au plus, quatre phrases par sens d'analyse. Cependant, on notera que cette information est insuffisante pour séparer tous les caractères.



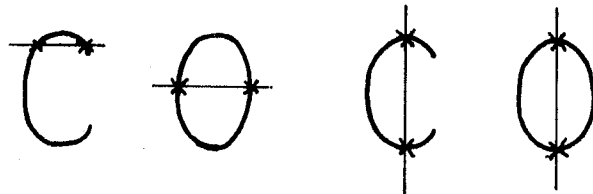
b/ COMPTAGE DU NOMBRE DE SEGMENTS HORIZONTAUX ET VERTICAUX DANS LES PHRASES φ_1

Effectuer ce comptage revient à comptabiliser le nombre de primitives "m" dans les phrases φ_1 . En ce qui concerne le caractère E de la figure 4.3, cette notion peut sembler redondante avec la précédente ; en effet, à 3 0 correspondent trois segments horizontaux. Par contre, il n'en est pas de même en ce qui concerne le caractère 2 dont l'oblique ne sera pas détectée comme un segment horizontal (le caractère 2 possèdera au plus 2 segments horizontaux).

Les segments horizontaux seront notés SH, les verticaux : SV. Les caractères alphanumériques possèdent au plus 4 segments horizontaux et, au plus, 4 verticaux.

L'information accumulée est encore insuffisante car certains caractères ne sont pas discernables. Ainsi, les caractères C et O de la figure 4.3 ont tous deux le même nombre d'éléments et le même nombre de segments. Nous effectuons un nouveau test.

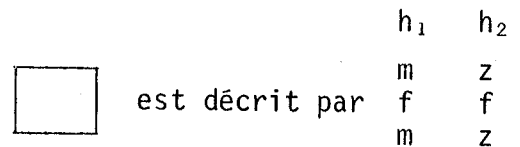
FIGURE 4.3.



c/ COMPTAGE DU NOMBRE DE CONTOURS FERMES

Préalablement, nous effectuons l'opération consistant à reconnaître un contour fermé. Il nécessite la présence du groupement "mfm" dans φ_1 et du groupement ".f." correspondant dans φ_2 (figure 4.4).

FIGURE 4.4.



Les notations adoptées sont :
CH pour un contour fermé vu dans un sens horizontal,
CV pour un contour fermé vu dans un sens vertical.

Dans un caractère, on ne dénombre que 0, 1, ou 2 contours dans un sens d'analyse. En reprenant l'exemple de la figure 4.3, le caractère C n'a pas de contour horizontal tandis que le caractère O aura un contour noté CH. On remarquera, par exemple, que les caractères B, 8, peuvent se séparer des autres caractères grâce à ce seul comptage. En effet, ce sont les seuls à posséder deux contours fermés horizontaux. Ce comptage se ramène à l'étude des boucles fermées du caractère.

A ce stade, la description induite se présente sous la forme :

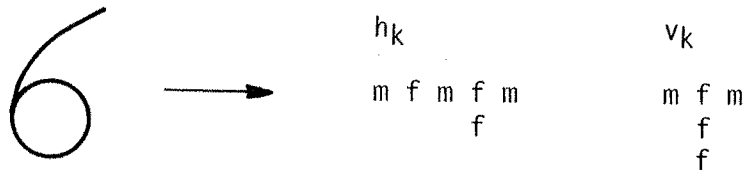
$$S = OH^a OV^b SH^c SV^d CH^e CV^f$$

où a, b, c, d, e, f représentent le nombre de nouvelles primitives OH, OV, SH, SV, CH, CV précédemment définies.

La description induite du caractère 6 par exemple (figure 4.5) sera :

$$S_6 = OH^2 OV^3 SH^3 SV^2 CH^1 CV^1$$

FIGURE 4.5.
Description du 6.



Cette description induite est produite par une grammaire réursive et régulière g_i :

$$g_i = (V_T, V_N, P, S) ,$$

V_T : ensemble des primitives = { OH, OV, SH, SV, CH, CV, N } = { p_j } ;

V_N : ensemble des non terminaux : { $P_1, P_2, P_3, P_4, P_5, P_6$ } ;

P : règles de production ;

S : phrase correcte sur le plan syntaxique.

Soit :
$$S \rightarrow P_1 \cdot P_2 \cdot P_3 \cdot P_4 \cdot P_5 \cdot P_6$$

les règles de production :

$$P_j \xrightarrow{(n_j)} P_j \cdot P_j$$

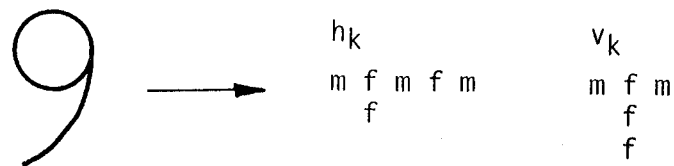
$$P_j \rightarrow N$$

n_j représentant le nombre d'occurrence de la règle et N la primitive nulle.

De plus, cette grammaire a le caractère particulier suivant : les terminaux entrant dans la production d'un non-terminal sont spécifiques à celui-ci. On conçoit que l'inférence d'une telle grammaire soit particulièrement aisée, que son utilisation dans une analyse syntaxique, par exemple bas-haut, le soit également.

Toutefois, on notera que les phrases S sont insuffisantes pour discerner certains caractères ; par exemple, 6 9 ; L T ; P A ; ... ne sont pas encore séparables. Le caractère 9 dont la description est précisée à la figure 4.6 donnera la description induite : $S_9 = OH^2 OV^3 SH^3 SV^2 CH^1 CV^1$ identique à S_6 . Dans notre exemple, seule la position de la boucle fermée permet de discriminer ces deux caractères, c'est pourquoi nous associerons à la description induite une information structurelle complémentaire apportée par des fonctions descriptives

FIGURE 4.6. Description du 9.



4.1.2. FONCTIONS DESCRIPTIVES

Dans la description induite, le caractère inconnu, en plus du comptage, sera décrit dans cette seconde phase, par une partie des phrases h_k et/ou v_k afin d'examiner un détail particulier.

Dans cette seconde phase, on utilisera les mêmes primitives m, f, z que précédemment, auxquelles s'ajoutera la primitive "indifférente" représentée par l'espace " ", ou éventuellement le point en cas d'ambiguïté (*); cette primitive " " représente indifféremment m, f ou z . Toutes les associations autorisées par les règles établies au cours du chapitre traitement sont permises.

Afin de déterminer la partie des phrases h_k et/ou v_k qui seront étudiées, nous avons créé trois fonctions descriptives permettant de localiser cette partie :

- . fonction début : DH ou DV
l'association de primitives doit constituer le début de la description horizontale ou verticale,
- . fonction fin : FH ou FV
dans ce cas, l'association doit se situer à la fin de la description horizontale ou verticale.

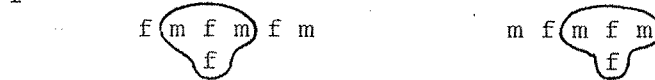
En reprenant l'exemple des caractères 6 et 9, on demandera pour le caractère 6 de posséder la boucle fermée dans le bas du caractère, c'est à dire à la fin de sa description horizontale. On prendra la fonction : FH($m f m$).
f

(*) On notera que l'espace " " a déjà été utilisé dans l'écriture de la description pour en simplifier la présentation. Dans cette description, l'espace représentant la primitive z . Cependant, toute confusion est impossible car dans la description il n'existe pas de primitives "indifférentes". Dans les fonctions descriptives, la primitive z sera effective-

A l'opposé, pour le 9, la boucle fermée étant en haut du caractère, elle devra constituer le début de la description horizontale, à savoir : $DH(m f m)$, ce qui se retrouvera dans les descriptions h_k suivantes :



Mais cette fonction $DH(m f m)$ ne sera pas présente dans les exemples suivants :



On trouve bien l'association $(m f m)$, mais pas au début de la description.

. fonction noyau : NH ou NV

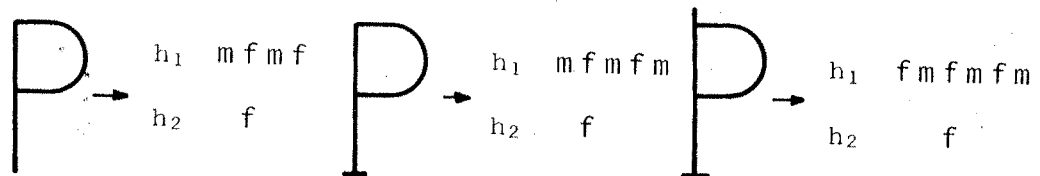
l'association de primitives doit se trouver dans h_k ou v_k sans précision quant à sa position ; seule, la présence de l'association importe.

Si l'on désire séparer les caractères A et P, on pourra utiliser la fonction NH :

pour le A : $NH(m f m f)$; pour le P : $NH(m f m f)$;
 $f f$; $f z$

ce qui permet de distinguer ces deux caractères, tout en permettant l'existence d'éventuels segments supplémentaires (figure 4.7).

FIGURE 4.7.



On pourra noter :

.l'équivalence de certaines fonctions : $FH(m f)$ \equiv $NH(m f z)$;
 f f

.le fait que les fonctions NH ou NV autorisent plus de libertés que les fonctions DH, DV ou FH, FV. Par exemple :

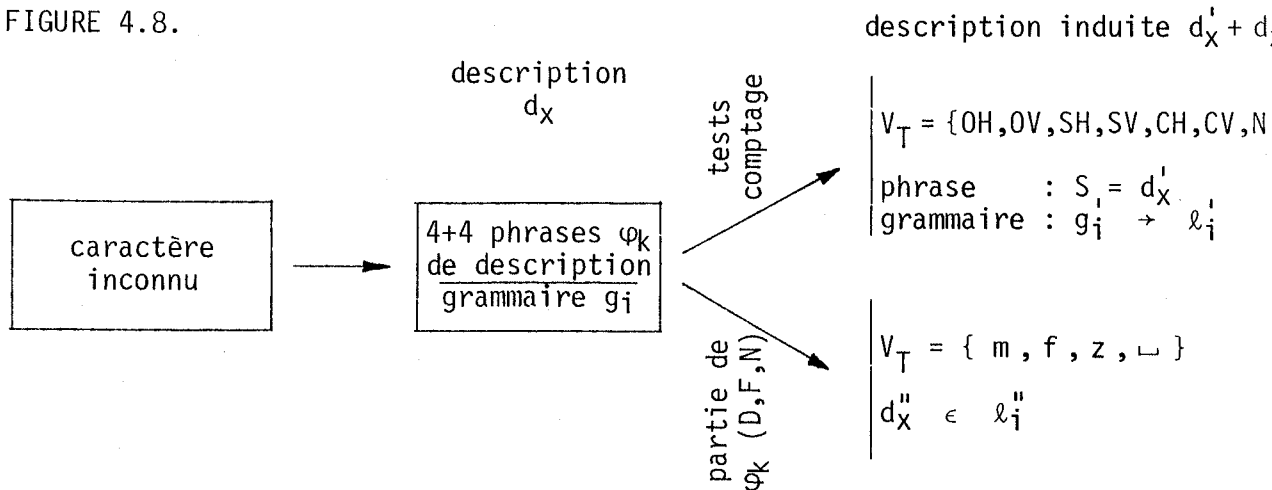
$FV(f m)$ vérifiée \Rightarrow $NV(f m)$ vérifiée
 f f

$NH(m f)$ vérifiée $\not\Rightarrow$ $DH(m f)$ vérifiée .
 f f

En résumé, ne pouvant atteindre le langage ℓ_i généré par la grammaire g_i , des phrases φ_k de la description, nous avons donc créé une description induite à partir des phrases φ_k (figure 4.8). Cette description induite se sépare en deux parties :

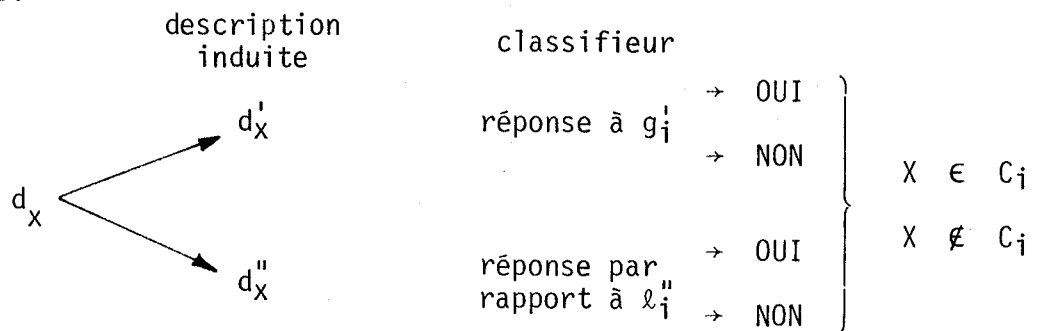
- ☆ des tests et des comptages permettent de décrire le caractère avec les primitives OH, OV, SH, SV, CH, CV contenant des informations sur les quatre ordres, sur une phrase, sur les boucles fermées. Le caractère se décrit partiellement avec une phrase d'_x . La grammaire g'_i génère ainsi un langage ℓ'_i ;
- ☆ une partie de la description de φ_k (notée d''_x) sélectionnée grâce aux fonctions descriptives, D, F ou N, constitue le langage ℓ''_i .

FIGURE 4.8.



La grammaire g'_i et le langage ℓ''_i n'ont de commun que leurs indices relatifs à la classe C_i . Au niveau du classifieur, l'interaction des réponses vis à vis de la grammaire g'_i et du langage ℓ''_i permettra de déterminer si le caractère inconnu appartient ou non à C_i (figure 4.9).

FIGURE 4.9.



4.2. STRATEGIE D'APPRENTISSAGE ET RESULTATS

La mise en oeuvre de notre système va s'opérer en trois étapes. Nous procéderons tout d'abord à l'élaboration d'un dictionnaire de description.

A l'aide de l'organe d'acquisition, de traitement et d'un classifieur lexical en mode d'apprentissage, nous avons rangé en mémoire un nombre important de descriptions $d_i^n = \{\varphi_k\}$ (figure 4.10).

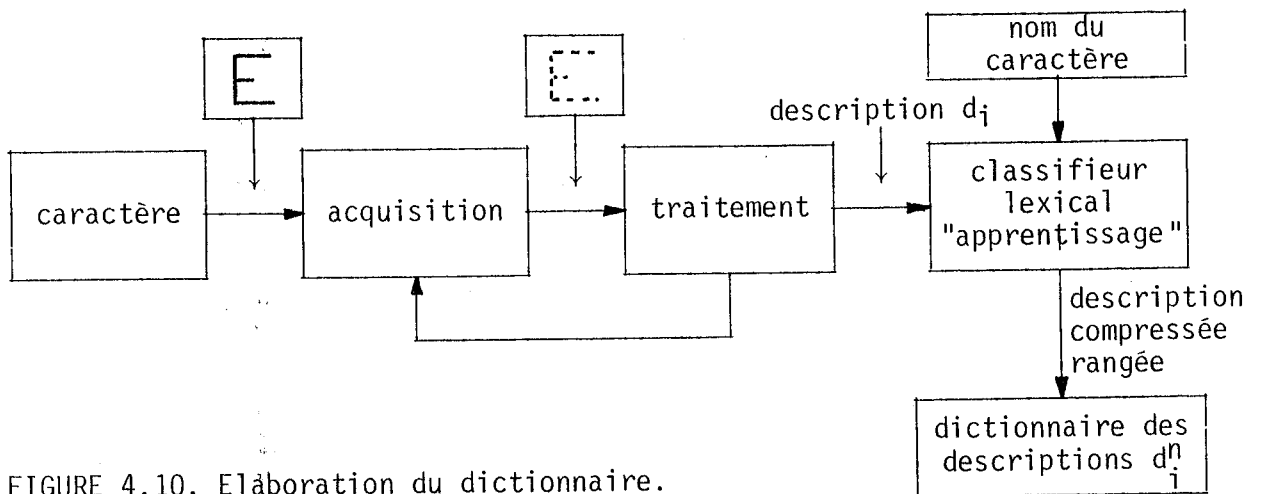
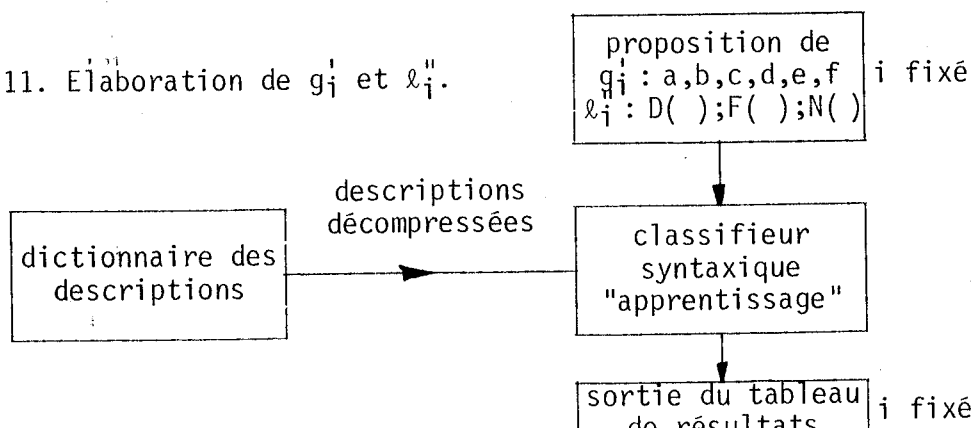


FIGURE 4.10. Elaboration du dictionnaire.

Un classifieur lexical est chargé de ranger les descriptions par classes, grâce au nom du caractère, fourni par l'opérateur. Ce classifieur ne rangera au sein d'une même classe que des descriptions différentes stockées sous forme compressée afin de gagner de la place et de pouvoir ranger ainsi un grand nombre d'informations. L'étape suivante consistera à rechercher la grammaire g_i et le langage ℓ_i .

Dans cette étape, l'opérateur, au vu des descriptions du dictionnaire, propose des valeurs pour les variables a, b, c, d, e, f, et des fonctions descriptives pour chaque classe C_i (figure 4.11).

FIGURE 4.11. Elaboration de g_i et ℓ_i .



Après proposition de g_i^1 et l_i^2 , le classifieur syntaxique en mode apprentissage va fournir la réponse (oui - non) de chaque caractère A_j^n soumis à g_i^1 et l_i^2 . La proposition sera satisfaisante si :

- . les A_j^n pour $j = i$ sont grammaticalement corrects au sens de g_i^1 et appartiennent à l_i^2 ;
- . les A_j^n pour $j \neq i$ ne sont pas grammaticalement corrects au sens de g_i^1 et n'appartiennent pas à l_i^2 .

Dans le cas contraire, g_i^1 et/ou l_i^2 seront modifiés en les restreignant si trop de caractères A_j^n ($j \neq i$) sont sélectionnés, en les augmentant si les A_j^n ($j = i$) sont peu nombreux.

Nous noterons également quatre avantages d'un tel système :

- . la grammaire g_i^1 et le langage l_i^2 proches du graphisme s'établissent facilement ;
- . la recherche des langages l_i^1 et l_i^2 ne peut avoir de répercussion sur les langages l_j^1 et l_j^2 ($i \neq j$) ;
- . tout au long de l'apprentissage, nous ne modifions pas la structure du classifieur ; il suffit d'agir sur les nombres a, b, c, d, e, f et fonctions D, F, N ;
- . après proposition de g_i^1 et l_i^2 , la scrutation de tous les caractères A_j^n est rapide (temps d'impression des résultats).

Les mêmes éléments utilisés de manière différente serviront à réaliser le système de reconnaissance. Nous conserverons en plus les tables des g_i^1 et des l_i^2 pour toutes les classes C_i , qui constitueront les données du classifieur syntaxique en mode reconnaissance (figure 4.12).

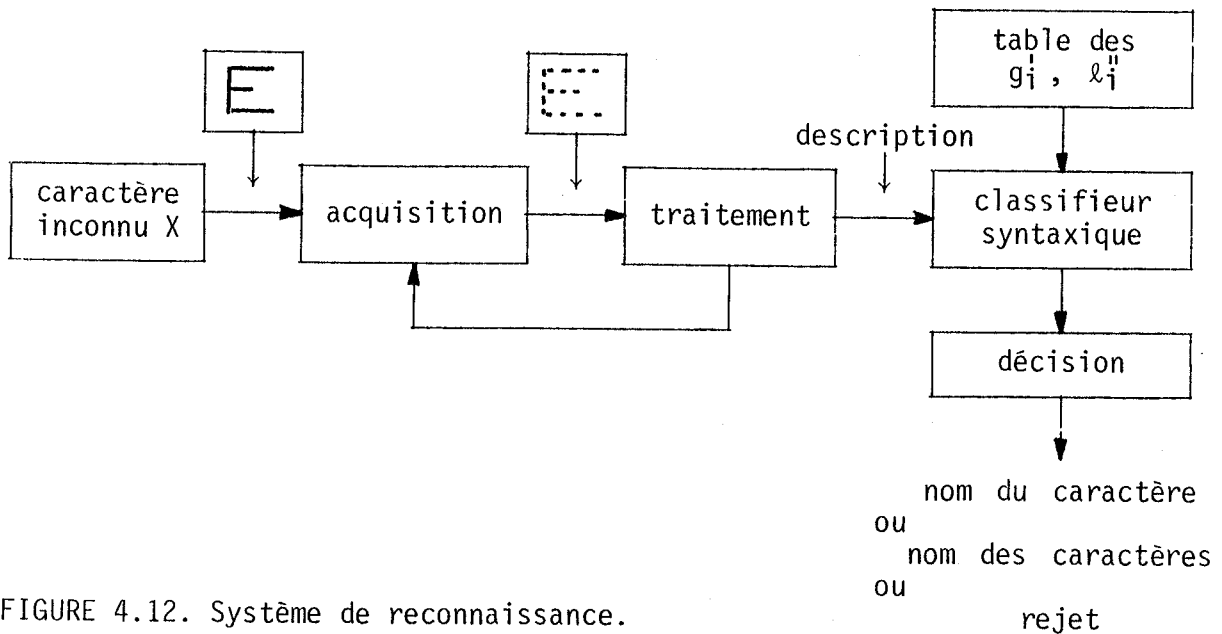


FIGURE 4.12. Système de reconnaissance.

Un caractère inconnu X est acquis, traité et présenté au classifieur sous forme de sa description d_x . Le classifieur va alors présenter d_x successivement à toutes les grammaires g_i' et à tous les langages l_i'' . Trois cas sont alors possibles ; l'organe de décision fournit une réponse :

- . d_x est grammaticalement correcte vis à vis de g_i' et ϵ à l_i'' pour i unique :
alors $X \in C_i$;
- . d_x est grammaticalement correcte vis à vis de g_i' et ϵ à l_i'' , g_j' et ϵ à l_j'' :
alors il y a ambiguïté : $X \in C_i$ ou $X \in C_j$;
- . d_x n'est grammaticalement correcte vis à vis d'aucune grammaire ou n'appartient à aucun langage :
alors il y a rejet.

Pour valider ces hypothèses, nous présenterons les résultats obtenus avec un sous-ensemble représentatif des caractères alphanumériques : les caractères hexadécimaux (A,B,C,D,E,F,0,1,2,3,4,5,6,7,8,9). Ils présentent l'avantage de regrouper quasiment tous les cas rencontrés en reconnaissance de caractères alphanumériques :

- doublets classiques : B, 8 et A, 4
- caractères simples : C, F, 1
- caractères en arc de cercle : principalement les chiffres.

4.2.1. LE DICTIONNAIRE

Le dictionnaire devant contenir un grand nombre de descriptions, nous avons compressé l'écriture de cette information afin de rendre compatibles nos ambitions et la taille réduite de notre système. La compression a été réalisée en tenant compte de certaines propriétés de la description :

- ☆ Une phrase d'ordre 1 ne peut contenir de primitives z , hormis la fin non significative de la description. On ne codera que la partie utile en utilisant un code binaire (m ou f). Un paramètre, cadrage, précise la taille de cette partie utile.
- ☆ Les phrases d'ordre supérieur à 1 possèdent le même nombre de primitives utiles que l'ordre 1, donc le même cadrage.
- ☆ Dans de nombreux cas, les ordres supérieurs ne contiennent que les primitives f ou z , que nous codons en binaire. Dans le cas contraire, présence de la primitive m , le code sera ternaire.
- ☆ Dans une description, le nombre d'ordre contenant une information non nulle est souvent égal ou inférieur à 4.

La description compressée aura la structure suivante :

- 1° octet : code horizontal (nombre d'ordre, code binaire ou ternaire, cadrage)
- 2° octet : code vertical (nombre d'ordre, code binaire ou ternaire, cadrage)
- 3° octet : information
- ⋮
- ⋮

La taille de l'information non compressée est de 16 mots de 8 bits, tandis que l'information compressée occupera au minimum 3 mots et, dans le pire des cas, 16 mots. Ne connaissant pas, a priori, l'importance relative des différentes classes, les descriptions ont été rangées sous forme série, par chaînage d'une classe à l'autre.

Nous avons demandé à 14 scripteurs d'écrire lisiblement en caractères majuscules d'imprimerie un ou plusieurs alphabets complets sur une feuille blanche. Au total, nous avons traité 20 échantillons de chacun des caractères hexadécimaux, auxquels nous avons ajouté un alphabet dactylographié ainsi qu'un alphabet "Letraset" représentant les caractères de très bonne qualité. Ces deux derniers alphabets ont été acquis intégralement. La figure 4.13 (page 59) représente des exemples de caractères acquis. On pourra apprécier la grande variété des graphismes : de la rigueur du "Letraset", les caractères évoluent en abordant les différentes variétés de style manuscrit.

Pour édifier le dictionnaire, nous avons acquis 380 caractères qui ont fourni 237 descriptions différentes rangées dans une mémoire comprenant 1064 octets ; en moyenne, une description compressée utilise 4,5 octets soit 36 bits/caractère. Une description non compressée utilisant 128 bits/caractère, nous avons donc réduit l'écriture de cette information dans un rapport 3,5. En détaillant davantage ces résultats, on constate que les caractères B et 8 utilisent en moyenne la place la plus importante : 5,3 octets/caractère soit 43 bits/caractère. A l'opposé, le caractère 1 par exemple ne nécessitera que 3 octets soit 24 bits/caractère.

L'élaboration du dictionnaire terminée, nous avons visualisé les différentes descriptions d'une même classe grâce à un algorithme de décompression qui permet de restituer les descriptions sous leur forme usuelle. Nous avons reproduit à la figure 4.14 (page 60) une partie de cette visualisation utilisée par la suite pour élaborer g_i et l_i . Outre la visualisation des descriptions, ce tableau nous indique les classes dans lesquelles se trouvent des descriptions identiques : par exemple, le deuxième A possède la même description qu'un R ; les B n° 2, 3, sont identiques à certains 8. La visualisation de ce dictionnaire a été menée de telle manière que les deux premières descriptions de chaque classe représentent les descriptions des caractères d'excellente qualité : "Letraset".

Nous avons constaté également que le traitement avalait une quantité non négligeable de bruits, de défauts et variété de graphisme puisque, vers la fin de l'élaboration du dictionnaire, en moyenne un graphisme sur trois possédait déjà sa descriptions rangée en mémoire dans la classe correspondante.

FIGURE 4.13. Caractères sources.

A B C D E F G H I J K L M N O P Q

R S T U V W X Y Z

1 2 3 4 5 6 7 8 9 0

A B C D E F G H I J K L M N O

P Q R S T U V W X Y Z

1 2 3 4 5 6 7 8 9

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

0 1 2 3 4 5 6 7 8 9

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

0 1 2 3 4 5 6 7 8 9

A B C D E F G H I J K L M N

O P Q R S T U V W X Y Z

1 2 3 4 5 6 7 8 9

Certains caractères sont favorisés, par exemple en reprenant l'exemple de la figure 4.14 (page 60), le dictionnaire comprend 19 caractères A tandis qu'il n'y a que 14 B pour un nombre équivalent d'acquisition. Cette différence provient de la complexité plus importante du caractère A, de la variété du graphisme plus étendue pour le A et du fait que le traitement privilégié, entre autres, les segments verticaux peu nombreux dans le A.

Enfin, à titre d'essai, nous avons acquis une seconde fois certains alphabets manuscrits. Le taux d'apparition de nouvelles descriptions est très faible puisqu'il est de l'ordre de 1/20 dépendant essentiellement des conditions d'acquisition : position du zoom de la caméra, inclinaison du caractère, niveau d'éclairage.

4.2.2. RECHERCHE DE g_i^1 ET g_i^2

La recherche d'une grammaire g_i^1 va consister à déterminer les valeurs des variables a, b, c, d, e, f. On rappelle qu'un caractère possède une description d_x qui donne une description induite d_x^1 (et d_x^2) :

$$d_x^1 = OH^a OV^b SH^c SV^d CH^e CV^f .$$

Or, une variable ne prendra pas une valeur unique comme le montre l'exemple du caractère 9 de la figure 4.6 (page 52). On avait :

$$d_9^1 = OH^2 OV^3 SH^3 SV^2 CH^1 CV^1 .$$

Après examen du dictionnaire d'apprentissage, on conçoit qu'un caractère tel que "9" ne possède pas de barre horizontale inférieure. Ceci va se traduire entre autres par la présence de seulement deux segments horizontaux et deux ordres verticaux. Dans ce cas, g^1 sera telle que :

$$g_9^1 \rightarrow L(g_9^1) = OH^2 OV^{2,3} SH^{2,3} SV^2 CH^1 CV^1 .$$

Le problème consiste donc à déterminer le domaine de variation possible des variables (figure 4.15.a, page 62), sachant qu'en tout état de cause :

$$a, b \in [1, 4] ; \quad c, d \in [0, 4] ; \quad e, f \in [0, 2] .$$

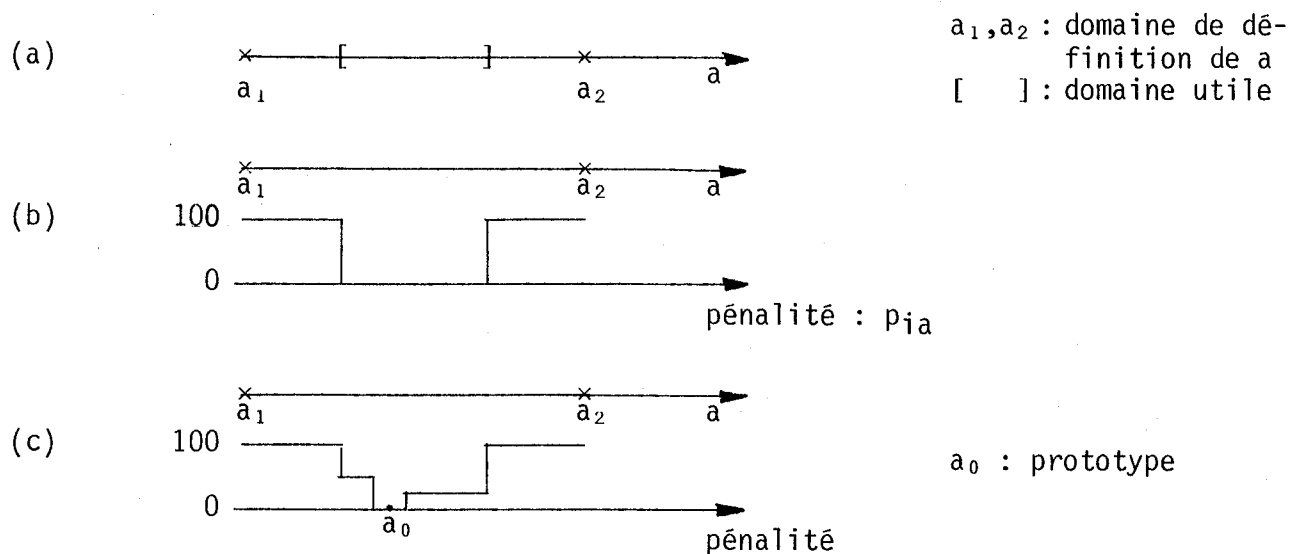
Si une variable prend toutes les valeurs possibles de son intervalle de définition, le langage correspondant sera indépendant de cette primitive qui n'apporte donc, dans ce cas, aucune information utile pour la reconnaissance du caractère.

En cours d'apprentissage, nous avons pu constater qu'une valeur particulière de chacune de ces variables apparaissait avec une fréquence plus élevée. Nous considérerons cette valeur particulière comme valeur prototype notée : a_0, b_0, \dots . Pour la suite, nous serons amenés à considérer non pas la valeur prise par les variables a, b, ... mais leur écart par rapport au prototype (a_0, a_0+1, a_0-1 , et autres).

Plutôt que de considérer les écarts permis autour du prototype, on peut admettre comme possibles tous les écarts à conditions de les affecter d'un coefficient de pénalité $p \in [0,100]$ (figure 4.15.b). Lorsque $p = 0$, l'écart est permis ; lorsque $p = 100$, l'écart est interdit : une valeur intermédiaire dénoterait un écart toléré mais permettrait de chiffrer la similitude avec le prototype (figure 4.15.c). Cette procédure permettrait alors d'introduire dans la grammaire g_i' des règles de production stochastiques en liant la valeur de la pénalité p à la fréquence d'utilisation de telle règle de production. Cependant, pour reconnaître les caractères hexadécimaux, nous nous sommes bornés à utiliser des pénalités à deux valeurs (0,100) ; notre échantillon d'apprentissage est de taille trop réduite pour effectuer des statistiques valables, et l'expérience nous a, par ailleurs, montré que les pénalités à deux valeurs étaient suffisantes.

En ce qui concerne les langages ℓ_i'' , chaque fonction descriptive utilisée est également affectée d'une pénalité $\in [0,100]$.

FIGURE 4.15.



Au début d'une reconnaissance, on attribue à chaque classe C_i un capital c_i de valeur 100. On présente alors successivement la description inconnue à chaque grammaire g_i' et langage ℓ_i'' . Le programme va soustraire du capital c_i les pénalités correspondantes. Si, au cours d'une telle analyse, c_i devient nul ou négatif, les opérations s'arrêtent et le système étudie la classe suivante C_{i+1} .

Lorsque toutes les classes sont traitées, l'organe de décision sélectionne la ou les classes d'appartenance du caractère inconnu X en fonction du capital c_i de chaque classe. On range alors les n capitaux obtenus par valeurs croissantes :

$$0 \leq \dots \leq c_k \leq c_j \leq c_i \leq 100 \quad .$$

1/ Si $c_i \geq \text{seuil } S_r$:

1.1/ si $c_i - c_j > \text{seuil } S'_r$,

alors : $x \in C_i$; le caractère est reconnu (correctement ou pas).

1.2/ si $0 \leq c_i - c_j \leq S'_r$,

alors : $\begin{matrix} x \in C_i \\ x \in C_j \end{matrix}$; il y a hésitation.

2/ Si $c_i < S_r$: il y a rejet.

Suivant le degré de confiance que l'on désire obtenir dans la réponse, il est possible d'adapter les seuils S_r et S'_r .

Notre système actuel n'utilisant que des pénalités à deux valeurs (0,100), les capitaux c_i résultants ne peuvent prendre que deux valeurs (0,100) ; les seuils S_r et S'_r peuvent être des entiers quelconques $\in]0,100[$.

Nous expliciterons à titre d'exemple la recherche du ou des langages décrivant les caractères B. Pour ce faire, l'opérateur propose dans un premier temps un langage λ_B décrivant la classe C_B au moins en partie :

$$\lambda_B \begin{cases} L(g'_B) = OH^2 OV^{2,3} SH^3 SV^2 CH^2 CV^1 \\ \lambda''_B : \text{indifférent} \end{cases}$$

Le système effectue alors l'analyse de chacun des éléments du dictionnaire : le résultat de cette analyse apparaît à la figure 4.16 (page 64) où, vis à vis de chaque caractère du dictionnaire, est imprimé le résultat c_i . On constate :

- . tous les caractères B sauf deux seront reconnus comme tels, leur description appartenant au langage λ_B ;
- . parmi tous les caractères autres que B, seuls quelques caractères "8" seront également reconnus comme "B". Les caractères 8 et B formant un doublet classique, nous ne chercherons pas à les séparer à ce stade.

On analyse alors le cas des caractères B rejetés par λ_B :

- l'un de ces caractères (n° 6) possède la description suivante :

$$\begin{array}{cccccc} m & f & m & f & m & & m & f & m \\ & & & & & & & & & & f & f \\ & & & & & & & & & & & f \end{array}$$

qui traduit, entre autres, la présence d'une seule boucle fermée en bas du caractère. On trouve en effet dans la description horizontale une seule association :

$$\begin{array}{ccc} m & f & m \\ & & f \end{array}$$

Ce caractère présente donc une coupure importante dans la partie supérieure, ce qui lui donne la structure d'un 6. Nous ne nous attacherons donc pas à reconnaître ce caractère qui fera, au stade de la reconnaissance, l'objet d'un rejet.

FIGURE 4.16. Résultat de l'analyse du dictionnaire vis à vis de \mathcal{L}_B .

		pénalités													
		x_0	x_0+1	x_0-1	autres										
a(OH) :	2	100	100	100	100										
b(OV) :	3	100	0	100	100										
c(SH) :	3	100	100	100	100										
d(SV) :	2	100	100	100	100										
e(CH) :	2	100	100	100	100										
f(CV) :	1	100	100	100	100										
						FONCTIONS DESCRIPTIVES :									
A:	A:	A:	A:	A:	A:	A:	A:	A:	A:	A:	A:	A:	A:	A:	A:
B: 0100	B: 0100	B: 0100	B: 0100	B: 0100	B:	B: 0100	B: 0100	B:	B: 0100	B: 0100	B:	B: 0100	B: 0100	B: 0100	B: 0100
C:	C:	C:	C:	C:	C:	C:	C:	C:	C:	C:	C:	C:	C:	C:	C:
D:	D:	D:	D:	D:	D:	D:	D:	D:	D:	D:	D:	D:	D:	D:	D:
E:	E:	E:	E:	E:	E:	E:	E:	E:	E:	E:	E:	E:	E:	E:	E:
F:	F:	F:	F:	F:	F:	F:	F:	F:	F:	F:	F:	F:	F:	F:	F:
1:	1:	1:	1:	1:	1:	1:	1:	1:	1:	1:	1:	1:	1:	1:	1:
2:	2:	2:	2:	2:	2:	2:	2:	2:	2:	2:	2:	2:	2:	2:	2:
3:	3:	3:	3:	3:	3:	3:	3:	3:	3:	3:	3:	3:	3:	3:	3:
4:	4:	4:	4:	4:	4:	4:	4:	4:	4:	4:	4:	4:	4:	4:	4:
5:	5:	5:	5:	5:	5:	5:	5:	5:	5:	5:	5:	5:	5:	5:	5:
6:	6:	6:	6:	6:	6:	6:	6:	6:	6:	6:	6:	6:	6:	6:	6:
7:	7:	7:	7:	7:	7:	7:	7:	7:	7:	7:	7:	7:	7:	7:	7:
8: 0100	8: 0100	8:	8:	8:	8:	8:	8:	8:	8: 0100	8:	8:	8:	8:	8:	8:
9:	9:	9:	9:	9:	9:	9:	9:	9:	9:	9:	9:	9:	9:	9:	9:

- L'autre caractère (n° 9) lui, possède bien la structure d'un B mais cette structure diffère sensiblement de la structure commune. Il doit pourtant être reconnu ; nous le considérerons comme appartenant à une sous-classe du B caractérisée par un langage spécifique : $\mathcal{L}_{B'}$.

Les éléments de cette sous-classe présentent une déformation fréquente en écriture manuscrite : \mathcal{B} ; leur description sera : $mfmf \quad mfm$;
 $f \quad f \quad f$
 f

elle sera décrite par le langage $\mathcal{L}_{B'}$:

$$\mathcal{L}_{B'} \left\{ \begin{array}{l} L(g_{B'}) = OH^2 OV^3 SH^2 SV^2 CH^1 CV^1 \\ \mathcal{L}_{B'} : DH(m \ f \ m \ f) \\ \qquad \qquad \qquad f \quad f \end{array} \right. .$$

FIGURE 4.18. Ensemble des langages ℓ_j .

	OH	OV	SH	SV	CH	CV	DH	FH	NH	DV	FV	NV
A	2	2	1,2	1,2	1	1			mfmf f f			
A	2	1,2	2	1,2	0	0,1	fmm zzz					
A	2	2	1,2	1,2	0,1	1	ffmf f f					
A	2	2	1,2	1,2	0,1	1			ffmf f	mf		
B	2	2,3	3	2	2	1						
B	2	3	2	2	1	1	mfmf f f					
C	1,2	2	1,2	1,2	0	0,1		m z		m z		
C	1,2	2	1,2	1,2	0	0,1		m z		m m		
D	2	2	2,3	1,2	1	1		m				
D	2	2	2,3	1,2	1	1	mfmf zz			fm		
E	1,2	3	3	1	0	0				mff fz f		
E	1,2	3	3	1	0	0				mfz f f		
F	1,2	2	2	1	0	0		mf f		m		
F	1,2	2	2	1	0	0		mf z		fmf f		
1	1,2	1,2	0,1	1	0	0				mz		
1	1,2	1,2	0,1	1	0	0	fmfz z z				fm	
1	1,2	1,2	0,1,2	1	0	0	fmff f				fm	
1	1,2	1,2	0,1,2	1	0	0	fmff f				fmf z	
1	1,2	1,2	0,1,2	1	0	0	fff f				fm	
2	2	-	2	-	0	-		mffm f				fm f f
2	1	-	2	-	0	-		mfm				fm f f
2	-	-	3	-	1	-		m				mfmf f
3	-	3	1,3	1	0	-						fm f f
3	-	3	2	1	0	-		ffm f				fm f f

FIGURE 4.18 (suite).

	OH	OV	SH	SV	CH	CV	DH	FH	NH	DV	FV	NV
3	-	-	2,3	1,2	0	-						fm ff f
4	2	2	1	1,2	0,1	0,1	ffmf f z					
4	2	2	2	1	0,1	0,1			mfmf f z			ffm f
4	2	2	2	1	0,1	0,1			mfmf f z	fmz f		
4	2	2	2	2	0,1	0,1			mfmf f z	mfmf f z		
4	2	2	2	2	0,1	0,1			mfmf f z	fmfm f		
5	1,2	2,3	3	1,2	0	0						mff fm
5	1	3	2	1,2	0	0						mf f f
5	1	3	3	2	0	1				mfm ff f		
5	1	3	3	1	0	0				mf f f		
5	1,2	2,3	2	1	0	0			mffm f			mf f
6	2	2,3	3	1	1	0			mfmf z f			
6	2	3	2,3	2	1	0,1			mfmf z f			mfm z
6	2	3	2	1	1	0			fmfm z f			
7	1,2	2	1,2	1,2	0	0	mf	f				fm f
7	1,2	2,3	2	1,2	0	0		f	mf			fm m
8	2	2,3	3	2	1,2	0,1			fmf f f			
8	2	2,3	4	2	2	0,1						
8	2	2,3	3	2	1,2	0,1			ffmf f f	mfm		
8	2	2,3	-	2	-	1			fff f f			mfm fff
9	2	3	2,3	1,2	1	0,1	mfmf f z					fm z
9	2	2	2,3	1	1	0,1	mfmf f z					fm z

L'implantation de l'ensemble des tables g_i' et l_i'' a nécessité une taille mémoire de 2 Koctets. Notons que l'accroissement de l'indice i , donc du nombre de classes ou sous-classes à reconnaître ne se traduira que par l'allongement de ces tables de données. L'algorithme du classifieur et de l'organe de décision demeurent inchangés quelle que soit la taille des tables. De ce fait, il est aisé d'augmenter les langages l_i afin de parfaire le système de reconnaissance, d'augmenter le nombre de classes C_i afin d'étendre la reconnaissance aux caractères alphanumériques.

4.2.3. TEST DU CLASSIFIEUR

Nous avons effectué, dans un premier temps, la reconnaissance des caractères dont les descriptions ont servi à l'apprentissage. Ceci nous permettra de juger de la validité des langages proposés. Nous reconnaitrons ensuite des caractères quelconques : ayant participé à l'apprentissage, mais faisant l'objet d'une nouvelle acquisition, dactylographiés et manuscrits n'ayant pas participé à l'apprentissage.

a/ RECONNAISSANCE DU DICTIONNAIRE

Nous avons proposé au classifieur, en vue d'une reconnaissance, successivement toutes les descriptions issues du dictionnaire stocké en mémoire sous forme compressée. Rappelons qu'à présent le classifieur se présente dans le mode reconnaissance : une description inconnue sera analysée par les 46 langages. En fonction des capitaux c_i résultants, le système attribuera le caractère inconnu à au moins une classe, ou rejettera le caractère.

La figure 4.19 (page 69) représente les résultats de cette opération qui s'est déroulée automatiquement. Les règles d'écriture y sont les suivantes :

nom du caractère
issu du dictionnaire : nom de la ou des classes reconnues .

- . si le nom de l'unique classe reconnue est : - identique au nom du caractère, il y a un succès ;
- différente du nom du caractère, il y a substitution ;
- . si la réponse comprend plusieurs classes, il y a ambiguïté ;
- . si la réponse est : ? , il y a rejet .

Rappelons que le dictionnaire ne contient que des descriptions toutes différentes au sein d'une même classe. Ainsi, à partir du même nombre d'échantillons de caractères hexadécimaux, nous avons obtenu 19 descriptions différentes du caractère A, 14 du caractère B, ... Les caractères le plus complexes et sujets à de fréquentes variations de graphisme possèdent ainsi un plus grand nombre de descriptions.

FIGURE 4.19. Reconnaissance des descriptions du dictionnaire.

A:A,	C:C,	1:1,	3:3,	5:5,	7:7,	9:9,
A:A,	C:C,	1:1,	3:3,3,	5:5,5,	7:7,	9:9,
A:A,	C:C,	1:1,	3:3,	5:E,5,	7:7,	9:9,
A:A,	C:C,	1:1,	3:3,	5:5,	7:7,	9:9,
A:A,	C:C,	1:?	3:3,	5:5,5,	7:7,	9:4,9,
A:1,	C:C,	1:1,	3:3,	5:5,5,	7:7,	9:9,
A:A,	C:C,	1:?	3:3,	5:5,	7:7,	9:9,
A:?	C:C,	1:1,	3:3,3,	5:5,5,	7:7,	9:?
A:?	C:C,	1:1,	3:3,	5:5,	7:7,	9:?
A:A,	C:C,	1:1,	3:?	5:5,	7:7,	9:9,
A:A,	C:C,	1:1,	3:3,	5:5,	7:7,	
A:A,	C:C,	1:1,	3:3,	5:E,5,	7:7,	
A:A,F,	C:C,	1:1,	3:?	5:5,	7:?	
A:A,		1:1,	3:3,	5:?	7:7,	
A:A,	D:D,	1:1,	3:3,	5:?	7:7,	
A:A,	D:?		3:3,	5:E,5,	7:7,	
A:A,	D:D,	2:2,	3:?	5:5,5,		
A:A,	D:D,	2:2,	3:2,	5:?	8:B,8,	
A:A,	D:D,	2:2,	3:3,	5:5,	8:B,8,	
	D:D,	2:2,		5:3,	8:8,	
B:B,8,	D:D,4,	2:2,	4:4,		8:8,	
B:B,8,		2:2,	4:4,	6:6,	8:?	
B:B,8,	E:E,5,	2:2,	4:4,	6:6,	8:8,	
B:B,8,	E:E,5,	2:2,	4:4,	6:6,	8:B,8,	
B:B,8,	E:E,5,	2:2,	4:?	6:6,	8:8,	
B:?	E:E,	2:2,	4:?	6:6,	8:8,	
B:B,8,	E:E,	2:?	4:4,	6:?	8:8,	
B:B,8,	E:E,	2:2,	4:4,	6:6,	8:8,	
B:B,		2:?	4:4,	6:6,	8:?	
B:B,8,	F:F,	2:2,	4:4,	6:6,	8:B,8,	
B:B,8,	F:F,	2:2,	4:?	6:6,	8:8,	
B:B,8,	F:F,	2:2,	4:4,	6:6,	8:B,8,	
B:B,8,	F:F,	2:2,	4:?	6:6,	8:8,	
B:B,8,	F:?		4:4,	6:?	8:3,	
	F:F,		4:4,	6:6,	8:?	
				6:?		

Les résultats sont les suivants :

nombre de caractères hexadécimaux à reconnaître	:	210	
nombre de succès	:	152	soit : 73 %
nombre de substitutions	:	4	soit : 2 %
nombre d'ambiguïtés (y compris les doublets)	:	26	soit : 12 %
nombre de rejets	:	28	soit : 13 % .

Nous analysons successivement les trois cas d'insuccès :

★ Parmi les ambiguïtés, on rencontre 23 cas dus au doublet classique B,8 mais également au doublet E,5 dont les éléments ne pourront être séparés par notre description que si leur graphisme est de bonne qualité. Compte tenu de cette remarque, le taux d'ambiguïté effectif doit être ramené à 1,5 % et le taux de reconnaissance à 83 %. On rencontre également des ambiguïtés entre les caractères voisins atteints de fortes déformations :

A : A,F ; D : D,4 ; 9 : 4,9.

★ Nous comptons 28 rejets de caractères parmi lesquels les caractères les plus souvent rejetés sont par fréquence décroissante :

4, 3, 5, 6, 8, A, 1, 2, 9, B, D, F, 7.

Notons que ces rejets étaient attendus, car lors de l'élaboration des langages, nous n'avons pas associé de langage à ces caractères, ayant en effet jugé que leur graphisme très particulier n'était pas significatif.

On remarquera cependant que les caractères possédant des grandes obliques (caractère : 4) ainsi que les caractères constitués d'arcs de cercles sont plus souvent rejetés, ce qui souligne une certaine faiblesse de notre système à décrire les obliques et de manière moins prononcée les arcs de cercle.

★ Enfin, on dénombre quatre substitutions parmi lesquelles deux cas ont essentiellement pour cause une déformation importante accompagnée d'un graphisme particulier :

A : 1 où le caractère est écrit : $\overset{A}{\curvearrowright}$
 8 : 3 où le caractère est écrit : $\overset{8}{\curvearrowright}$

Les deux autres cas (3:2 ; 5:3) relèvent du problème suivant : ces caractères sont constitués de deux arcs de cercle en forme de "c" :

3 = $\overset{3}{\curvearrowright}$; 2 = $\overset{2}{\curvearrowright}$; 5 = $\overset{5}{\curvearrowright}$

La projection horizontale de tel graphisme ne peut, en aucun cas, les séparer.

Pour des caractères dactylographiés ou manuscrits de bonne qualité, la projection verticale permet de les discerner ; mais si l'on rencontre une écriture manuscrite possédant des caractères étroits, la séparation grâce à la projection verticale devient délicate voire impossible, ce qui explique ces substitutions ainsi que les rejets plus fréquents pour ce type de caractère.

Parmi les succès, on pourra noter le cas des réponses multiples : 3:3,3 et 5:5,5 dont les descriptions appartiennent à plusieurs langages de même nom non disjoints.

La figure 4.20 illustre les résultats obtenus après essai de reconnaissance de caractères autres qu'hexadécimaux. On constate l'intégrité des résultats :

- . la plupart des caractères sont rejetés,
- . certains caractères, faisant partie notamment des doublets classiques, sont assimilés à des classes de structure voisine : I:1 ; O:D ; P:4 ; Q:D ; R:A ; Z:2.

FIGURE 4.20. Analyse de caractères alphanumériques.

G:?	M:?	R:A,	W:?
G:?	N:?	R:A,	W:?
H:?	N:?	S:?	X:?
I:1,	O:D,	T:?	Y:?
I:1,	P:?	U:?	Z:2,
J:?	P:4,	V:?	
K:?	Q:D,		
L:?	Q:?		

b/ RECONNAISSANCE DE CARACTERES

Afin de tester le système de reconnaissance, nous avons acquis, traité, classé des caractères :

- . caractères ayant servi à l'élaboration du dictionnaire d'apprentissage, mais ayant fait l'objet d'une nouvelle acquisition ;
- . caractères écrits une seconde fois par les scripteurs ayant participé à l'apprentissage ;
- . caractères issus de scripteurs nouveaux n'ayant pas participé à l'apprentissage.

L'ensemble des caractères analysés représente 22 alphabets issus de 14 scripteurs différents (certaines personnes ont écrit plusieurs alphabets). Nous avons représenté les différents taux de reconnaissance à la figure 4.21.

FIGURE 4.21. Taux de reconnaissance.

	sans pondération		avec pondération	
	nombre	soit (%)	nombre	soit (%)
acquisitions	354		329	
succès	270	76	270	82
ambiguïtés	5	1	5	1,5
rejets	59	17	34	10,3
substitutions	20	6	20	6,1

Dans tout ce paragraphe, les essais ont été réalisés en tenant compte des remarques suivantes :

★ D'après ce qui précède, nous pouvons dire que les caractères B,8 et E,5 ne seront pas forcément séparés à ce niveau, mais lors d'une seconde étape où ils subiront des tests spécifiques. Abusivement, lors de l'analyse d'un caractère B ou 8, chaque fois que la réponse a été l'ambiguïté B,8, nous l'avons considérée comme étant correcte. Il en a été de même pour l'ambiguïté E,5, si elle est le résultat de l'analyse d'un E ou d'un 5.

★ Certains caractères, en particulier les caractères possédant des graphismes sophistiqués, ont été analysés deux fois afin de vérifier la constance de la réponse.

★ Nous avons introduit dans les résultats une pondération qui tient compte des caractères rejetés pendant l'apprentissage car considérés, peut-être à tort, comme non significatifs.

Nous avons représenté à la figure 4.22 (page 72) un tableau mettant en évidence les différentes réponses du classifieur en fonction des caractères analysés.

. 13 .

Nous constatons que :

☆ Certains caractères, comme le C, E, F, sont très bien reconnus quels que soient les scripteurs. Nous avons toutefois effectué des essais complémentaires de reconnaissance sur des caractères dactylographiés multifontes non appris (appellation IBM : ORATOR, OCR, ADVOCATE, COURRIER). Nous avons relevé la substitution E → B lors de l'analyse du caractère E. La reconnaissance de caractères manuscrits et de caractères dactylographiés constitue deux démarches différentes : combler les trous fréquents dans les caractères manuscrits (figure 4.23, page 74) s'oppose à détailler les empattements de certains caractères dactylographiés. Or, notre traitement, et plus particulièrement l'algorithme de simplification, vise à corriger les discontinuités accidentelles des caractères. C'est pourquoi nous devons renoncer à reconnaître certaines fontes de caractères dactylographiés (celles dont les caractères possèdent des empattements).

☆ En examinant le tableau suivant l'axe vertical, on peut obtenir une indication sur le degré de confiance que l'on peut avoir dans la réponse du classifieur. Si cette réponse est : B,C,E,3,4,6 ou 9, on peut accorder une grande confiance en ce résultat qui, dans le cadre de nos tests, a toujours été exact.

☆ On notera également l'apparition d'ambiguïtés de type nouveau : 2,3 ; 2,8, indiquant que ces différents langages ne sont pas disjoints et qu'ils contiennent des éléments non rencontrés pendant l'apprentissage.

☆ Si on classe les caractères en fonction du nombre de rejets (non pondéré) qu'ils ont subi : C,E,F (aucun rejet) ; A,D,6,7 (1,2 rejets) ; B,1,2,3,5,9 (3,4,5,6,7 rejets) ; 4,8 (supérieur à 7 rejets), nous mettons en évidence d'une part le fait que les caractères 4 et 8 subissent le plus de déformations et que, d'autre part, pendant l'apprentissage nous avons considéré à tort certaines descriptions comme étant non significatives et nous n'avons pas cru devoir créer des langages aptes à les contenir. Pendant la phase de test, nous avons notamment relevé 4 descriptions identiques de caractères 8 relevant d'une déformation particulière non contenues dans un langage. Il est donc nécessaire de compléter certains langages (notamment ℓ_4 et ℓ_8) en fonction des rejets obtenus pendant l'apprentissage et les tests, avant de passer à l'étape suivante dans l'élaboration du classifieur.

☆ Mis à part les caractères C,E,F, tous les autres caractères ont subi quelques substitutions. C'est pourquoi, quelle que soit la qualité de la description, en reconnaissance de caractères manuscrits, la constance de la réponse n'est pas totalement assurée. Il est donc tout à fait souhaitable de mettre en oeuvre des tests supplémentaires avant d'affirmer un résultat. Cette procédure, d'ailleurs obligatoire pour les doublets classiques, ne pourra qu'améliorer les résultats du Tableau 4.22 (page 72).

Nous avons détaillé à la figure 4.23 (page 74) un échantillon des différentes écritures ayant servi à tester le classifieur. La figure 4.24 (page 75) représente la digitalisation, le traitement et la classification de certains caractères issus de la figure 4.23.

FIGURE 4.23. Variétés d'écritures manuscrites.

SCRIPTEUR n°

1

0 1 2 3 4 5 6 7 8 9

A B C D E F

2

0 1 2 3 4 5 6 7 8 9

A B C D E F

3

A B C D E F

1 2 3 4 5 6 7 8 9

4

A B C D E F 1 2 3 4 5 6 7

8 9

NOUVEAU SCRIPTEUR
SANS APPRENTISSAGE

5

A B C D E F 1 2 3 4 5 6

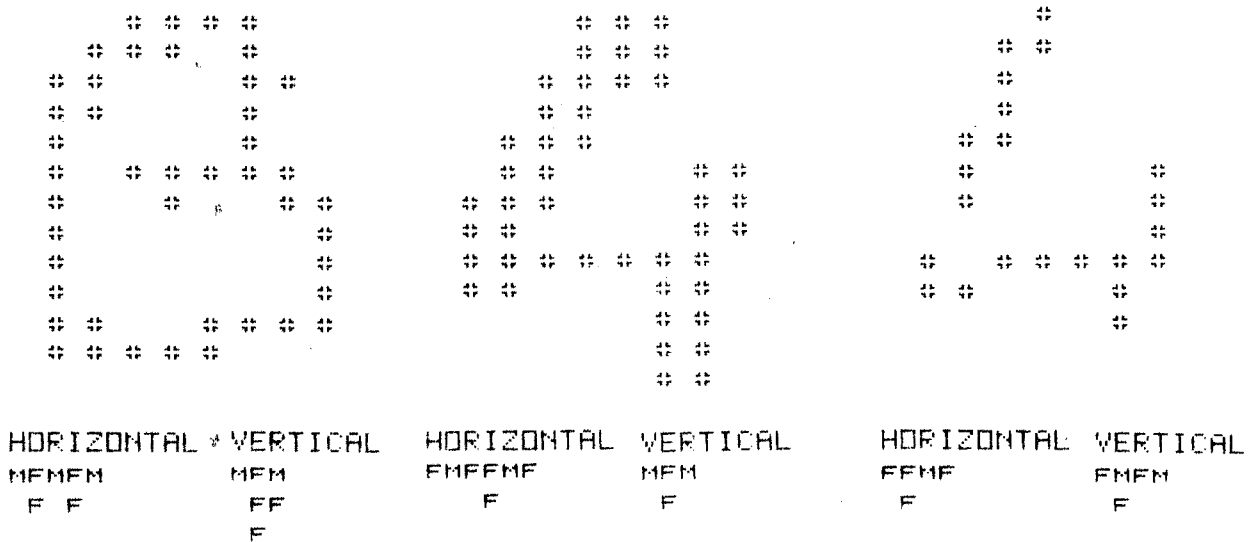
7 8 9

6

A B C D E F

1 2 3 4 5 6 7 8 9

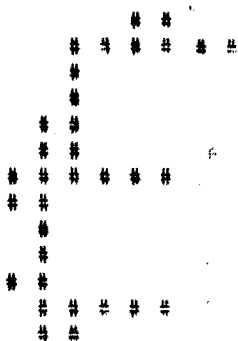
FIGURE 4.24. Traitement détaillé pour quelques caractères.



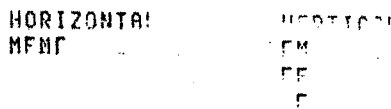
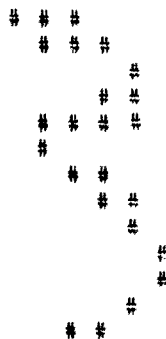
B,8
caractère B
du scripteur n° 1

A
caractère 4
du scripteur n° 1

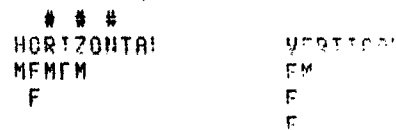
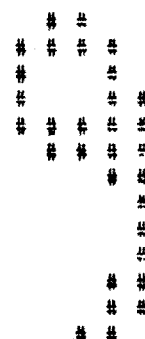
4
même caractère 4
du scripteur n° 1



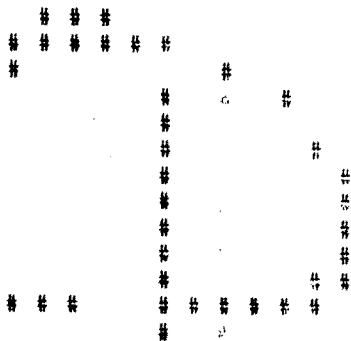
05
E, 5,
caractère E
du scripteur n° 3



04
3,
caractère 3
du scripteur n° 3



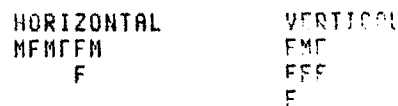
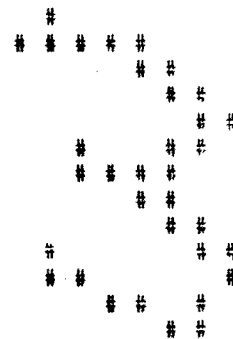
02
9,
caractère 9
du scripteur n° 3



06
D,
caractère D
du scripteur n° 5



07
2,
caractère 2
du scripteur n° 5

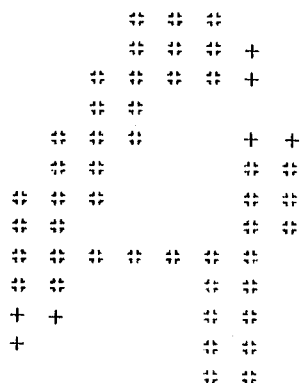


05
3,
caractère 3
du scripteur n° 5

Nous constatons que 8 caractères sont correctement reconnus bien qu'atteints de fortes déformations (B,E,3,D,3). L'organe de choix du seuil d'acquisition a fixé un seuil tel, que ces caractères sont pratiquement squelettisés. En effet, cet organe recherche le maximum de détails dans la descriptions du caractère ; rappelons que, pour la majorité des cas, ce but est atteint lorsque le caractère est à la limite de la coupure correspondant ainsi à une "squelettisation".

Par contre, on notera que le premier caractère 4 possède des segments d'épaisseur très différente et a été reconnu comme un caractère A (une seconde reconnaissance du même caractère a produit la réponse correcte : 4). En l'examinant, on remarquera qu'effectivement il suffit d'ajouter quelques points (notés +) pour que, visuellement, celui-ci se rapproche du caractère A (figure 4.25), ce qui justifie que la réponse erronée n'est pas totalement illogique.

FIGURE 4.25.



L'organe de choix du seuil recherche en premier lieu la description possédant le maximum de segments, aussi, lors de l'écriture de ce caractère (figure 4.24, page 75), l'arrêt du stylo au sommet de l'oblique gauche du caractère a provoqué une petite tache qui a été reconnue comme un segment horizontal lors de l'analyse horizontale. On peut expliquer que ce segment de très petite dimension ait été détecté, par le fait que le seuil de détection des maximums M tient compte de l'épaisseur des traits supposée à peu près constante dans le caractère analysé. Nous ne remettons pas en cause le choix de la moyenne de points comme seuil permettant de détecter les segments, mais ce qui précède confirme la nécessité d'analyser des caractères possédant une certaine constance dans l'épaisseur de leur trait (une variation d'épaisseur d'un pixel semble représenter, pour tous les cas, une limite acceptable).

4.3. SECOND NIVEAU DU CLASSIFIEUR

A l'issue de l'étape précédente, plus des 2/3 des caractères analysés sont correctement reconnus. Dans les caractères restants, nous trouvons : les doublets B-8, E-5, dont nous savons que notre procédé de description ne pourra les séparer dans tous les cas ; des classes d'ambiguïté nouvelles apparaissent, dues soit au procédé de description, soit au graphisme des caractères, soit à un défaut d'apprentissage ; des substitutions sont également possibles pour les mêmes raisons que précédemment.

L'objet de la reconnaissance de deuxième niveau sera de corriger ces erreurs ou imperfections par application de tests locaux adaptés à chaque cas de figure. Pour cela, si nous examinons chaque colonne du tableau de la figure 4.22 (page 72), nous constatons trois situations :

★ La réponse du classifieur est unique et toujours correcte : C,3,4,6,9. Nous considérerons que ces réponses sont correctes ; la reconnaissance est effectuée dès la fin du premier niveau.

★ La réponse du classifieur est double, mais cette double réponse a toujours correspondu au même caractère à reconnaître :

$$D \cap 4 \rightarrow D \quad ; \quad 2 \cap 3 \rightarrow 2 \quad ; \quad 2 \cap 8 \rightarrow 2 \quad ; \quad 4 \cap 9 \rightarrow 9 .$$

Nous pourrions, dans ce cas, élaborer la réponse définitive par une opération triviale de transcodage. Cependant, eu égard à la taille modeste de l'ensemble de test, nous préférerons confirmer la réponse par des tests complémentaires.

★ La réponse du classifieur est unique, mais a donné lieu à des substitutions lors de l'essai précédent. Dans ce cas, nous considérerons cette réponse unique comme une indication de classes possibles auxquelles appartiendrait le caractère à reconnaître. Nous assimilerons à ce cas de figure la réponse unique ou non, spécifique des doublets B-8 et E-5.

réponse		classes possibles
A	→	A,1,4
D	→	D,B,6,8,9
F	→	F,5
1	→	1,A
2	→	2,3,8
7	→	7,5
$\bar{E} \cap 5$	→	5,3
$E \cap \bar{5}$	→	E
$E \cap 5$	→	E,5
$B \cap \bar{8}$	→	B
$\bar{B} \cap 8$	→	8,A
$B \cap 8$	→	B,8,D .

Cette analyse montre que nous aurons à effectuer des tests ayant pour objet :

. de confirmer une réponse unique ou non :

- $D \cap 4 \rightarrow D$
- $2 \cap 3 \rightarrow 2$
- $2 \cap 8 \rightarrow 2$
- $4 \cap 9 \rightarrow 9$
- $B \cap \bar{8} \rightarrow B$
- $E \cap \bar{5} \rightarrow E ;$

. de séparer des classes possibles :

- | | | | | | | | | |
|---|---|-----------|------------------|---|-------|------------------|---|-------|
| A | → | A,1,4 | 2 | → | 2,3,8 | $\bar{B} \cap 8$ | → | 8,A |
| D | → | D,B,6,8,9 | 7 | → | 7,5 | $B \cap 8$ | → | B,8,D |
| F | → | F,5 | $\bar{E} \cap 5$ | → | 5,3 | | | |
| 1 | → | 1,A | $E \cap 5$ | → | E,5 | | | |

4.3.1. ENSEMBLE DE TESTS DU DEUXIEME NIVEAU

L'examen des divers cas exposés dans le paragraphe précédent nous a conduit à définir cinq tests appliqués au caractère digitalisé en des zones spécifiques de la réponse du classifieur de premier niveau.

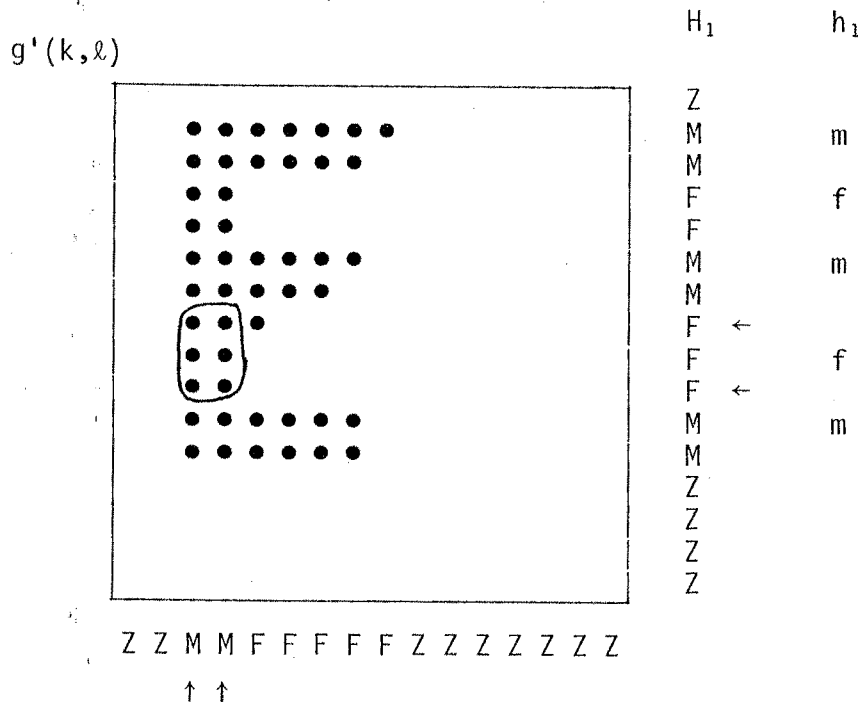
Si l'on désire par exemple séparer les caractères E et 5, nous nous intéresserons à deux zones particulières du caractère (figure 4.26).

FIGURE 4.26.



En ce qui concerne notre système, cette opération de localisation des zones de la matrice $g'(k, \ell)$ utilisera la description du caractère (h_k et v_k) et les chaînes horizontales et verticales de caractères (H_k et V_k) qui sont mémorisées, pour obtenir les coordonnées horizontales et verticales de la zone à analyser. En reprenant l'exemple de la figure 4.26, on obtient pour le caractère E (figure 4.27, page 79) les coordonnées de la première zone qui doit se situer entre le deuxième et le troisième segment horizontal à gauche du caractère. Sur h_1 puis H_1 , on tire la position ligne : entre la huitième et la dixième ligne incluse. Une analyse identique sur v_1 et V_1 nous donne la position colonne : colonnes 3 et 4.

FIGURE 4.27. Localisation d'une zone de test.



Le système localise ainsi les zones utiles :

- . quel que soit l'emplacement du caractère dans la matrice d'acquisition g' ,
- . quel que soit le graphisme de ce caractère,
- . de manière simple et précise.

Les points définissant les zones constitueront ainsi les paramètres d'entrée de ces tests que nous expliciterons en utilisant un exemple significatif.

• TEST DE LIAISON

Ce test répond à la question : Y a-t-il des pixels "1" entre les points A et B ? Nous noterons l'interrogation par :

$$A \overset{L}{-} B$$

en ayant précisé auparavant les points A et B. En ce qui concerne l'exemple de la figure 4.26 (page 78), la liaison entre A et B existe pour le caractère E et n'existe pas pour le caractère 5.

• TEST DE TROUS

Ce test est plus particulièrement utilisé pour séparer les caractères B et 8. Dans l'exemple de la figure 4.28 (page 80), le segment vertical gauche du caractère 8 a été assimilé, par l'organe de traitement, à un segment sans discontinuité.

Il suffit d'examiner, sur le caractère digitalisé, la région centrale du segment vertical et de rechercher la présence de trous fréquents dans le caractère 8 ; le cas contraire se rencontrera sur le caractère B.

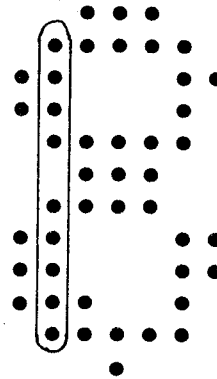


FIGURE 4.28.

• TEST DE LONGUEUR

Cette mesure s'exprime en pixels et, pour ce faire, il suffit de rechercher le nombre L_{ik} (ou C_{jk}) (rangé en mémoire) correspondant au segment qui contient l'information. Si un segment a une épaisseur de plusieurs pixels, nous prendrons comme mesure le plus grand des nombres L_{ik} (ou C_{jk}).

• TEST DE COURBURE

En reprenant l'exemple de la figure 4.28, on recherchera, dans ce cas à gauche du segment, la présence de points. Si de tels points sont présents, le segment sera dit bombé et sera significatif d'un caractère 8 ; dans le cas contraire (absence de points), le segment est rectiligne et se rencontrera notamment dans le caractère B.

• TEST DES COINS

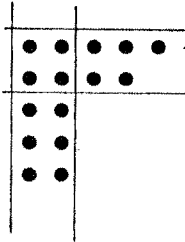
En conservant l'exemple du doublet B-8, nous analysons l'angle supérieur gauche formé par la rencontre du segment horizontal supérieur et du segment vertical gauche. Soient, dans l'espace délimité par la rencontre de ces deux segments :

p : le nombre de points appartenant au caractère,
 P : le nombre de pixels formant l'espace considéré,

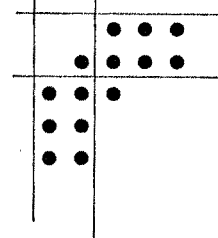
- si $p = P$, l'angle est droit (figure 4.29.a) : cas du B ;
- $0 \leq p < P$, on a un arc de cercle (fig.4.29.b) : cas du 8 .

FIGURE 4.29. Détermination de l'angle.

(a) angle droit.



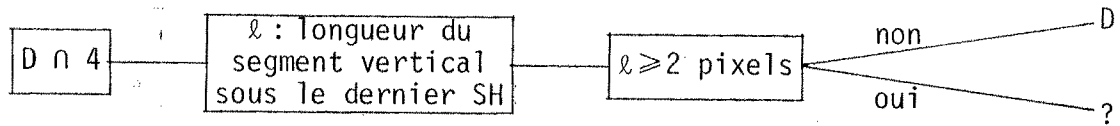
(b) arc de cercle.



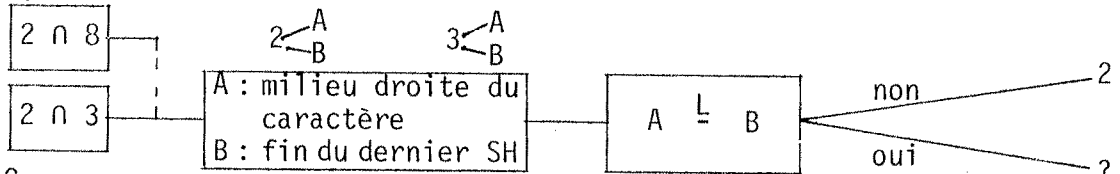
4.3.2. CLASSIFICATION ARBORESCENTE DU DEUXIEME NIVEAU

Comme nous l'avons dit précédemment, les tests du deuxième niveau ont pour objet de : soit confirmer une réponse, soit séparer les classes à l'intérieur d'un groupe. Nous proposons dans ce qui suit et pour chaque cas les tests à effectuer. Ces tests ont été établis en examinant les caractères qui ont donné lieu à une ambiguïté ou à une substitution. Certains de ces caractères étant particulièrement déformés, les tests que nous proposons pourront conduire à un rejet.

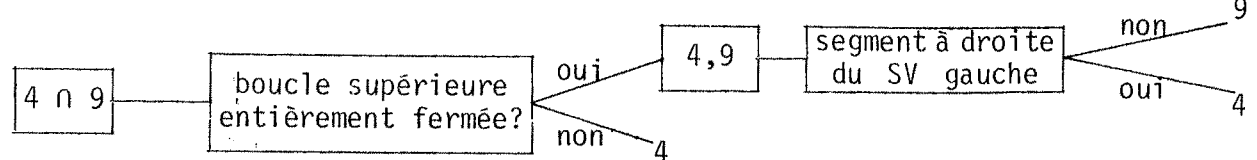
★ D n 4



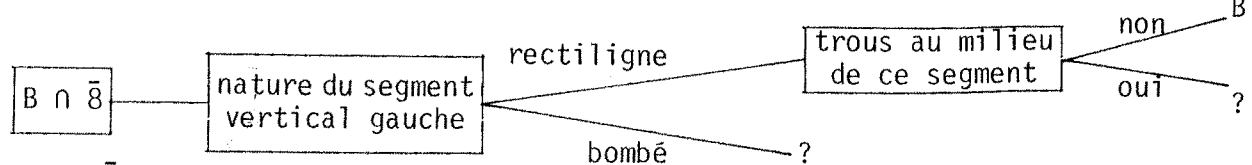
★ 2 n 8, 2 n 3



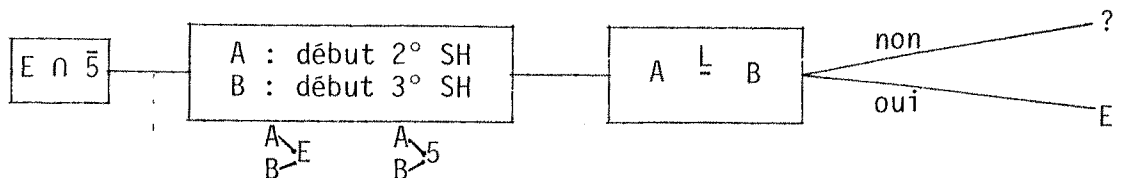
★ 4 n 9



★ B n 8



★ E n 5

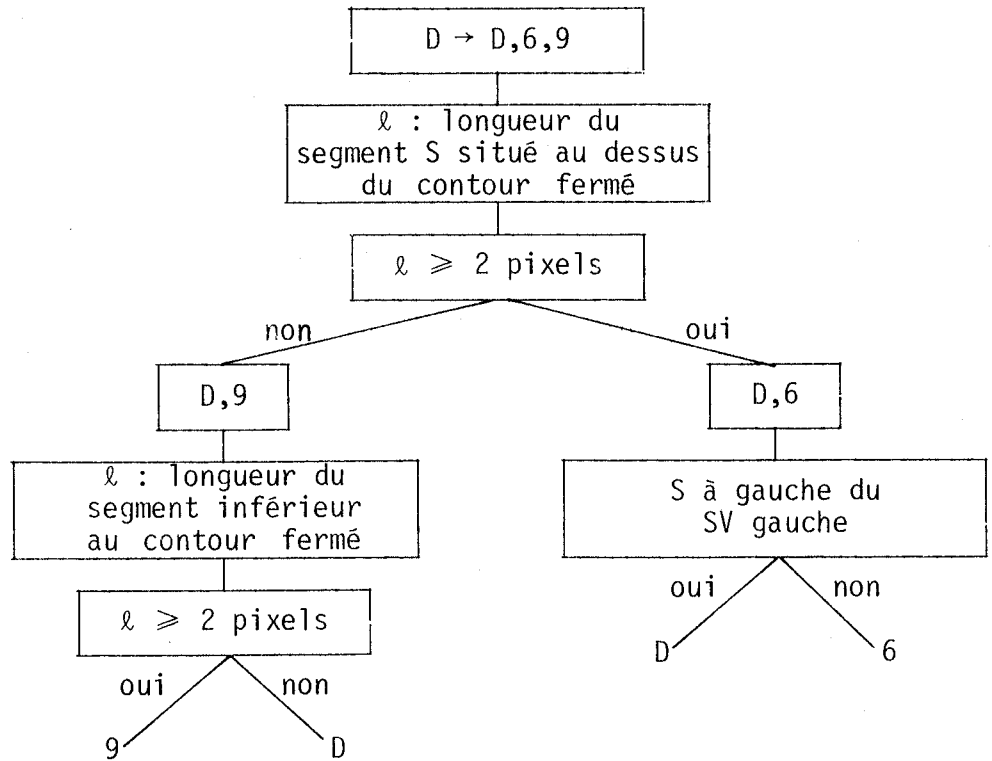


★ A → A,1,4

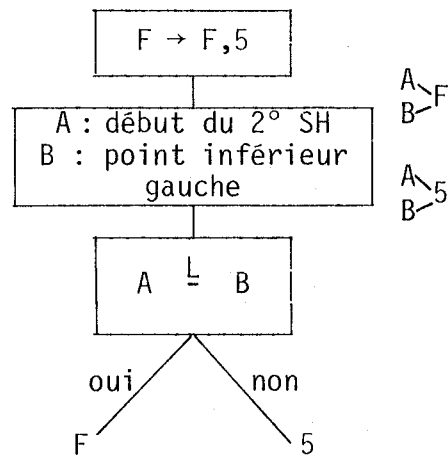
La dispersion que l'on a observée pendant l'apprentissage et les essais sur ces caractères, notamment A,1, est telle que nous n'avons pas établi de tests pour ce groupe.

★ D → D,B,6,8,9

De ce groupe, nous retirons les cas uniques des substitutions B et 8 assimilés par le classifieur à un caractère D. En effet, les deux caractères responsables de cette substitution sont très déformés et, de toute manière, ne pourront être reconnus.



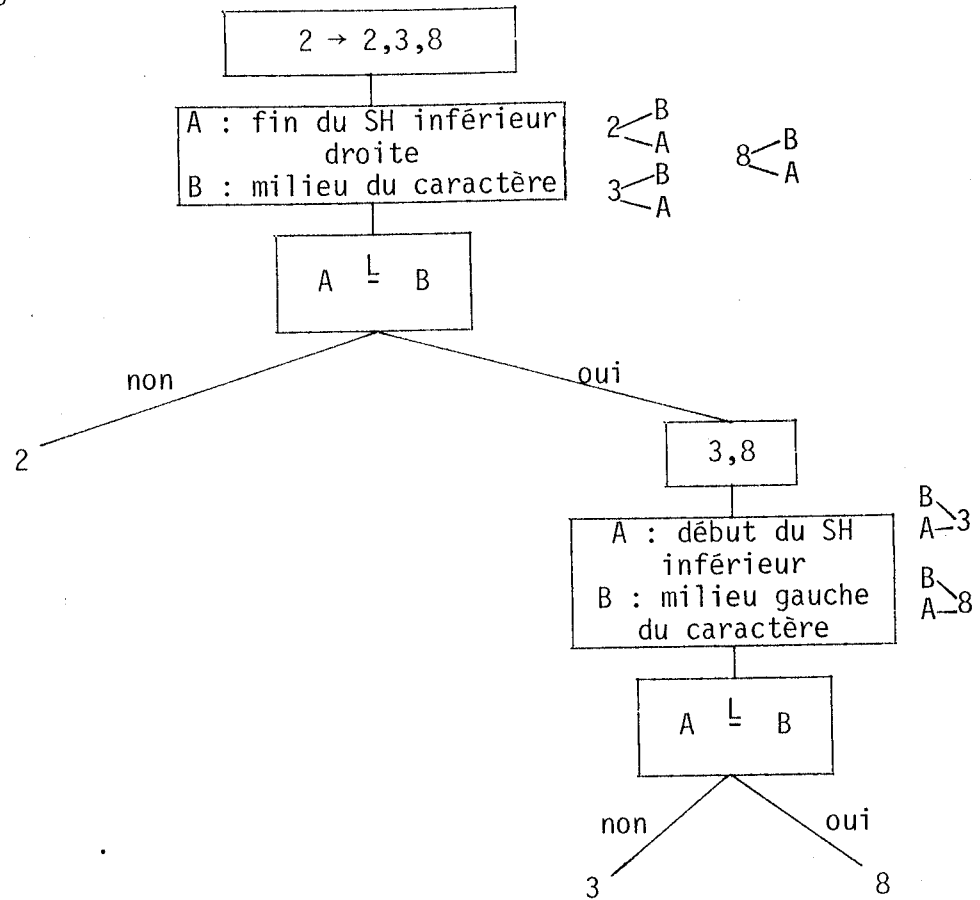
★ F → F,5



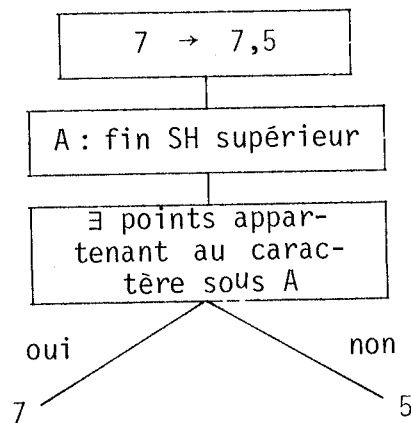
★ 1 → 1,A

Pour des raisons semblables à celles du groupe A → A,1,4, nous n'avons pas établi de tests pour ce groupe.

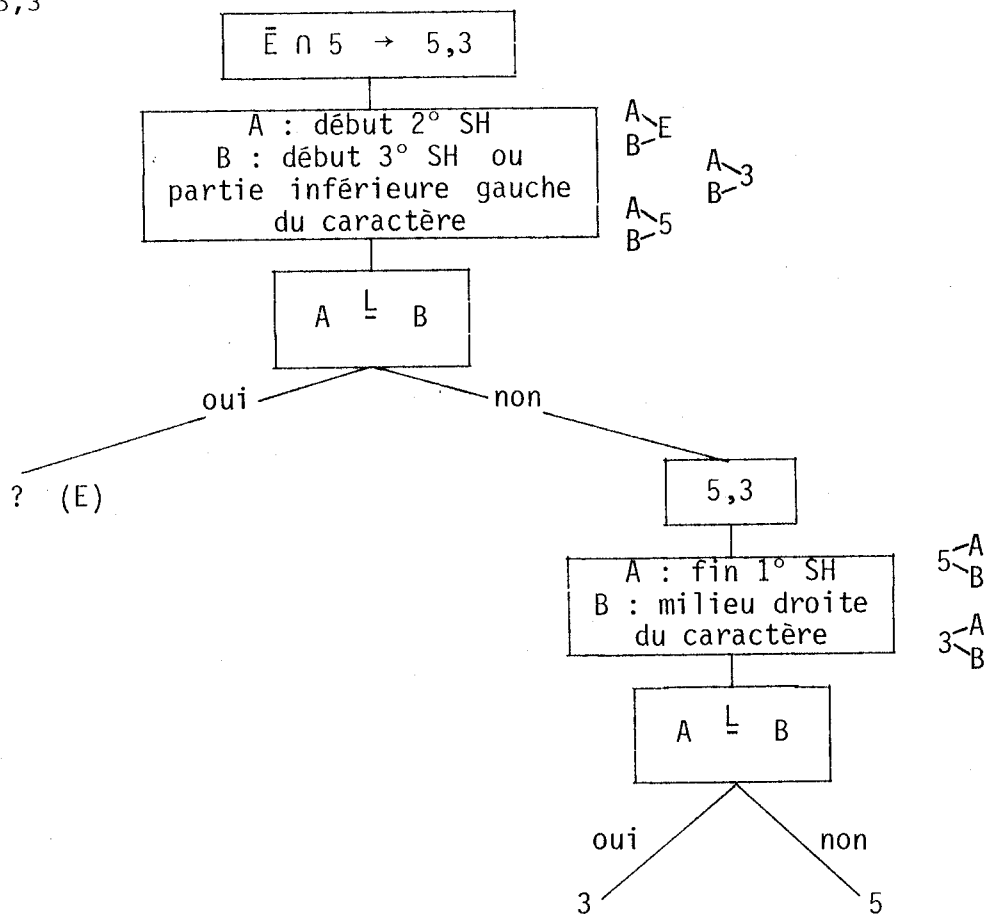
★ 2 → 2,3,8



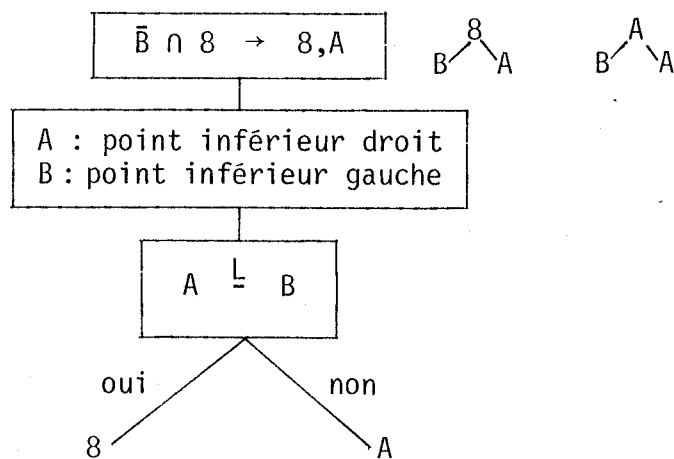
★ 7 → 7,5



★ $\bar{E} \cap 5 \rightarrow 5,3$



★ $\bar{B} \cap 8 \rightarrow 8,A$



★ Enfin, pour séparer les doublets B∩8 qui ont des graphismes très voisins, et les "doublets" E∩5 dont les descriptions sont pour nous semblables, nous avons élaboré des tests multiples. Chaque test conduira à une réponse qui sera prise en compte lors de la décision finale qui est à caractère majoritaire.

★ B ∩ 8 → B,8,D

Le seul caractère D ayant été assimilé au doublet B∩8 ne sera pas séparé car celui-ci atteint par trop de déformations n'est pas identifiable. En ce qui concerne le caractère B manuscrit, le segment vertical gauche ainsi que l'angle supérieur gauche ont été, pour tous les scripteurs, les parties tracées en premier lieu et, de ce fait, les moins déformées ; cette remarque s'applique de façon moins systématique au caractère 8 ; nous examinerons donc en priorité cette zone.

. Test des trous au milieu du segment vertical gauche :

présence de trous : 8
absence de trous : B

. Détermination de la nature du segment vertical gauche :

bombé : 8
rectiligne : B

. Mesure de la distance d séparant les deux segments verticaux :
soit par exemple la phrase V1 : MMFFFMZ¹⁰, alors d = 3

mesure de la longueur l du segment horizontal supérieur :
si l > d+1 : B
si l ≤ d+1 : 8

. Test du coin supérieur gauche :

arc de cercle : 8
angle droit : B

La réponse finale est obtenue après décision à caractère majoritaire. En cas d'égalité, la réponse du classifieur sera l'ambiguïté : B ∩ 8 .

★ E ∩ 5 → E,5

. Soient : A début du 2° SH (figure 4.26, page 78)

B début du 3° SH
existe-t-il une liaison entre A et B ? oui : E
non : 5

. Soient : C fin du 2° SH (figure 4.26)

D fin du 3° SH
existe-t-il une liaison entre C et D ? oui : 5
non : E .

Comme précédemment, la décision est à caractère majoritaire ; en cas de conflit, la réponse demeurera l'ambiguïté.

5 - CONCLUSION

Le but que nous nous sommes fixé dans ce travail est de définir un procédé de reconnaissance de caractères manuscrits et dactylographiés, et de réaliser un appareil utilisant ce procédé. Pour tester les divers algorithmes, nous avons utilisé des moyens micro-informatiques qui nous ont permis de concevoir un appareillage autonome, d'une grande souplesse d'utilisation. S'agissant d'un appareillage utilisé pour la recherche, nous n'avons pas cherché à optimiser les performances de vitesse.

Dans cette étude, nous avons privilégié la reconnaissance de caractères manuscrits, pensant que la reconnaissance de caractères dactylographiés serait acquise du même coup. Nous avons donc choisi une reconnaissance de type structural et nous avons été amené à définir :

- un algorithme de projection et de description du caractère qui, à partir des projections horizontales et verticales, va décrire la structure du caractère à l'aide de chaînes de caractères qui, après simplification et normalisation, donnent 8 phrases de description. Nous avons noté l'invariance des résultats vis à vis des homothéties, translations, rotations (15 à 20° de tolérance) et déformations présentes dans les caractères manuscrits. Toutefois, ces remarques ne s'appliquent que si les caractères sont "correctement" digitalisés : notamment espace intérieur au caractère suffisamment important.
- un organe de choix du seuil d'acquisition qui va, en fonction des descriptions, sélectionner, en interaction avec l'organe d'acquisition, le seuil le mieux adapté au système de traitement. Les résultats obtenus sont très satisfaisants, mais, en contrepartie, cet organe est avide en temps de calcul. Nous verrons que les observations résultant des tests du classifieur apporteront des solutions à ce problème.
- un classifieur de type syntaxique permettant la reconnaissance de caractères manuscrits déformés. Un apprentissage effectué après élaboration d'un dictionnaire a permis d'établir les grammaires et les langages du premier niveau contenant entre autres les caractères

hexadécimaux vus pendant l'apprentissage. La structure adoptée pour le classifieur permet d'accroître aisément le nombre de langages afin de décrire des caractères de graphismes particuliers, et le nombre de classes afin de reconnaître d'autres caractères (l'ensemble des caractères alphanumériques). A ce niveau, les caractères simples sont séparés, et nous avons établi dans un second niveau des tests spécifiques destinés à séparer les doublets classiques, les ambiguïtés et à confirmer les réponses ayant subi des substitutions lors des essais. Ces tests locaux portent sur un nombre réduit de caractères (3 ou 4) préalablement sélectionnés par le premier niveau.

De notre étude, nous avons tiré les conclusions suivantes :

- 1 - Notre procédé de reconnaissance est applicable aux caractères dactylographiés, à la condition toutefois que ceux-ci soient exempts d'empattements : en effet, l'écriture manuscrite laisse souvent apparaître des graphismes incomplets ou troués qui sont généralement comblés par l'organe de traitement, alors que les empattements des caractères dactylographiés doivent au contraire être éliminés, ce qui supposerait que l'on effectue préalablement la reconnaissance de ceux-ci !
- 2 - Les taux de reconnaissance que nous avons obtenus (supérieurs à 75 %) l'ont été avec des caractères non conformes à un modèle. Nous avons simplement demandé aux différents scripteurs d'écrire lisiblement : si pour certains caractères :

C , C , C ; o , D ; ...

les déformations sont acceptées, le problème peut devenir inextricable pour d'autres :

2 , 2 , 2 , 8 , ...

Il aurait été intéressant de tester les taux de reconnaissance de caractères manuscrits guidés. Tout nous permet de croire qu'ils auraient été bien supérieurs.

- 3 - Les problèmes essentiels que nous avons rencontrés lors de la reconnaissance de caractères de bonne qualité ont été principalement dus à l'opérateur de prétraitement - réduction. Cette opération qui réalise une sommation locale sur une surface de 4×4 pixels, conduit à déterminer le seuil de décision avec une grande précision. De plus, après réduction, les caractères à reconnaître sont inscrits dans une matrice 16×16, ce qui semble limite pour les caractères manuscrits. Il serait intéressant de tester les performances du système en utilisant des réductions moindres, voire sans réduction. Outre la disparition de l'organe de choix du seuil d'acquisition, on note que l'organe de traitement resterait inchangé et ses performances accrues ; par contre, le temps de traitement serait allongé.

- 4 - Nous avons remarqué que, dans l'écriture manuscrite, même avec le matériel moderne (excluant par exemple la plume sergent-major !), l'épaisseur des traits n'est pas rigoureusement constante. Ceci pose dans certains cas limites des problèmes d'adaptation du seuil de détection des maximums dans une projection horizontale ou verticale. Là encore, il serait souhaitable de tester les performances du système lorsqu'une squelettisation [8,30] préalable a été effectuée par l'organe d'acquisition indépendamment des étages suivants.

De ces remarques, nous concluons que la qualité d'un procédé de reconnaissance est fortement liée à la qualité du prétraitement.

B I B L I O G R A P H I E

- [1] Ching Y. SUEN
Feature extraction in automatic recognition of handprinted characters
Signal processing : Theories and applications, Eurasip, 491-501 (1980)
- [2] M.D. FREEDMAN
Optical character recognition
I.E.E.E. Spectrum, 44-52 (mars 1974)
- [3] G.J. BALM
An introduction to optical character reader considerations
Pattern Recognition, 2 n°3, 151-166 (1970)
- [4] A. GRASSELLI
Automatic interpretation and classification of images
Academic Press, New-York (1969)
- [5] A. GÜDESEN
Quantitative analysis of preprocessing techniques for the recognition
of handprinted characters
Pattern Recognition, 8 n°4, 219-227 (1976)
- [6] G. GAILLAT
Nouveaux regards sur la reconnaissance des caractères
Revue technique Thomson-CSF, 9, n° 3, 529-546 (septembre 1977)
- [7] K.M. SAYRE
Machine recognition of handwritten words : A project report
Pattern Recognition, 5 n°3, 213-228 (1973)
- [8] MICLET, GRENIER, KALISZEWSKI
Reconnaissance des formes
Cours E.N.S.T. (1978)
- [9] A. MURE-RAVAUD
Système de reconnaissance automatique des chiffres manuscrits utili-
sant la transformation de Walsh-Hadamard
Automatisme, 212-218 (juillet/août 1978)

- [10] H.C. ANDREWS
Multidimensional rotations in feature selection
I.E.E.E. Trans. on Comp., C.20, 1045-1051 (septembre 1971)
- [11] H. NIEMANN
Classification of characters by man and by machine
Pattern Recognition, 9 n°4, 173-179 (1977)
- [12] M.G. THOMASON, R.C. GONZALEZ
Syntactic recognition of imperfectly specified patterns
I.E.E.E. Trans. on Comp., C.24, 93-95 (janvier 1975)
- [13] R. NARASIMHAN, V.S.N. REDDY
A syntax-aided recognition scheme for handprinted english letters
Pattern Recognition, 3 n°4, 345-361 (1971)
- [14] K.S. FU
Syntactic methods in pattern recognition
in Mathematics in science and engineering, 112
Academic Press, New-York (1974)
- [15] M.R. ITO, T.L. CHUI
On-line computer recognition of proposed standard ANSI(USASI) hand-
printed characters
Pattern Recognition, 10 n°5/6, 341-349 (1978)
- [16] G. GAILLAT
Panorama des techniques d'extraction de traits caractéristiques en
lecture optique des caractères
Congrès AFCET IRIA, Toulouse, 3, 353-368 (1979)
- [17] S. CASTAN, G. PERENNOU
Reconnaissance structurelle de graphismes
Congrès AFCET IRIA, Chatenay-Malabry, 1, 63-72 (1978)
- [18] I.K. SETHI, B. CHATTERJEE
Machine recognition of constrained handprinted Devanagari
Pattern Recognition, n° 2, 69-75 (1977)
- [19] A. CHEHIKIAN, F. HADJ HASSAN
Un algorithme de séparation des primitives verticales horizontales
et obliques d'une forme bidirectionnelle
Congrès AFCET IRIA, Chatenay-Malabry, 2, 771-788 (1978)
- [20] A. CHAPPELLE, J. SERRA, G. LORETTE, J.P. REQUIER
Reconnaissance automatique de caractères manuscrits bâton
Congrès AFCET IRIA, Chatenay-Malabry, 2, 560-567 (1978)
- [21] L. MICLET, C. GUERIN
Inférence multiple en grammaires régulières
Congrès AFCET IRIA, Chatenay-Malabry, 1, 73-88 (1978)
- [22] L. MICLET
Inférence d'automates finis
Congrès AFCET IRIA, Toulouse, 1, 389-396 (1979)

- [22] G. BOUVIER
Système d'acquisition et de reconnaissance de caractères alphanumériques
Thèse 3ème Cycle, ENSER - INP Grenoble (1977)
- [24] A. KINDER
Calculateur adapteur au codage des formes
Rapport D.E.A., ENSER Grenoble (1978)
- [25] A. KINDER, G. BOUVIER, A. CHEHIKIAN
Codage des caractères alphanumériques par des chaînes de caractères
Congrès AFCET IRIA, Toulouse, 3, 324-331 (1979)
- [26] A. CHEHIKIAN
Conception et réalisation d'un automate de reconnaissance de caractères alphanumériques
Thèse d'Etat, USM-INP Grenoble (1977)
- [27] B. DUERR, W. HAETTICH, H. TROPF, G. WINKLER
A combination of statistical and syntactical pattern recognition
applied to classification of unconstrained handwritten numerals
Pattern Recognition, 12 n° 3, 189-199 (1980)
- [28] G. BOUVIER, A. KINDER
Processeur de saisie et de prétraitement de documents
proposé au Congrès AFCET IRIA de Nancy
- [29] R.C. GONZALEZ, P. WINTZ
Digital image processing
Addison-Wesley Pub. Co., New-York (1977)
- [30] P. MARTHON, A. BRUEL, G. BIGUET
Squelettisation par calcul d'une fonction discriminante sur un voi-
sinage de 8 points
Congrès AFCET IRIA, Toulouse, 2, 107-114 (1979) .

