TECHNIQUES
DE L'INGÉNIEUR

Réf. : **H1348 V1**

Reconnaissance de l'imprimé

Date de publication :
10 mai 1999

Cet article est issu de : **Archives**

par **Philippe LEFÈVRE**

Pour toute question :
Service Relation clientèle
Techniques de l'Ingénieur
Immeuble Pleyad 1
39, boulevard Ornano
93288 Saint-Denis Cedex

Par mail :
infos.clients@teching.com
Par téléphone :
00 33 (0)1 53 35 20 20

Document téléchargé le : **16/10/2019**
Pour le compte : **7200082594 - sorbonne universite // 195.220.213.14**

© Techniques de l'Ingénieur | tous droits réservés

Reconnaissance de l'imprimé

par **Philippe LEFÈVRE**
Ingénieur ESE
Direction des Études et Recherches d'EDF

L'invention du procédé d'impression typographique par Gutenberg vers 1440 a transformé radicalement notre société par une diffusion plus large et plus rapide des connaissances. L'avènement actuel des réseaux et la dématérialisation de l'information, qui devient électronique et numérique, constituent une révolution de même importance.

*Le rêve d'un **monde sans papier**, qui hante les professionnels de l'informatique et de la documentation depuis bientôt quatre décennies, semble sur le point de devenir une réalité : on ne peut plus ouvrir une revue informatique sans y trouver plusieurs articles sur Internet, les bases de données en ligne, les CD-ROM... L'information est devenue aujourd'hui omniprésente, et sa maîtrise est considérée comme un facteur essentiel de réussite. Or cette information est constituée à 80 % de données textuelles. Les connaissances, qu'elles soient techniques, scientifiques, historiques, économiques, juridiques, médicales... sont en majorité mémorisées et véhiculées par des textes. Celles qui ont été publiées récemment sont directement accessibles sous forme électronique. Par contre, la majorité du patrimoine culturel et technique de l'humanité n'est encore disponible que sous forme de documents papier. Les entreprises et les collectivités sont ainsi confrontées à un besoin énorme de retraitement, dit aussi **conversion rétrospective**, pour passer à un format électronique.*

*Ce besoin, en plus du défi de faire lire l'ordinateur comme un être humain, a motivé de nombreuses études depuis les années 1960. Elles ont produit de multiples logiciels de **reconnaissance de caractères**. Les résultats ont souvent été décevants, car la complexité du problème avait été largement sous-estimée au départ, et les puissances informatiques nécessaires à l'accomplissement d'une telle tâche avec une productivité suffisante ne sont disponibles que depuis peu.*

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

1. Domaine d'intérêt, types de documents et applications

1.1 Domaine d'intérêt

Le domaine étudié ici concerne la **reconnaissance des documents dactylographiés et imprimés sur support papier** : livres, journaux, revues, notes techniques, imprimés divers... qui sont de nature majoritairement textuelle. En sont exclus tous les documents manuscrits, ainsi que les formulaires, plans, schémas, dessins, photos... c'est-à-dire les documents à dominante graphique ou photographique.

1.2 Supports et formes de documents

Du fait des progrès techniques, la notion de document a évolué au cours du temps. Elle recouvre aujourd'hui plusieurs supports et formes (ou formats) qui coexistent. Nous distinguerons :

— les documents imprimés sur support papier, que nous appellerons **forme papier** d'un document ;

— les documents électroniques, ou numériques, stockés sur support informatique, qui se divisent eux-mêmes en :

- documents électroniques en mode image, que nous appellerons **forme image**,
- documents électroniques codés, par exemple en ASCII, dits **forme codée**,
- documents électroniques codés et structurés (avec des marques de titre, paragraphe...), ou **forme structurée**.

1.3 Formes images

Du fait de la complexité de la reconnaissance, la gestion électronique de documents (GED) a jusqu'ici principalement utilisé la forme image, ou image numérique, obtenue après numérisation des documents à l'aide d'un scanner. Celle-ci est une représentation de la page par une succession de *pixels*, ou points élémentaires, qui sont l'équivalent des grains d'une photographie. Plusieurs codages des images sont utilisés. En GED, on utilise essentiellement un codage biniveau : chaque pixel est représenté par un seul bit qui prend deux valeurs : 1 ou 0 (noir ou blanc). Cependant, la reproduction fidèle des photos et des documents de qualité dégradée nécessite un codage en niveaux de gris, voire en couleurs (§ 2.2 et 4.1).

1.4 Formes électroniques codées

La forme électronique codée d'un texte est une succession de codes : EBCDIC, ASCII, ANSI, UNICODE... représentant la suite des caractères du texte, plus d'autres codes décrivant la typographie et la mise en page. De nombreux formats électroniques existent, associés aux traitements de texte et aux logiciels de publication assistée par ordinateur (PAO). Le format RTF (*rich text format*) est aujourd'hui un des plus répandus.

La forme codée et structurée décrit en plus la structure logique du document (§ 2.4.2), c'est-à-dire qu'elle inclut les éléments de texte dans une organisation logique : titres, chapitres, paragraphes, notes... La norme SGML (*standard generalised markup language*) [1] permet le codage des documents structurés.

1.5 Reconnaissance de documents

Définition : on appelle reconnaissance de documents le passage de la forme papier ou d'une forme image d'un document, à une forme électronique codée ou structurée (figure **1**).

On utilisait autrefois le terme OCR (*optical character recognition*) ou reconnaissance de caractères. Comme l'OCR ne représente qu'une étape du processus, on préfère employer aujourd'hui les termes *reconnaissance de documents* ou *conversion rétrospective*, ce dernier faisant référence à la reprise de documents papier anciens.

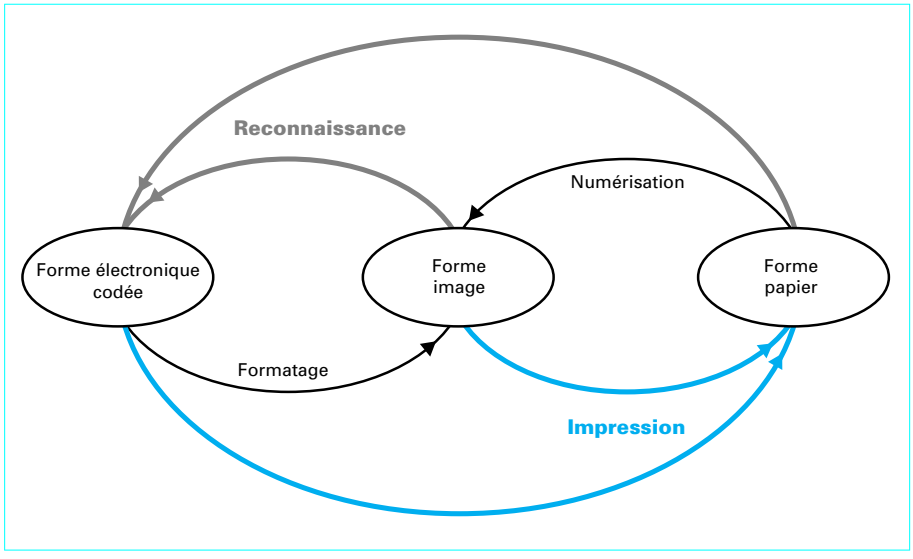

Figure 1 – Les formes des documents et la reconnaissance

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

1.6 Avantages des formes électroniques codées par rapport aux formes image

Les avantages des formes électroniques codées sont multiples :

Compacité

Une page de 3 000 caractères occupe sous forme codée un peu plus de 3 000 octets, alors que son image numérique, selon sa nature et le type de compression, a une taille de 60 kilooctets à 2 mégaoctets, voire plus pour des images à niveaux de gris ou en couleurs. Ce gain, d'un facteur 20 à 1 000, prend toute son importance lorsqu'il s'agit de stocker et surtout de transmettre à distance les documents (coûts et délais de transmission).

Modification facile du contenu

Cet avantage est encore plus important que le précédent. Le fichier texte d'une forme électronique codée peut être corrigé : tous les traitements de texte génèrent des documents dont le format a été conçu pour permettre leurs modifications. Par contre, un texte sous forme image est une entité figée : on ne peut que le consulter de façon passive, et les possibilités de modifications sont minimes.

Accès potentiel à la sémantique du texte

Seule la forme électronique codée permet l'accès au contenu (potentiellement au sens) du texte, à l'aide des techniques de traitement automatique du langage naturel (TALN) : recherche d'informations, analyse syntaxique des phrases, mise en correspondance d'informations, *text mining*, extraction de terminologies, classement et classification, diffusion ciblée, représentation de la signification du texte, résumé automatique et traduction assistée...

Structuration de l'information et hypertextes

Bien plus, les documents électroniques codés peuvent être structurés, c'est-à-dire organisés selon un plan précis, avec des renvois. Il devient ainsi possible de les consulter non plus linéairement, mais selon leur logique interne, et d'y « naviguer » lorsqu'ils sont munis de liens hypertextuels, comme les documents HTML.

En résumé, seule la représentation codée supporte des documents « vivants » et permet d'accéder véritablement à leur contenu. Cela tient au fait que pour les ordinateurs, elle donne accès aux informations signifiantes, alors que la forme image n'est interprétable que par les humains.

1.7 Applications

Actuellement, les applications de la reconnaissance de documents se multiplient [2], et on assiste à une véritable explosion de ce marché. Le développement d'Internet et des échanges d'informations sur les réseaux informatiques n'y sont pas étrangers. Nous avons classé les applications en deux catégories, selon qu'elles utilisent des textes structurés ou non.

1.7.1 Textes non structurés

1.7.1.1 Reprise de documents pour traitement de texte et réédition

Que ce soit en bureautique ou dans le monde de l'édition, on a souvent besoin de reprendre des documents qu'on possède uniquement sous forme papier, pour les rééditer ou insérer une partie de leur contenu dans un nouveau document. Dans ce cas, soit un texte structuré n'est pas nécessaire, soit la structure est celle du document cible et non pas celle du document source. Une reconnaissance de structure est donc inutile.

1.7.1.2 Édition de textes sur CD-ROM

Le problème est analogue, avec la nuance qu'un ensemble de textes sur CD-ROM est organisé. Là encore, l'organisation est propre au nouveau support créé et à l'application informatique associée, et la reconnaissance de la structure des documents à reprendre n'est pas forcément nécessaire.

1.7.1.3 Constitution de bases documentaires en texte intégral

Dans tous les domaines (presse, techniques, droit, médecine...) on crée aujourd'hui des bases de données documentaires en texte intégral, interrogeables à l'aide de moteurs de recherche. Pour retrouver des informations, une indexation préalable sur les mots du texte est nécessaire. Les techniques de reconnaissance de documents sont indispensables pour alimenter ces bases à partir de fonds papier anciens.

1.7.1.4 Diffusion sélective d'informations et diffusion ciblée

Les nouvelles techniques de TALN permettent de mettre en correspondance des documents et des utilisateurs décrits par un profil. Un tel profil se présente sous la forme d'un ensemble de mots ou de descripteurs définissant un domaine d'intérêt. Les documents à diffuser, s'ils sont sous forme papier, doivent au préalable être reconnus par OCR, pour que les logiciels de mise en correspondance, qui travaillent sur le texte intégral, puissent s'appliquer.

1.7.1.5 Traduction assistée par ordinateur (TAO)

L'OCR est un préalable indispensable à l'utilisation d'outils de TAO, pour tous les cas où les textes sous forme électronique ne sont pas directement disponibles.

1.7.2 Textes structurés

1.7.2.1 Revues de presse électroniques

Les revues de presse actuelles, constituées de photocopies d'articles distribuées sous forme papier, vont progressivement être remplacées par des équivalents électroniques, diffusés par messagerie ou mis à la disposition des clients dans des bases documentaires d'information. Beaucoup de journaux et revues ne sont encore disponibles que sous forme papier : la reconnaissance est donc une étape obligatoire. La prise en compte d'un minimum de structuration est ici nécessaire : pour chaque article, il faut distinguer le nom de l'auteur, le titre de l'article, les en-têtes et chapeaux, le corps du texte, les légendes...

1.7.2.2 Conversion rétrospective des notices bibliographiques

Les notices bibliographiques sont des fiches cartonnées utilisées par les bibliothèques pour répertorier les ouvrages qu'elles possèdent. Elles portent des informations, appelées *métadonnées*, telles que titre, auteur, éditeur, année de publication, nombre d'exemplaires, lieu de stockage... Ces fiches sont aujourd'hui remplacées par des fichiers numériques structurés conformes à la norme UNIMARC. Le besoin de reconnaissance de structure est évident dans ce cas.

1.7.2.3 Serveurs Minitel, serveurs *on-line* et serveurs W3

Il existe plusieurs types de serveurs consultables à distance, accessibles par le réseau téléphonique, *via* un minitel ou un modem, ou par Internet. Tous mettent à disposition des textes dont certains (petites annonces...) sont déjà disponibles sous forme imprimée. Le niveau de structuration nécessaire dépend fortement de l'application. Par exemple, les serveurs W3 mettent à disposition de l'information fortement structurée.

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

1.7.2.4 Constitution de documentations techniques structurées

La documentation technique fournie lors de la livraison ou de l'homologation de systèmes complexes (avions, trains, centrales nucléaires, médicaments...) va être de plus en plus soumise à des contraintes de qualité, incluant en particulier la structuration des documents selon des normes précises. Jusqu'alors, ces documents étaient déjà structurés, mais fournis uniquement sous forme papier. Une bonne partie d'entre eux est réutilisée dans les nouveaux projets. Un travail énorme de conversion rétrospective devra être réalisé pendant la période de transition. Cela met en jeu des millions de pages, dans diverses industries : nucléaire, aéronautique et spatiale, laboratoires pharmaceutiques...

2. Documents imprimés : contenu et structure

2.1 Mise en page

Les pages constituent les unités physiques de base des documents papier, que ce soit pour la composition ou la lecture. Les dimensions acceptées par les logiciels de reconnaissance sont au maximum les formats *letter* ou A4 en bureautique, exceptionnellement le format A3 pour certains systèmes industriels. Ces pages sont constituées d'un fond blanc ou de couleur, sur lequel sont imprimées les informations utiles. En général, des marges sont laissées entre le bord de la page et le cadre qui entoure ces informations. La mise en page obéit à des règles typographiques précises.

2.2 Zones présentes dans un document

Les informations imprimées sont de plusieurs natures, qui correspondent la plupart du temps à des zones distinctes dans les pages — mais on peut aussi avoir imbrication ou recouvrement de ces zones.

Texte

Il constitue la partie la plus importante dans le cadre de cet article. L'information textuelle est par nature biniveau.

Dessins, graphiques et schémas

Il s'agit de tous les éléments graphiques constitués majoritairement de traits et de points : dessins, schémas, plans, etc. La plupart, même s'ils incluent des couleurs, peuvent se réduire à une information biniveau.

Tableaux

Ils sont constitués d'informations textuelles ou numériques présentées en lignes et en colonnes. Leur structure est régulière ou non, avec ou sans présence de filets qui délimitent les cellules contenant l'information élémentaire. L'information est biniveau.

Formules mathématiques

Elles se distinguent du texte par la présence d'opérateurs arithmétiques, algébriques, logiques... ainsi que d'indices ou exposants, et l'utilisation de caractères tirés de l'alphabet grec, en plus des chiffres et des caractères latins. La nature de cette information est biniveau.

Photographies

Elles sont caractérisées par le fait qu'elles incluent des plages de couleur ou des dégradés de gris. En imprimerie, la plupart des photographies sont restituées par un tramage de points noirs ou colorés (§ 4.1), souvent suffisamment fin pour être invisible à l'œil nu. L'information photographique est donc selon les cas biniveau (tramée), à niveaux de gris, ou en couleurs.

2.3 Notions de typographie

Un texte est composé de caractères (majuscules, minuscules, chiffres, signes de ponctuation et quelques symboles), regroupés en lignes et en paragraphes ou blocs. Ces caractères ont des formes et des tailles variées, qui diffèrent beaucoup d'une famille à une autre, et ils sont définis par des caractéristiques précises (§ 5.1) [3] [6].

Fonte ou police

C'est l'ensemble ou l'assortiment complet des dessins des caractères d'une même famille, c'est-à-dire présentant une unité de style et de taille. Le mot *fonte* est employé en typographie, alors qu'on utilisait plutôt le mot *police* en dactylographie. Les deux termes ne sont pas totalement synonymes, car une police correspond à une taille déterminée.

Corps

Il s'agit de la taille d'une fonte, déterminée par la hauteur totale des caractères les plus grands, plus les blancs de séparation avec les lignes supérieure et inférieure. Cette hauteur est mesurée entre le haut des majuscules et des minuscules à hampe montante (l, d, k) et le bas des minuscules à jambage (q, p, j). Le corps s'exprime en une unité ancienne : le *point Didot*, qui vaut 0,3759 mm ; dans les pays anglo-saxons on utilise le *point anglais*, égal à 0,351 mm. En imprimerie, les corps des caractères d'une même fonte varient dans de grandes proportions (6 à 100). Un texte dactylographié habituel utilise des caractères de corps 9 à 12.

Chasse ou échappement

C'est la largeur d'un caractère, plus l'espace entre caractères. À l'origine, la chasse désignait la largeur du bloc métallique portant le caractère en relief. Sur les machines à écrire classiques, chaque caractère était inscrit dans un rectangle de largeur constante, avec comme conséquence une chasse fixe qui facilitait l'OCR puisque, une fois le début du mot trouvé, il suffisait de se déplacer de la valeur de la chasse pour cadrer chaque caractère. On employait alors le terme d'échappement, pour désigner l'entraxe entre caractères, équivalant à la chasse dans ce cas. Celui-ci est exprimé en nombre de caractères par pouce : les valeurs standards sont 10, 12, et 15. La chasse est dite variable, ou l'échappement proportionnel, lorsque les caractères ne s'inscrivent pas dans un rectangle de largeur constante : un « m » ou un « w » sont plus larges qu'un « l » ou un « i ». C'est toujours le cas pour l'imprimé, et les traitements de texte récents, car cela donne un confort de lecture bien supérieur. Par contre, la séparation des caractères est rendue plus difficile.

Style

Ce terme désigne le type des caractères, qui présentent une grande diversité au niveau des proportions, de la graisse, de l'inclinaison de l'axe vertical, de l'empattement... (§ 5.1).

Alignement, justification

L'alignement vertical des débuts et fins de ligne définit plusieurs types de composition [3]. Le texte est dit *justifié* lorsque les débuts et fins de ligne sont alignés verticalement, ce qui est obtenu en jouant sur l'espace entre les mots. Dans ce cas, la composition est dite *en alinéa*, *en sommaire* ou *au carré*, selon que la première ligne du paragraphe est en retrait, dépasse, ou est alignée avec les autres. Le texte est dit *en drapeau* lorsqu'il est aligné d'un seul côté, à droite ou à gauche. Il est enfin dit *centré* lorsqu'on aligne les milieux des lignes successives, de longueur variable.

2.4 Structure physique et structure logique

Un document papier peut être considéré selon deux points de vue différents [4] :

— sa mise en page, sa présentation visuelle, qui correspond à la structure physique ;

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

— l'organisation logique de son contenu, qui correspond à la structure logique.

Le passage de la structure logique à la structure physique est appelé **formatage** ; il dépend des dimensions des pages, et précède ou est réalisé pendant l'impression (figure **1**). Ces notions sont bien définies dans la norme d'architecture de documents ODA (*open document architecture*) [5].

2.4.1 Structure physique

Un document est constitué de composants physiques à structure répétitive. Par exemple, il peut être constitué de plusieurs ensembles de pages, le premier comprenant la page de garde et le sommaire, le second le corps du texte, le troisième les annexes et index. Un ensemble de pages comporte des sous-ensembles et des pages individuelles. Dans une page, on définit une surface rectangulaire utile, appelée gabarit, elle-même subdivisée en pavés. Chaque pavé contient une information de nature unique : texte, graphique, photo... On décrit ainsi un document par une première arborescence, basée sur les pages, qui a comme feuilles terminales les contenus des pavés d'informations homogènes.

2.4.2 Structure logique

Un document est constitué de composants logiques à structure répétitive. Par exemple, il est organisé en chapitres, eux-mêmes divisés en paragraphes. Ceux-ci peuvent à leur tour contenir des sous-paragraphes, des figures, des tableaux... Il est donc possible de le décrire par une deuxième arborescence, basée sur son plan, qui a comme feuilles terminales les unités logiques d'information. Les feuilles terminales des deux arbres ne se correspondent qu'en partie : un paragraphe qui constitue un tout d'un point de vue logique peut en effet être à cheval sur plusieurs colonnes ou plusieurs pages.

2.4.3 Macrostructure et microstructure

Les structures considérées précédemment, inspirées de la norme ODA, concernent l'ensemble du document ou des pages. C'est ce qu'on appelle la *macrostructure*. Pour certaines applications, il est nécessaire de prendre en compte une structuration plus fine, dite *microstructure*, qu'on peut mettre en évidence au niveau des paragraphes, ou fragments de contenu. Par exemple, dans un paragraphe constitué d'énumérations introduites par des tirets, il faut respecter la structuration interne du texte en items.

Exemple : l'application la plus évidente de cette notion est la reconnaissance des notices bibliographiques, constituées d'une succession de portions de texte ayant une signification précise : titre, auteur, éditeur, mentions diverses... Ces éléments doivent être distingués pour être retranscrits dans la norme UNIMARC qui en donne une représentation structurée.

3. Composantes d'un système de reconnaissance

La figure **2** indique la suite des traitements qui doivent être effectués pour aboutir à la reconnaissance complète d'une page [2] [10].

3.1 Acquisition ou numérisation

La première opération consiste à acquérir l'image numérique de la page avec un scanner. Cependant, les documents à traiter sont de contenu et de qualité divers ; il est donc quelquefois nécessaire d'acquérir les images en niveaux de gris (§ 4.1). Pour éviter l'encombrement de la mémoire, des images biniveaux sont préférables. Les résolutions utilisées pour l'OCR sont au minimum de 300 dpi (*dots per inch*), voire 400 dpi pour les petits caractères (§ 4.2).

3.2 Redressement

Les scanners d'exploitation ont un débit de 60 à 120 pages à la minute. Les feuilles, lors de leur défilement rapide sur la glace d'exposition, sont susceptibles de s'incliner de plusieurs degrés par rapport à la direction de balayage des images. Pour une bonne exploitation des documents, il faut détecter cet angle et redresser les images, de telle sorte que les lignes de texte soient parallèles aux bords (§ 4.3). Certains scanners réalisent cette opération, d'autres non.

3.3 Binarisation

La binarisation consiste à transformer une image à niveaux de gris en une image biniveau, et est la plupart du temps réalisée électroniquement dans les scanners.

Elle entraîne dans certains cas une perte importante d'informations, et selon la manière dont elle est effectuée (§ 4.4), l'image biniveau résultante est exploitable ou non pour des traitements ultérieurs. Notons que les photos doivent être traitées différemment du texte (§ 4.1).

3.4 Segmentation des pages

On dispose maintenant en mémoire d'une image biniveau. Avant de passer à une analyse plus fine, il est nécessaire de donner au système l'équivalent de la vision globale d'un lecteur humain. Au niveau des pages, on appelle *segmentation* (§ 4.5) l'opération d'analyse globale qui consiste à détecter et à localiser les zones distinctes : textes, graphismes par traits, tableaux, photos... À l'issue de cette opération, on soumet les différentes zones à des traitements spécifiques. En particulier, les blocs contenant du texte sont envoyés à un module de reconnaissance de caractères (OCR).

3.5 Reconnaissance des caractères

Après la phase de repérage des zones de texte, il faut identifier les caractères, les coder et les regrouper en mots : c'est en cela que consiste la reconnaissance de caractères (§ 5.4 à 5.12). Cette opération fait appel à des techniques diverses ; la complexité du problème provient de la grande diversité des formes et des tailles des caractères imprimés (§ 5.1), ainsi que des défauts d'impression et de numérisation (§ 5.2).

3.6 Reconnaissance de la structure logique

La plupart du temps, on se contente de récupérer le texte « au kilomètre ». Il existe cependant des applications (exemple des revues de presse, des notices bibliographiques...) dans lesquelles on alimente une base de données à partir du texte reconnu. Dans ce cas, certaines parties du document d'origine vont jouer un rôle particulier dans l'alimentation de la base : titre de l'article, chapeau, nom de l'auteur, date de parution, etc. Il est alors important de connaître la nature ou le rôle de l'information, en plus de son contenu. Cela nécessite la reconnaissance de la structure logique du document. Elle est fondée sur la reconnaissance d'éléments de mise en page, eux-mêmes en relation avec cette structure logique.

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

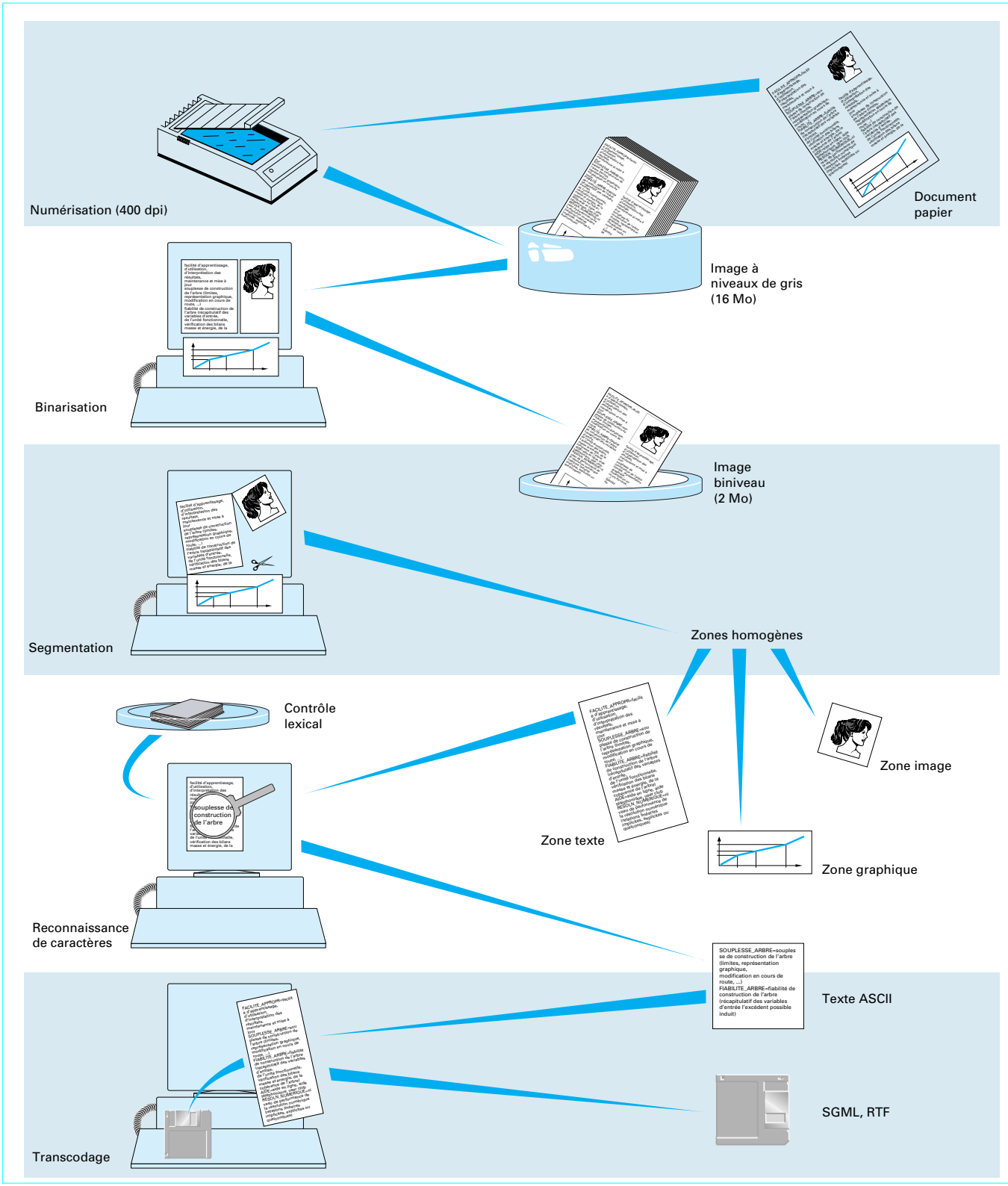

Figure 2 – Les étapes de la reconnaissance de documents

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

3.7 Transcodage

Il ne suffit pas de reconnaître un document : encore faut-il en donner une traduction dans un format adéquat, exploitable par la ou les applications qui vont utiliser ce document. Par exemple, si on reconnaît un texte pour l'insérer dans un document en cours de création avec un traitement de texte d'usage courant, on va souhaiter le récupérer dans un format RTF, qui s'importe facilement dans MS-Word, Wordperfect ou tout autre traitement de texte ou logiciel de PAO. Dans ce cas précis, on préfère généralement récupérer le texte « au kilomètre », car la mise en page va dépendre plutôt du nouveau document que de l'ancien.

4. Traitements préliminaires

4.1 Variété et encombrement des images de documents

On a vu que le type d'image couramment utilisé en GED est une représentation biniveau des documents. Elle présente l'avantage d'occuper une place mémoire relativement réduite : 1,2 mégaoctets en 300 dpi, 2 mégaoctets en 400 dpi. Grâce aux méthodes de compression de l'information (CCITT Groupe 3 et Groupe 4), ces valeurs sont divisées par un facteur 10 à 20 — on obtient ainsi des images de pages dont la taille est comprise entre 60 et 250 kilooctets.

Ce codage biniveau est adéquat pour le texte et les schémas, mais pas pour les photos. Pour les traduire correctement, un codage en niveaux de gris ou en couleurs est nécessaire : à chaque point élémentaire sont associés soit une information de luminance sur 8 bits (d'où 256 valeurs de niveau de gris possibles), soit trois informations de couleurs (3×8 bits).

En fait, les images des documents ne sont pas stockées sous cette forme, car cela occupe trop de place en mémoire : 16 mégaoctets pour une page A4 en 256 niveaux à 16 points au mm (400 dpi), 29 mégaoctets pour une page couleur à 12 points au mm (300 dpi). En outre, cela présenterait peu d'intérêt car l'œil humain a une perception moins fine des couleurs que des contrastes. La solution dans ce cas est donnée par le tramage : on simule les niveaux de gris par une densité variable de pixels noirs sur fond blanc (figure **3**). Une image biniveau peut donc contenir à la fois du texte, des schémas et des photos tramées.

4.2 Acquisition des images et choix des modes de travail

Les logiciels d'OCR opèrent la plupart du temps à partir d'images biniveaux, pour les raisons d'encombrement mémoire expliquées précédemment. La reconnaissance de caractères impose des contraintes sur la résolution. Une numérisation à 300 dpi (environ 12 points/mm) suffit pour les documents contenant des caractères de corps 10 et plus ; en dessous du corps 10, une résolution de 400 dpi (environ 16 points/mm) est nécessaire. Néanmoins, pour la tâche plus large de reconnaissance de documents, des images à niveaux de gris sont nécessaires dans les deux cas suivants :

— les documents contiennent des photos qu'on souhaite également récupérer. Comme les scanners ne sont pas capables de distinguer automatiquement les photos, la solution est d'acquérir les images en niveaux de gris (ou en couleurs) et de réaliser la séparation des photos puis leur tramage, par logiciel ;

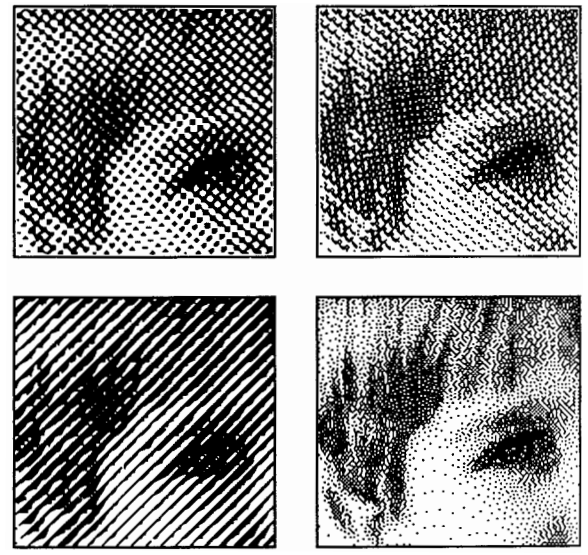

Figure 3 – Exemples de tramage

— la qualité d'impression du texte est variable au sein d'une même page, ou le texte est imprimé sur un fond non homogène ou de plusieurs couleurs : seul un traitement logiciel permet alors de récupérer une image biniveau correcte de l'ensemble du texte.

Avant l'acquisition, une analyse préalable du contenu du document en fonction de sa destination est donc nécessaire pour définir la résolution et le type des images (biniveau ou niveaux de gris).

Le logiciel permettra donc de choisir le type et la résolution des images nécessaires à une analyse correcte, ainsi que la suite des traitements à appliquer. Il permettra aussi de définir le fonctionnement global : feuille à feuille (acquisition puis reconnaissance de chaque page avant de passer à la suivante) ou par lots (acquisition et stockage des images de l'ensemble des pages, puis lancement de la reconnaissance en traitement différé sur celles-ci).

4.3 Redressement

Pour améliorer la qualité de la segmentation et de l'OCR, il est souhaitable que l'image du document soit parfaitement droite : cela facilite notamment la recherche des colonnes de texte dans le cas où deux colonnes consécutives sont très proches l'une de l'autre. Il faut donc détecter l'**angle de déviation globale** de la page, en anglais *skew*, puis effectuer une rotation inverse de la valeur trouvée. Cette technique permet également de détecter les pages au format *paysage*, aussi dites *à l'italienne*. Le plus difficile dans cette tâche est la détection précise de l'angle d'inclinaison.

Nota : on s'intéresse seulement aux cas où il n'y a qu'une inclinaison globale à détecter. Les cas de déviations multiples au sein d'une même page ne sont pas pris en compte : bloc de texte volontairement imprimé en biais par rapport aux autres ou déviation des lignes sur une photocopie de livre dans la zone proche de la reliure.

Plusieurs algorithmes ont été mis au point pour détecter l'angle d'inclinaison du texte. Citons les méthodes de Trincklin [11], de Baird [12] et de Postl [13].

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

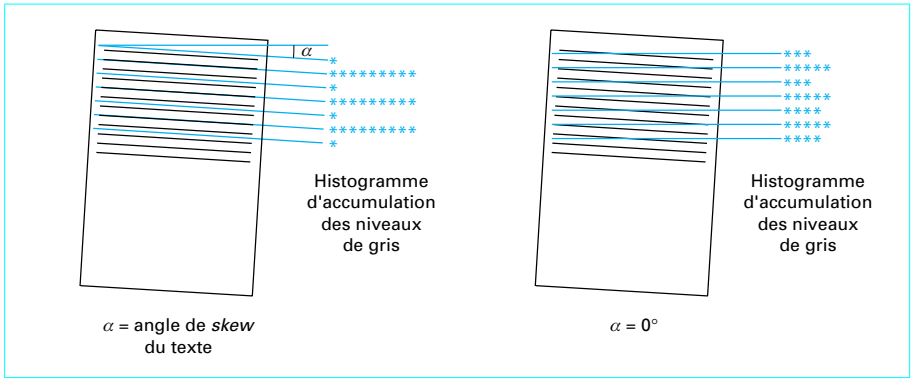

Figure 4 – Analyse de la page pour deux valeurs de l'angle α

Exemple : nous allons décrire rapidement l'**algorithme de Postl** [13], qui s'applique à des images à niveaux de gris ou biniveau. La méthode opère à partir d'un sous-échantillonnage de 1/*N* en hauteur et de 1/*P* en largeur, et consiste à :

— tracer virtuellement sur l'image *N* lignes parallèles équidistantes faisant un angle α avec l'horizontale ;

— calculer le long de chacune de ces *N* lignes la somme *S* des niveaux de gris rencontrés en prenant 1 pixel sur *P* seulement ;

— calculer sur l'ensemble des *N* lignes la somme *P* (pour *Premium*) des carrés des différences des sommes *S* d'une ligne à la suivante :

$$P = \sum_{i=1}^{N} (S_{\text{ligne } i} - S_{\text{ligne } i+1})^2$$

— faire varier l'angle α, et trouver la valeur pour laquelle cette valeur *P* passe par un maximum.

Cela revient à cumuler les projections de pixels le long d'une ligne inclinée et à chercher l'angle pour lequel ces accumulations se mélangent le moins quand on passe d'une ligne à la suivante (figure **4**).

Pour trouver l'angle α qui correspond au maximum, on travaille par exemple sur une plage [-10^o, $+10^o$], en plusieurs passages. Un pas initial de 2^o permet de trouver l'inclinaison approximative, puis on diminue le pas en même temps que la plage de scrutation, pour finir avec un pas de $0{,}1^o$ égal à la précision souhaitée.

4.4 Binarisation

Réalisée électroniquement dans les scanners, elle est souvent réduite à l'application d'un **seuil ajustable**, mais unique pour une page donnée. Tous les pixels plus lumineux que le seuil sont pris comme blancs, les autres comme noirs. Pour des documents bien contrastés et homogènes, cette méthode rudimentaire permet d'obtenir des résultats acceptables. Elle est par contre tout à fait insuffisante dès que la qualité des documents est médiocre. En outre, si la plupart des scanners possèdent des possibilités de tramage pour l'acquisition des photos, cette fonction doit être sélectionnée par un opérateur. En effet, seuls quelques rares scanners haut de gamme sont capables de localiser automatiquement les photos dans une page, d'utiliser des algorithmes de tramage localement et d'appliquer d'autres algorithmes au reste de la page constitué de texte et de graphismes.

Plusieurs étapes fonctionnelles sont ainsi définies pour la binarisation :

— restituer correctement le texte, les tableaux et les graphiques quelles que soient les variations de contraste et de couleur du fond ;

— détecter, localiser et tramer les photos ;

— détecter et signaler les encarts colorés.

La première fonction est la plus importante par rapport à l'OCR.

4.4.1 Seuillage global

Le seuillage global consiste à prendre un seuil, ajustable, mais identique pour toute l'image. Pour une échelle à 256 niveaux de gris, cette valeur est comprise entre 0 et 255, généralement dans la plage médiane. Chaque pixel est comparé à ce seuil : en polarité directe, ceux de niveaux de gris inférieurs sont mis à « blanc » (0), ceux supérieurs mis à « noir » (1).

La difficulté réside dans le choix du seuil. Si le seuillage est fait par le scanner, un opérateur peut effectuer un réglage en fonction de l'aspect visuel du document et du résultat d'une première acquisition. Pour le seuillage automatique d'une image à niveaux de gris [14], une méthode consiste à tracer un histogramme de ces niveaux de gris, et choisir le seuil au fond de la vallée qui sépare le pic correspondant au niveau de gris du fond et le premier pic suivant (voir figure **5**, image en polarité inverse).

Cette méthode convient pour les documents simples et de bonne qualité. Les limites en sont les suivantes :

— lorsque la qualité d'impression du texte n'est pas constante dans toute la page, des caractères peuvent être partiellement perdus (figure **6**) ;

— lorsque le fond est bruité ou non homogène, des taches parasites peuvent apparaître ;

— lorsque la page contient des encarts de couleur (texte imprimé sur fond coloré), tout le texte de ces blocs risque d'être perdu ;

— les photos traitées de cette manière ne sont plus reconnaissables : elles sont rendues par une mosaïque de zones noires et blanches.

La solution à cet ensemble de problèmes est donnée par le seuillage adaptatif d'une part, la séparation des photos d'autre part.

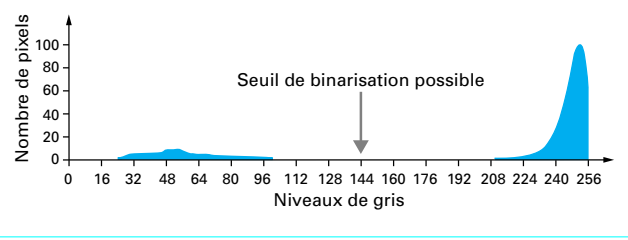

Figure 5 – Choix du seuil sur l'histogramme de l'image à niveaux de gris

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

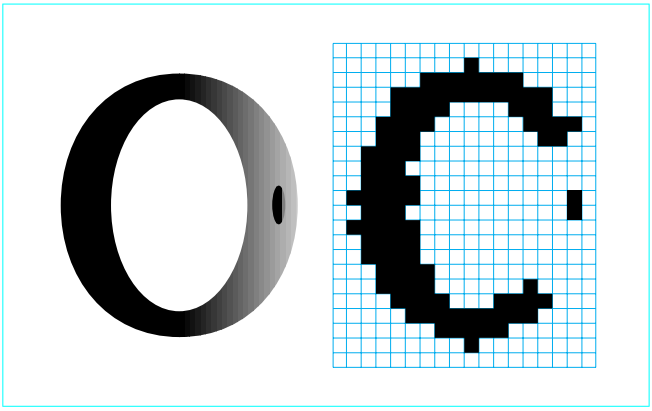

Figure 6 – Effet du seuillage sur un caractère mal imprimé

4.4.2 Seuillage adaptatif

Le seuillage adaptatif consiste à faire varier le seuil localement, en fonction des niveaux de gris de l'information (texte) et du fond. Plusieurs solutions de seuillage adaptatif ont été proposées pour restituer correctement le texte dans tous les cas [14]. On distingue des méthodes basées sur un découpage de l'image en régions et des méthodes locales proches du traitement du signal.

4.4.2.1 Découpage de l'image en régions

Une première approche consiste à découper l'image en un quadrillage de petites zones rectangulaires, à tracer un histogramme des niveaux de gris dans chaque rectangle et à calculer un seuil adapté à chacun. Les inconvénients de ce genre de méthode sont multiples : nécessité d'avoir la totalité de l'image en mémoire, temps de calcul très longs, difficulté d'ajustement du seuil aux frontières des zones, et surtout non-adéquation *a priori* du quadrillage à l'information de la page. En effet, si un rectangle du quadrillage se trouve à cheval sur une frontière de couleur du fond de l'image, une des deux parties sera mal binarisée.

4.4.2.2 Méthodes locales proches du traitement du signal

Un autre type de solution consiste à déterminer le seuil de binarisation de chaque pixel à partir de propriétés locales du voisinage de ce pixel. Cela revient à utiliser une fenêtre glissante, et appliquer des opérateurs locaux, linéaires de type convolution ou non linéaires. Cette solution pallie certains manques de l'approche précédente comme la fixité de la grille.

Exemple : une méthode [15] consiste à détecter par un calcul de gradients les frontières des objets (les caractères) et à remplir leur intérieur en noir. La difficulté réside ici dans le choix des dimensions du voisinage de chaque pixel et du critère de remplissage, qui dépendent de la taille des caractères, inconnue *a priori*.

4.4.2.3 Opérateur laplacien avec apprentissage de seuil

Une approche plus efficace est celle de l'opérateur laplacien avec apprentissage de seuil [16] [17] [18]. On applique un opérateur laplacien, qui correspond à la dérivée seconde de la variation du niveau de gris autour de chaque point de l'image analysée (figure **7**).

Un **exemple d'opérateur laplacien**, dit *gaussien*, est donné par la formule suivante : soient X le pixel testé et a à h les pixels voisins les plus proches, on a :

$$LAPL = (a + 2b + c + 2d - 12X + 2e + f + 2g + h)/16$$

	a	b	c	
	d	X	e	
	f	g	h	

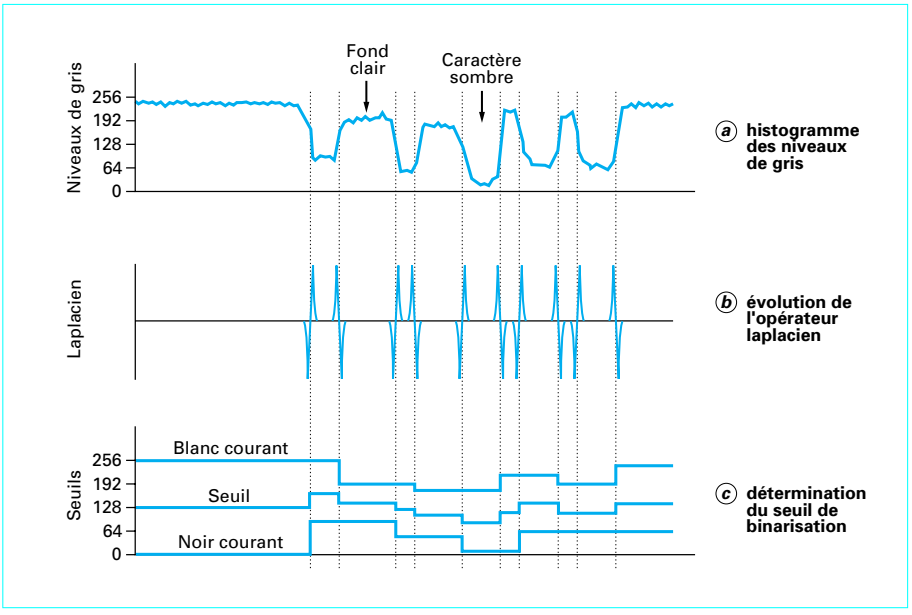

Figure 7 – Évolution de l'opérateur laplacien et des seuils (polarité inverse)

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

On utilise l'opérateur laplacien pour détecter les points appartenant à des caractères ou à des traits, dès qu'un contraste suffisant apparaît entre un pixel et ses voisins. Les niveaux de gris de l'image qui correspondent d'une part au noir des caractères et d'autre part au blanc du fond, sont mémorisés et utilisés pour calculer le seuil de binarisation (moyenne). Pour les points dont le contraste n'est pas significatif, on utilise les valeurs courantes de ces niveaux sans les remettre à jour. La figure ***7c*** montre les niveaux de gris représentant le noir et le blanc courants (courbes N et B). Ces deux niveaux sont initialisés arbitrairement en début de ligne à 0 et 255, puis mis à jour en fonction de l'évolution du résultat de l'opérateur laplacien. Le seuil de binarisation est la moyenne de ces deux niveaux et évolue à chaque mise à jour de l'un ou de l'autre (courbe « seuil »).

4.4.3 Séparation des encarts colorés et des photos

Les techniques précédentes permettent d'extraire le texte quelles que soient les variations de son contraste et du fond, mais ne permettent pas la localisation des encarts colorés ni des photos. Pour cela, une approche consiste à découper l'image en régions et à calculer des seuils locaux, comme décrit au paragraphe 4.4.2.1. Un classement des seuils permet de déterminer le plus bas : on l'utilise pour obtenir une nouvelle image biniveau de la page qui restitue les encarts colorés et les photos sous forme de pavés noirs. Après remplissage de ceux-ci et filtrage des petits objets, on obtient des masques sous forme de gros objets qui définissent l'enveloppe des encarts colorés et des photos. Une analyse de texture à l'intérieur de chaque gros objet connexe permet de différencier les photos des encarts colorés.

4.5 Segmentation des pages à partir du texte

La segmentation a pour fonction de distinguer les diverses composantes d'un document. Cela recouvre plusieurs problèmes :

— délimiter les caractères, les mots et les lignes de texte (trouver leur enveloppe rectangulaire) ;

— distinguer et délimiter les grandes zones d'information dans une page : texte, dessins et graphismes au trait, formules mathématiques, tableaux, photos ; c'est ce qu'on appelle la segmentation des pages.

C'est ce deuxième problème que nous allons étudier. Trois types de méthodes existent pour réaliser cette opération : méthodes descendantes, ascendantes et mixtes [19].

4.5.1 Méthodes de segmentation descendantes

Elles consistent à partager le document en grandes régions, qui sont à leur tour divisées en sous-régions. Nous examinerons les méthodes de remplissage et celles de projection récursive.

Nota : ces méthodes ne fonctionnent correctement que sur un document qui a été préalablement redressé.

4.5.1.1 Méthodes de remplissage

Un exemple caractéristique est la méthode de Wong, Casey et Wahl, qui utilise un algorithme appelé RLSA (*run length smoothing algorithm*) [20]. Elle consiste à construire une nouvelle image biniveau en balayant horizontalement et verticalement l'image originale, et à noircir les plages blanches de longueur inférieure à un seuil donné. Cette technique permet de mettre en évidence les blocs d'images qui ont de nombreuses transitions entre le blanc et le noir (donc le texte), en agglomérant les plages noires voisines. La variation du seuil permet d'agglomérer des objets plus ou moins espacés et de déterminer le découpage du texte en lignes et en colonnes.

La limitation de ce type de méthode est double : d'une part, son fonctionnement correct dépend étroitement des seuils utilisés, qui présupposent la connaissance de la taille des caractères et des espaces intercaractères et interlignes ; d'autre part, on constate souvent que l'espace entre deux mots au sein d'une même ligne est supérieur à l'espace intercolonnes. Il y a donc un risque de réunir des colonnes adjacentes. On y pallie en alternant judicieusement les balayages horizontaux et verticaux ou en calculant le ET logique des résultats successifs. Après le découpage de la page en zones, celles-ci sont analysées pour déterminer leur contenu : texte, graphiques, etc. Une des méthodes consiste à réaliser des mesures statistiques de texture sur les plages blanches et noires.

4.5.1.2 Méthodes de projection récursive

Elles sont basées sur une analyse macroscopique de l'image du document et consistent à projeter successivement celle-ci selon les directions horizontale et verticale, pour mettre en évidence les grandes « cheminées » et les interlignes blancs [11] [21]. Après une découpe réussie dans un sens, horizontal ou vertical, on réitère l'opération dans l'autre sens sur les morceaux obtenus. Le résultat est une description hiérarchique de la page sous forme d'un arbre *XY*. Le succès de la segmentation est basé sur une stratégie correcte de récursion. En particulier, selon la mise en page, le sens de la première découpe doit être horizontal ou vertical.

Les limites de ce genre de méthodes sont dues au fait qu'il existe des structures de page suffisamment imbriquées pour ne pas permettre une découpe récursive (figure **8**). Par ailleurs, elle suppose que tous les blocs qui constituent la page sont rectangulaires, ce qui n'est évidemment pas toujours le cas.

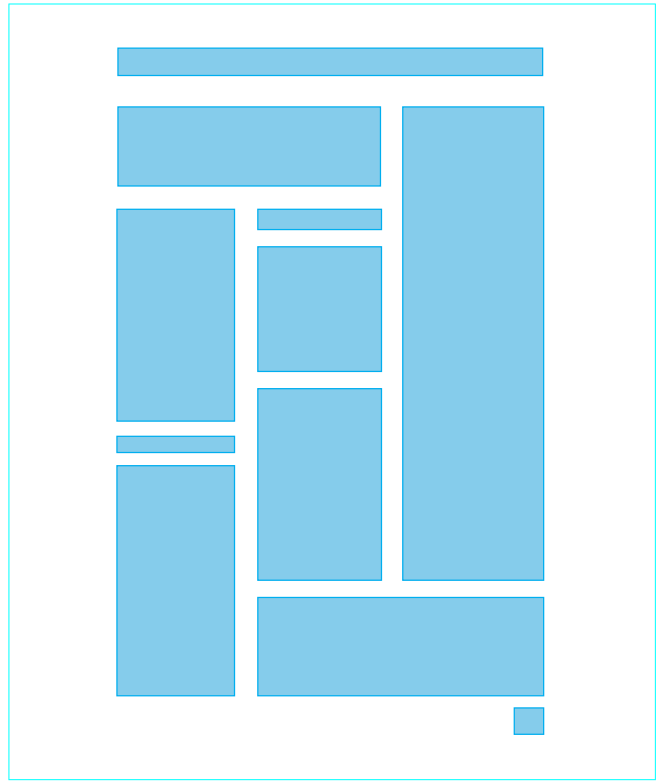

Figure 8 – Exemple de structure non découpable récursivement

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

4.5.2 Méthodes de segmentation ascendantes

Les méthodes de segmentation ascendantes, au contraire, partent des objets élémentaires d'une image pour reconstituer des groupements d'objets. Les objets élémentaires sont ce qu'on appelle les composantes connexes : groupements de pixels noirs séparés les uns des autres (caractères, ponctuation...). Leur premier avantage par rapport au traitement des pixels tient au facteur de réduction considérable du nombre d'objets à manipuler : une page dense contient 5 000 composantes connexes — son image 2 millions de pixels ! On peut regrouper les composantes connexes extraites en structures de plus en plus élaborées : mots, groupes de mots, lignes, paragraphes...

L'avantage de cette approche est qu'elle s'adapte à toute mise en page, même incluant des blocs non rectangulaires ou imbriqués, et qu'elle tolère une certaine inclinaison des pages. Son inconvénient tient à la nécessité d'une connaissance *a priori* de la taille des caractères, à une complexité de mise en œuvre et au besoin d'utiliser en complément une méthode projective (descendante) pour éviter d'agglomérer des colonnes de texte proches. On aboutit ainsi aux méthodes mixtes.

4.5.3 Méthodes mixtes

Dans la pratique, pour être capable de segmenter correctement des structures de pages complexes, il faut utiliser des méthodes mixtes.

Exemple : méthode ascendante-mixte : cette méthode a été mise au point en 1988 [22], et a été utilisée depuis avec succès dans plusieurs applications industrielles, dont le logiciel PRASAD. Elle est basée sur le constat que le texte d'un document dactylographié ou imprimé est caractérisé par une structure régulière et répétitive, organisée en plusieurs niveaux : mot, ligne, paragraphe, colonne.

La méthode prend en compte deux niveaux de regroupement : horizontal puis vertical. On agglomère d'abord les objets élémentaires en lignes ou portions de lignes, puis celles-ci en paragraphes et colonnes. En final, toute portion de l'image qui est structurée ainsi est considérée comme du texte et le reste de la page comme des zones graphiques. On décompose le processus de reconstruction de la structure du texte de la page en six étapes.

1 - Recherche et classement des objets élémentaires

Un automate d'états finis, basé sur l'étude des transitions des pixels noirs d'une ligne sur l'autre, permet l'extraction des objets connexes de l'image en un seul balayage. On récupère les coordonnées de leurs rectangles englobants. En fonction de leurs dimensions, taux de remplissage... les objets sont répartis en cinq classes : petits objets, ponctuation, caractères, caractères longs (plusieurs caractères collés, ou un caractère qui coupe un soulignement), autres objets.

2 - Agglomération des objets de type caractère

À partir d'un caractère courant, on teste l'existence d'un caractère suivant, situé à droite du premier, à distance proche et de taille homogène. Si c'est le cas, on établit un lien entre les deux caractères et on réitère l'opération à partir du suivant. Un problème majeur de l'agglomération est d'éviter de relier le dernier caractère d'une ligne d'une colonne au premier caractère d'une ligne au même niveau dans la colonne suivante. On utilise pour cela un contrôle projectif vertical, qui relève d'une approche descendante. À l'issue de ces traitements, on obtient des chaînes de caractères potentiels (figure **9**).

3 - Validation des lignes

Pour que ces chaînes soient validées en tant que lignes, elles doivent obéir à certains critères : horizontalité, rectitude, homogénéité des hauteurs de caractères, séparation de l'environnement. Les contrôles peuvent entraîner l'acceptation de la ligne, son rejet ou son découpage en deux ou plusieurs portions obéissant aux critères précédemment énoncés. Un contrôle projectif horizontal permet cette fois de vérifier que la ligne est correctement séparée de son environnement. Il obéit également à une démarche descendante.

4 - Regroupement des lignes en blocs

La mise en évidence des colonnes et le regroupement des lignes en blocs sont ensuite effectués. On commence par rechercher les alignements de débuts et de fins de ligne, puis on partitionne les groupes de lignes selon un critère de proximité. Les lignes sont étiquetées de manière non exclusive comme début ou fin de colonne. Cela permet d'identifier les colonnes de la page. Il faut ensuite fixer les limites d'extension de ces colonnes et en déduire les blocs : on établit des liaisons entre lignes successives qui sont comprises entre deux colonnes ou ont un certain taux de recouvrement vertical. La structure résultante est un graphe orienté, où chaque nœud représente une ligne, et chaque arc une liaison entre deux lignes.

5 - Découpage en paragraphes et validation des blocs de texte

À partir des graphes de lignes ainsi formés, on réalise des tests d'homogénéité, et on ne conserve que les liens entre les lignes de hauteur homogène, et qui correspondent à des espaces interlignes également homogènes. Après élimination des liaisons non valides, on obtient des graphes connexes, chacun représentant un bloc. À ce stade, les blocs et les paragraphes ont donc été identifiés. Après quelques tests complémentaires, ils sont définis par les enveloppes des lignes associées dans un même graphe.

6 - Ordonnancement des blocs de texte

Les blocs doivent être numérotés selon un ordre de lecture vraisemblable par rapport à la topologie trouvée. Une solution consiste à utiliser une méthode semblable à la projection récursive décrite au paragraphe **4.5.1.2** : on projette les blocs trouvés alternativement en *X* et *Y*, et on observe si une partition apparaît clairement lors d'une des projections. On réitère l'opération en sens inverse sur chacun des sous-ensembles trouvés, et ainsi de suite. L'ordre de numérotation choisi *a priori* est haut-bas et gauche-droite. En toute rigueur, l'utilisation de règles adaptées à la mise en page du document est nécessaire.

4.6 Détection et localisation des tableaux

Un tableau est constitué d'un ensemble de cellules rectangulaires regroupées dans une zone du document et liées les unes aux autres par des relations de voisinage plus ou moins hiérarchiques. Chaque cellule possède un seul type de contenu : textuel ou numérique. On définit plusieurs types de tableaux, selon que leur maillage est régulier ou non, que le tableau est complètement, partiellement ou pas du tout encadré, et complètement, partiellement ou pas du tout cloisonné. Encadrements et cloisonnements sont plus fréquents dans les documents européens que dans les documents américains.

La première étape du traitement consiste à détecter et localiser un tableau dans une page, puis à déterminer sa topologie en cellules : c'est ce qui correspond pour les tableaux à la phase de segmentation. Peu de travaux ont été réalisés sur ce sujet [23] [24]. Il est possible de procéder par détection des cadres et filets, par juxtaposition de blocs, ou en combinant les deux méthodes.

Figure 9 – Visualisation de l'agglomération d'objets caractères

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

4.6.1 Détection des filets

Une première approche consiste à voir un tableau comme un maillage de segments de droite. On isole ainsi dans un premier temps une zone potentielle de tableau par extraction de grands objets connexes noirs. Ensuite, une détection des segments de droite et une recherche de leurs intersections permet de valider la zone et de délimiter les cellules du tableau. Cette approche n'est évidemment applicable que sur les tableaux possédant un cadre et un cloisonnement. Il faut noter que les filets qui délimitent les cellules des tableaux peuvent être continus, pointillés ou tiretés.

4.6.2 Juxtaposition de blocs

Une autre approche consiste à considérer un tableau comme un assemblage régulier ou hiérarchique de blocs de contenu. On commence par rechercher sur toute l'image des blocs homogènes de texte (on peut utiliser la méthode définie au paragraphe 4.5.3 ou un algorithme de type RLSA). Un ensemble de blocs voisins et disposés selon une structure régulière et/ou hiérarchique définit un tableau. Cette méthode est la seule possible lorsqu'on traite des tableaux non encadrés ou non complètement cloisonnés.

4.6.3 Méthode mixte

Comme toujours en reconnaissance de documents, un résultat correct sur une grande variété de cas n'est obtenu qu'en associant plusieurs approches. Une méthode de segmentation des tableaux mise au point en 1995 pour le logiciel *PRASAD* [25] procède en trois étapes.

① **Recherche grossière des zones** potentielles de tableaux, définies par les grands objets connexes noirs, les ensembles de filets parallèles et les groupements de blocs de texte disposés régulièrement ou hiérarchiquement. Les segments de droite sont détectés par des projections horizontales et verticales, et par un algorithme de suivi prédictif, indispensable pour les filets pointillés ou tiretés.

② **Validation de ces zones** potentielles comme tableaux réels : vérification de la quantité de segments de droite et de blocs régulièrement disposés, et du taux de remplissage des cellules. L'approche par blocs est intéressante pour valider les tableaux dont toutes les cellules ne sont pas délimitées par des filets.

③ **Délimitation des cellules** du tableau par les intersections des segments de droite et la position des blocs homogènes de texte. Les deux approches se complètent pour obtenir une meilleure précision.

4.7 Détection et localisation des formules mathématiques

Les quelques travaux menés sur le sujet se sont intéressés à la reconnaissance de formules mathématiques supposées identifiées comme telles, mais pas à leur détection. À notre connaissance, aucun logiciel prototype, et encore moins commercial, ne réalise cette fonction. La difficulté de la détection des formules mathématiques tient à ce qu'elles ne présentent pas de particularité structurelle systématique. Par exemple, une équation à structure linéaire de type $ax + by + c = 0$ ne se singularise pas par rapport au texte. Les principales caractéristiques des formules mathématiques sont :

— la présence d'opérateurs arithmétiques, algébriques, logiques, etc. ;
— la présence d'indices et d'exposants ;
— la présence de caractères grecs et de chiffres ;
— la présence de symboles dont la taille dépend de leur contenu ou de leur suite (racines, intégrales) ;
— la présence de barres de fraction, de crochets de matrices...

La détection et l'extraction des formules mathématiques de leur environnement passent donc par la reconnaissance préalable de certains caractères particuliers : symboles arithmétiques, algébriques, caractères grecs, etc. Il s'agit donc d'un problème particulier de reconnaissance de caractères (§ 5.4), avec une difficulté supplémentaire du fait de la présence de symboles de taille variable.

4.8 Détection et localisation des graphiques

Les zones graphiques comportent des figures, schémas, dessins au trait, graphes, organigrammes... Aucune structure caractéristique ne permet de les distinguer *a priori*, si ce n'est la présence de segments de droites, définissant un cadre ou des axes, ou constituant des parties de dessin. Cela même n'a rien de systématique. On définira donc un graphique par la négative : un graphique est une zone biniveau contenant une information qui n'est ni du texte, ni un tableau, ni une formule mathématique. Il est possible d'envisager des traitements qui localisent les zones graphiques potentielles (par recherche de droites, analyse de texture), mais seule la capacité de détection et localisation des tableaux et des formules mathématiques permettra de distinguer ces deux types d'objets des graphiques.

5. Reconnaissance des caractères (OCR)

Rappelons que dans cet article, on se limite à la reconnaissance des caractères dactylographiés ou imprimés de l'alphabet latin.

5.1 Structure des caractères et variété

Plusieurs éléments de style peuvent différer d'une fonte à l'autre, ce qui explique les multiples variétés existantes [6]. Citons la graisse, la stature (rapport hauteur-largeur), l'italique (inclinaison), le contraste entre les pleins et les déliés, l'aplatissement des rondeurs, la présence ou l'absence d'empattements, la nuance anglaise (arrondissement des empattements). La classification Vox, retenue par l'Association typographique internationale, regroupe les fontes en une dizaine de classes (figure **10**).

Quelques valeurs numériques sont nécessaires pour situer l'ampleur du problème : en dactylographie, on recense facilement entre les divers constructeurs plus de 200 polices différentes, chacune dans un seul corps ; en imprimerie, le catalogue d'un seul fournisseur, comme la Monotype, comporte plus de 8 000 fontes, avec tous les corps possibles pour chacune (échelle de 6 à 100).

5.2 Déformations dues à l'impression, la reproduction, la numérisation

En fait, les logiciels de reconnaissance sont confrontés à des millions de formes possibles, car aux variantes de fontes viennent s'ajouter les différences dues aux qualités d'**impression** et de **reproduction** des documents. Une impression ou une photocopie de mauvaise qualité engendrent des caractères collés, coupés, ou du bruit de fond. Il faut tenir compte en plus des dégradations dues à l'acquisition : problèmes d'échantillonnage (résolution), de numérisation et de seuillage déjà évoqués. Cela accentue les phénomènes de déformation, collage ou cassure des caractères (figure **11**).

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

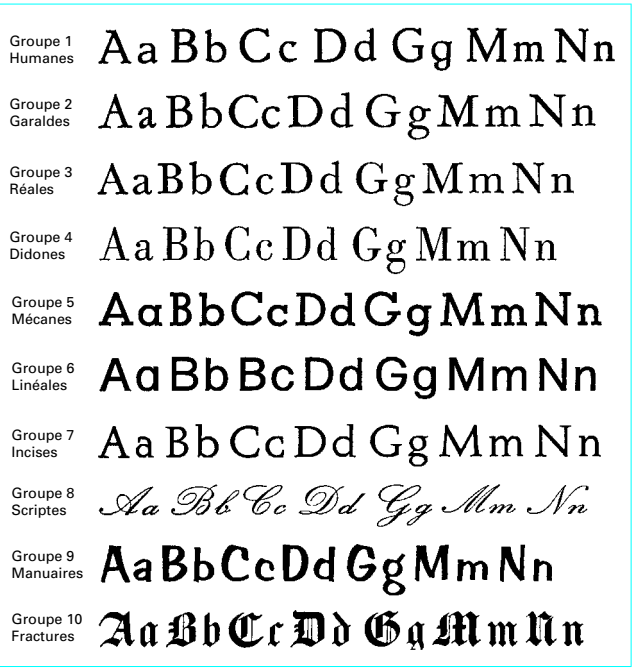

Figure 10 – Exemples de la classification Vox

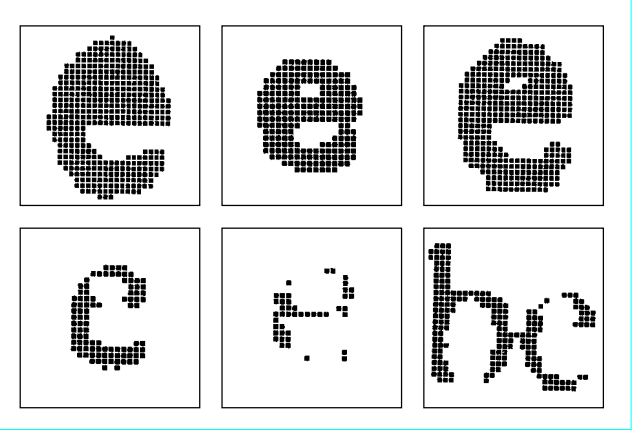

Figure 11 – Exemples d'images de caractères rencontrées

5.3 Erreurs de reconnaissance et mesure des performances

Les défauts sur les caractères engendrent pour les logiciels d'OCR des erreurs de reconnaissance. L'évaluation de la qualité de ces logiciels nécessite de caractériser puis d'évaluer quantitativement les erreurs.

5.3.1 Types d'erreurs

On classe les erreurs en quatre catégories :

— **rejet** : un caractère n'est pas reconnu, le logiciel le détecte cependant et le signale par un code spécial : $, #. C'est le cas le moins grave ;

— **substitution** : un caractère est pris pour un autre. C'est le cas le plus ennuyeux ;

— **omission** : un caractère est perdu (cas des caractères collés) ;

— **ajout** : il y a un caractère supplémentaire. Dans ce cas, une tache est prise pour un caractère, ou un caractère est coupé en deux.

Les erreurs d'omission et d'ajout sont le plus souvent liées à une substitution ; par exemple, un « m » coupé est reconnu comme « rn » : substitution + ajout, ou « ri » collés vont être reconnus comme « n » : omission + substitution. Les cas les plus fréquents de substitutions ont lieu par exemple entre les lettres « I », « l », et le chiffre « 1 » ou entre la lettre « O » et le chiffre « 0 », qui ne se distinguent que par le contexte.

5.3.2 Taux d'erreur

On définit le **nombre total d'erreurs** de reconnaissance comme la somme des erreurs de chaque type :

$$Err = Rejet + Substitution + Omission + Ajout$$

Si le texte reconnu comporte N caractères, le **taux d'erreur** est simplement le rapport des deux nombres :

$$Taux\ d'erreur = \frac{Err}{N}$$

On définit de même le **taux de rejet** et le **taux de confusion** :

$$Taux\ de\ rejet = \frac{Rejet}{N}$$

$$Taux\ de\ confusion = \frac{Substitution + Omission + Ajout}{N}$$

Le **taux de reconnaissance** est le complément à 1 du taux d'erreur :

$$Taux\ de\ reconnaissance = \frac{N - Err}{N}$$

Comme ordre de grandeur pour ces taux, il faut savoir qu'au-dessus de 1 à 1,5 % de taux d'erreur, un logiciel d'OCR est considéré comme difficilement exploitable. Pour obtenir l'erreur sur les mots, il faut en effet multiplier le taux d'erreur sur les caractères par un facteur proche de 7 (en considérant qu'un mot comporte en moyenne 7 caractères) : un taux d'erreur de 1,5 % implique donc qu'un relecteur doit corriger 1 mot sur 10 dans le texte, ce qui devient vite impraticable. Le taux d'erreur d'un logiciel de reconnaissance n'est pas affichable *a priori* : il dépend de la qualité du document reconnu. Des logiciels récents comme *Fine Reader* ou *PRASAD* ont des taux d'erreurs de 0,2 % ou moins sur des documents originaux de bonne qualité.

5.4 Définition de la reconnaissance de caractères

Avant de poursuivre, il est nécessaire de préciser la notion de « reconnaissance de caractères ». Ce terme recouvre deux concepts.

Le premier que nous désignerons par **reconnaissance de bas niveau**, en anglais *pattern matching*, consiste à décider de l'identité quasi point à point, d'une forme-échantillon avec une forme-modèle prédéfinie (gabarit) [4]. Par exemple la forme #, constituée de quatre traits parallèles deux à deux, peut servir de modèle. Si maintenant, nous retrouvons ce même signe # imprimé dans un document, nous le reconnaissons, c'est-à-dire nous constatons son identité visuelle avec le signe imprimé sur la ligne précédente.

Le deuxième que nous désignerons par **reconnaissance symbolique** consiste à analyser une forme-échantillon, pour lui attribuer une étiquette, un code ou un nom symbolique, qui caractérise un modèle générique et abstrait. Ce problème est beaucoup plus complexe car un modèle symbolique correspond à une multitude de formes-modèles qui se distinguent les unes des autres par des

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

variantes topologiques et métriques. Par exemple, les formes suivantes :

f, f, ***f***, f correspondent toutes à la sixième lettre de l'alphabet, et sont nommées « caractère f », bien qu'elles soient graphiquement assez différentes, par leur taille comme par d'autres caractéristiques.

Les logiciels d'OCR qui se sont succédé sur le marché pendant les 30 dernières années ont évolué d'une reconnaissance de bas niveau vers une reconnaissance symbolique.

5.5 Processus mis en œuvre

La reconnaissance, qu'elle soit de bas niveau ou symbolique, est une tâche complexe qui recouvre plusieurs étapes quasiment identiques dans les deux cas. Globalement, il s'agit de comparer des caractères-échantillons en entrée, à des modèles (formes-modèles ou modèles symboliques) décrivant des classes prédéfinies, et de décider à laquelle des classes existantes l'échantillon appartient. C'est donc un problème de classement.

5.5.1 Étapes

En détaillant les processus mis en œuvre, il est possible d'identifier trois grandes étapes.

① **Segmentation des caractères**

Il faut au préalable découper les lignes et les mots en caractères isolés ; cette étape pourrait sembler indépendante de la reconnaissance. En fait il n'en est rien, car les logiciels sont confrontés à des images de caractères collés ou coupés en plusieurs morceaux. Cela impose une interaction entre segmentation et reconnaissance : par exemple, une détection de mauvaise segmentation au moment de la reconnaissance, puis des essais successifs de reconnaissance avec des segmentations différentes.

② **Extraction des caractéristiques**

L'extraction de caractéristiques ou primitives, en anglais, *feature extraction*, transforme la *bitmap* d'origine, caractère ou forme-modèle, en une description numérique ou symbolique dans un espace abstrait, selon un formalisme prédéfini. La description sera évidemment beaucoup plus simple pour une reconnaissance de bas niveau que pour une reconnaissance symbolique. Dans le premier cas, par exemple, on pourra se contenter de détecter les intersections du caractère avec une grille fixe, dans le deuxième on pourra décrire avec un véritable langage les segments, concavités et boucles du caractère analysé. Dans le premier cas l'image du caractère est transformée en un vecteur de valeurs numériques, dans le deuxième en une suite de symboles.

③ **Décision ou reconnaissance proprement dite**

L'étape de décision est assimilable à une tâche de classement dans un espace correspondant aux caractéristiques choisies. Rappelons que le *classement* est le rangement dans des classes déjà définies. On dispose donc en mémoire d'un ensemble de classes, décrites par des modèles : formes-modèles ou modèles symboliques. À partir de la description d'un caractère-échantillon, il faut définir à quelle classe cet échantillon appartient. Plusieurs résultats sont possibles :

— le caractère-échantillon ne correspond à aucune classe existante : c'est le phénomène du rejet déjà vu ; dans ce cas on décide ou non de refaire une segmentation du caractère et de son voisin ;

— le caractère-échantillon correspond à une classe unique ; si les classes ont été correctement définies, c'est le succès : le caractère est correctement reconnu ;

— le caractère-échantillon correspond à plusieurs classes : on va alors choisir celle qui correspond au meilleur score mais c'est typiquement le cas où une substitution est possible ; on peut également décider de refaire une segmentation ;

— le caractère-échantillon est détecté d'emblée comme mal segmenté : on va alors le segmenter à nouveau avant de tenter une nouvelle reconnaissance.

5.5.2 Synthèse

La figure **12** synthétise ces trois étapes qui sont globalement identiques pour la reconnaissance de bas niveau ou symbolique.

Certains logiciels mettent en œuvre successivement les deux types de reconnaissance, ce qui donne un modèle global plus complexe : il est possible en effet, soit d'effectuer la reconnaissance symbolique d'un caractère directement après sa segmentation, soit de commencer par constituer une bibliothèque de formes-modèles des caractères présents dans la page analysée (reconnaissance de bas niveau), puis de réaliser la reconnaissance symbolique des formes-modèles.

5.5.3 L'apprentissage, préalable à la reconnaissance

Cette tâche de reconnaissance n'est possible que parce que le logiciel d'OCR dispose en mémoire d'un ensemble de classes prédéfinies, décrites par des modèles. Pour cela, un travail d'apprentissage doit avoir été réalisé au préalable. Il faut décider, manuellement ou automatiquement, quelles sont les caractéristiques discriminantes, mettre au point des modèles descriptifs, traiter les données d'une base d'apprentissage, définir des classes disjointes. Les techniques mises en œuvre pour l'apprentissage sont multiples : modélisation, apprentissage supervisé, classification automatique ou assistée. La *classification*, contrairement au classement, consiste à organiser des données brutes en un ensemble de classes homogènes et disjointes, donc à définir des classes et des méthodes de classement. Les approches sont variées : depuis l'analyse intellectuelle et la construction d'un arbre de décision « à la main », jusqu'aux réseaux de neurones à apprentissage quasi automatique.

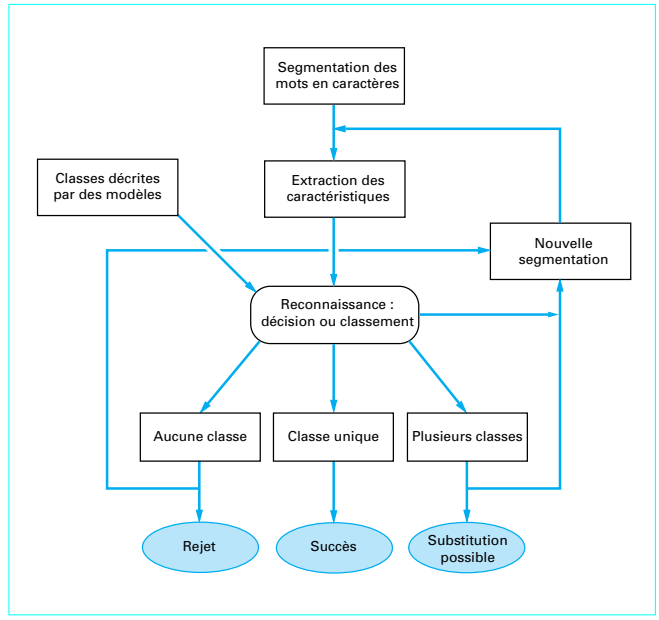

Figure 12 – Processus mis en œuvre dans la reconnaissance de caractères

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

5.5.4 Chronologie des étapes de fonctionnement d'un logiciel d'OCR

Nous résumerons ainsi la chronologie et la logique des étapes de constitution et fonctionnement d'un logiciel de reconnaissance :

— choix manuel ou automatique (par exemple dans le cas d'un réseau de neurones) de caractéristiques qui permettent de modéliser les caractères, par des formes-modèles proches des images, ou des modèles symboliques ;

— apprentissage sur une base de référence, qui nécessite une classification, manuelle ou automatique, des images des caractères de la base ; celles-ci sont modélisées, puis réparties en classes homogènes, et enfin labellées ;

— reconnaissance de caractères : prise de décision, qui consiste en un classement des échantillons de caractères par rapport aux classes définies lors de l'apprentissage.

5.6 Les trois générations de logiciels d'OCR. Typologie des méthodes

Les générations successives de lecteurs optiques correspondent à une mise en œuvre progressivement plus complète et sophistiquée de toutes ces fonctions. La première génération de lecteurs dits monofontes (années 1960 et 1970) n'était capable de reconnaître que certaines fontes prédéfinies, en particulier les fontes OCR-A et OCR-B normalisées pour l'OCR. Puis dans les années 1980 sont apparus des systèmes plus souples, dits multifontes à apprentissage, qui permettaient l'apprentissage de tout type de fonte à partir d'un document, par interaction avec un opérateur. Des exemples de tels systèmes furent les machines Kurzweil (États-Unis), puis les logiciels de la famille *Readstar* de la société française Inovatic. Tous ces systèmes mettaient en œuvre une reconnaissance de bas niveau à partir de formes-modèles figées ou apprises. Ce n'est qu'à partir des années 1990 qu'on a vu apparaître les logiciels modernes capables de reconnaissance symbolique et d'une relative universalité, mis au point par des sociétés comme Caere-Calera, Kurzweil-Xerox, Mimétics...

5.6.1 Logiciels monofontes et *pattern matching*

Les logiciels et systèmes monofontes étaient basés sur des techniques dites de *pattern matching* ou *template matching*, permettant une reconnaissance de bas niveau figée, qui détecte l'identité de deux objets, ou d'un objet à reconnaître avec une forme-modèle définie au niveau des pixels. On a vu que cette opération se ramène à un problème de classement, dont la résolution passe par la définition d'une distance entre objets, ou entre objet et modèle.

À cette étape de traitement, la comparaison se fait point à point, et il faut définir une distance entre images de caractères. La distance la plus connue pour la comparaison d'objets *bitmap* est la distance dite de Hamming. Elle est définie, pour deux objets X et Y composés de pixels d'indice i, par :

$$d(X, Y) = \sum_i |x_i - y_i|$$

Pour des images biniveaux, elle s'obtient en mettant en correspondance les deux images, puis en réalisant un OU exclusif : le nombre de pixels noirs du résultat donne la distance de Hamming.

On peut également calculer le coefficient de corrélation entre deux images biniveaux, à partir du ET logique des images : si Nc est le nombre de pixels résultant et S la surface totale en nombre de pixels (moyenne ou maximum des deux en cas de légère différence), le coefficient de corrélation K est donné par la formule :

$$K = \frac{Nc}{S}$$

Cette approche, la première qui se présente à l'esprit, est en fait limitée [7], comme on le voit sur l'exemple de la figure **13** : le « S » empâté est mieux corrélé avec un « 8 » qu'avec un « S » de référence.

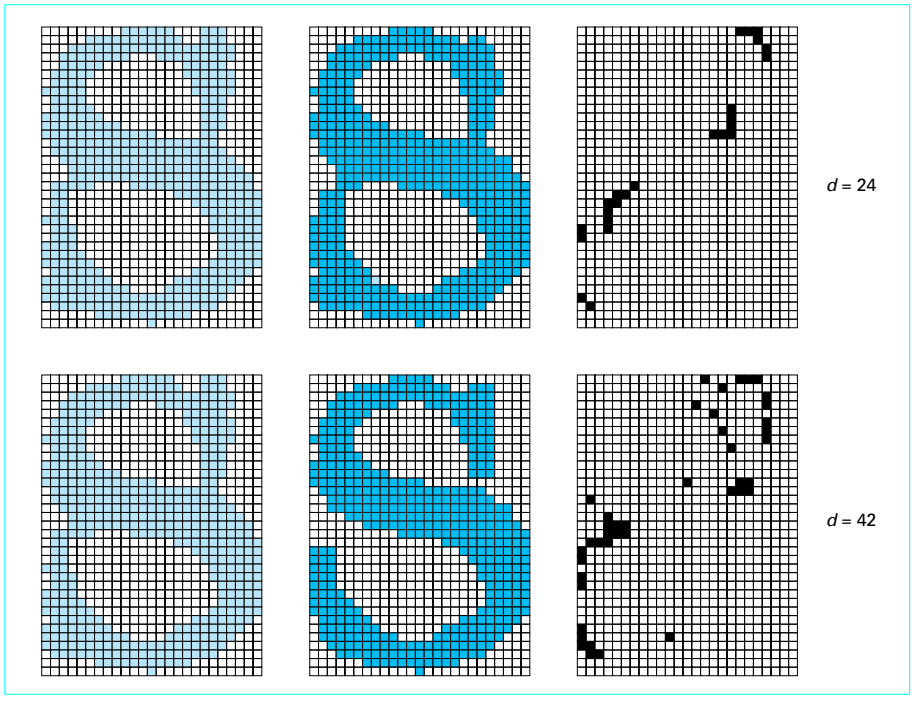

Figure 13 – Distance de Hamming entre plusieurs caractères

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

Tableau 1 – Typologie des méthodes de reconnaissance

Décision	Extraction des caractéristiques							
	Méthodes de bas niveau				Méthodes structurelles			
	Comparaison globale	Masques partiels	Descripteurs géométriques	Intersection de droites	Grille ajustable	Projections partielles	Analyse des zones blanches	Description topologique
Arbre de décision	I	I						
Discrimination fonctionnelle linéaire	I	I						
Méthodes bayesiennes			II	II	II			
Programmation dynamique et automates			II	II	III	III	III	III
Réseaux de neurones			II	II	III	III	III	III
Modèles de Markov							III	III

I 1re génération.
II 2e génération.
III 3e génération.

Pour lever ce genre de difficultés, une solution consiste à utiliser des distances plus élaborées, qui tiennent compte à la fois de la métrique et de la topologie des caractères. Par exemple, on pondérera les pixels en fonction de leur distance à une frontière de l'objet.

La phase d'apprentissage dans ce cas est implicite et réduite à sa plus simple expression : elle consiste à établir la correspondance, une fois pour toutes, entre une forme précise et un code ASCII.

5.6.2 Logiciels multifontes et apprentissage

On s'attaque maintenant à une autre catégorie de problèmes : les caractères à reconnaître appartiennent à des fontes multiples, et une même classe finale — celle du caractère « a » par exemple — correspond à plusieurs formes-modèles distinctes, non prédéfinies. Les logiciels à apprentissage ne possèdent pas de modèles préalablement mémorisés, et la phase d'apprentissage fait partie intégrante de l'exploitation. Dans un tel contexte, les caractères sont *a priori* inconnus, et il faut commencer par définir des classes, avant de chercher à les nommer. On a vu que ce qui correspond à la phase d'apprentissage est typiquement un problème de classification. Le logiciel extrait les caractères successifs de l'image, et il commence à constituer des modèles de référence en comparant ces caractères deux à deux. Les images semblables sont regroupées dans une même forme-modèle, qui correspond à un masque utilisé pour les calculs de distance. Une base de modèles est créée et les images correspondantes des masques sont présentées à l'opérateur sur écran, avec éventuellement leur contexte. L'apprentissage est donc supervisé par l'opérateur qui étiquette le masque en entrant le code ou la suite de codes correspondants. Il faut bien noter qu'ici encore, la tâche de reconnaissance symbolique — la désignation de la forme — est manuelle. Lorsque la base de modèles est suffisante, la tâche d'OCR commence. Dans cette seconde phase, le logiciel effectue la dernière étape définie au paragraphe 5.5.1 : le travail de classement.

5.6.3 Logiciels omnifontes sans apprentissage

Les logiciels actuels sont dits omnifontes sans apprentissage, c'est-à-dire qu'ils sont capables de reconnaître la quasi-totalité des caractères latins dactylographiés et imprimés, sans interaction avec un utilisateur. Ils sont donc capables de **reconnaissance symbolique**. Cela suppose que le travail d'apprentissage (modélisation symbolique et classification des caractères d'une base d'apprentissage) a été réalisé auparavant. On l'a vu, cette reconnaissance symbolique peut être effectuée directement sur les caractères extraits du texte, ou sur les formes-modèles obtenus par une reconnaissance de bas niveau. Dans ce cas, le logiciel doit réaliser successivement : la segmentation des caractères, la création de formes-modèles en regroupant les caractères semblables, l'extraction de leurs caractéristiques, leur classement pour mettre les formes-modèles en correspondance avec les modèles symboliques existants, et leur étiquetage (attribution d'un code ou d'une suite de codes ASCII ou autres).

5.6.4 Typologie des méthodes

Le tableau **1** présente une typologie des méthodes de reconnaissance, fondée sur l'extraction des caractéristiques et la décision.

Les principales approches passées en revue aux paragraphes 5.7 et 5.8 sont citées, et on voit apparaître les trois générations de logiciels que nous avons distinguées.

5.7 Extraction des caractéristiques et primitives

Les caractéristiques choisies pour décrire ou quantifier les caractères sont très nombreuses. On en trouve un inventaire dans [26] [27]. Nous nous contenterons d'en citer quelques-unes parmi les plus représentatives. Il faut noter qu'elles ne sont pas totalement indépendantes des méthodes de décision décrites au paragraphe 5.8, ni forcément exclusives les unes des autres. On les répartit en deux grandes catégories : les caractéristiques de bas niveau, de type global, qui correspondent à des mesures numériques ou booléennes, et les caractéristiques de niveau symbolique, dites aussi primitives, plutôt de type structurel, obtenues par une analyse du caractère en formes élémentaires [28] [29] [30].

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

5.7.1 Caractéristiques pour la reconnaissance de bas niveau

On cherche dans ce cas à déterminer la correspondance d'un caractère-échantillon avec des modèles de bas niveau, proches des images.

5.7.1.1 Comparaison globale point à point

L'idée la plus simple consiste à faire une comparaison point à point entre des images de caractères-échantillons et des masques (§ 5.6.1). On définit plusieurs types de masques :

— binaires : bitmaps du même type que les formes-échantillons ;
— ternaires : définis par une enveloppe ou zone d'extension maximale dans laquelle tous les pixels de l'échantillon doivent être inclus, et un noyau ou zone de réduction extrême, qui doit être entièrement contenu dans l'échantillon ;
— statistiques : chaque pixel du modèle se voit attribué un poids en fonction de l'importance de sa présence pour corréler les échantillons.

La notion d'extraction de caractéristiques est limite pour ces comparaisons globales. Néanmoins, le calcul d'une distance ou d'un coefficient de corrélation est assimilable à une mesure, dont le résultat est une caractéristique. Si on n'utilise pas d'autre primitive en complément, il faut comparer l'échantillon à l'ensemble des masques pour déterminer avec lequel il coïncide le mieux.

5.7.1.2 Comparaison à des masques partiels

On peut, au contraire des masques globaux, utiliser des masques de petite taille par rapport aux caractères, et positionnés à des endroits fixes, pour détecter certaines particularités locales. La caractéristique correspondant à chaque masque partiel est une fonction booléenne ou numérique, qui indique la présence de la particularité testée. Ce genre d'approche qui permet uniquement une reconnaissance monofonte, est plutôt lié à une décision par un arbre déterministe : à chaque nœud de l'arbre, on teste une caractéristique particulière, qui permet d'éliminer plusieurs possibilités, et on atteint la solution, à l'extrémité de l'arbre, après un nombre réduit de tests.

5.7.1.3 Calcul de descripteurs géométriques

Certains descripteurs géométriques élémentaires permettent de réaliser un préclassement des images de caractères ; nous citerons, de manière non exhaustive :

— **l'élongation** : rapport longueur/hauteur ;
— **les moments** : pour un objet numérique biniveau représenté par la fonction $f(i, j)$, le moment d'ordre (p, q) est donné par la formule :

$$m_{pq} = \sum_i \sum_j x_i^p \, y_j^q \, f(i,j)$$

Le moment d'ordre 0 donne la surface totale (nombre de pixels noirs) de l'objet ; les moments d'ordre 1 permettent de trouver son centre de gravité, et les moments centrés d'ordre 2 permettent d'en définir les axes d'inertie... :

— **la compacité** : donnée par le rapport de la surface de l'objet au carré de son périmètre ;
— **le taux de remplissage** : rapport de la surface de l'objet à la surface de son rectangle englobant.

5.7.1.4 Intersection avec des droites horizontales ou verticales

Un autre groupe de primitives intéressantes est le décompte des intersections des caractères avec des droites parallèles, horizontales ou verticales (ou l'ensemble des deux), de position fixe par rapport aux bords du caractère. À partir de cela, soit on constitue des vecteurs qui ont pour composantes le nombre d'intersections avec chaque droite, soit on construit un arbre de décision comme au paragraphe 5.7.1.2.

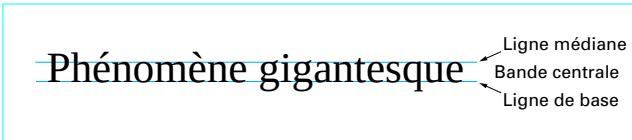

Figure 14 – Ligne de base et ligne médiane

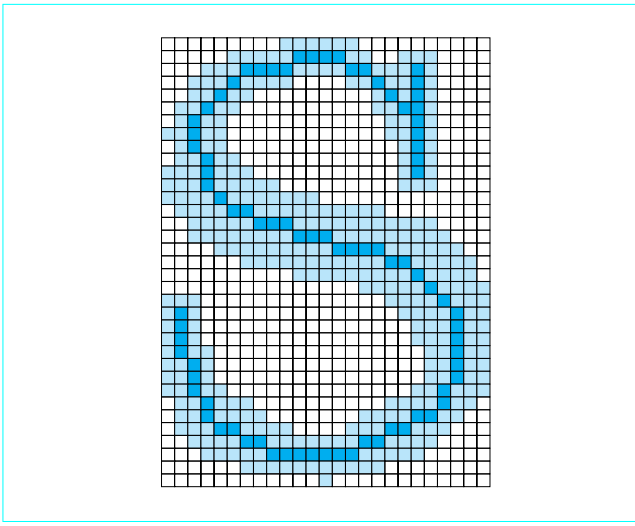

Figure 15 – Squelette d'un caractère

En particulier, il est intéressant d'observer la position du caractère par rapport à la ligne de base et à la ligne médiane : les caractères courts (a, e, u, s, r, n...) sont alignés horizontalement, et entièrement situés dans une bande délimitée par une ligne de base inférieure, et une ligne médiane, dite bande centrale (figure **14**) ; le positionnement des caractères par rapport à ces deux lignes permet de les répartir en quatre classes : majuscules et caractères à hampe, minuscules courtes, caractères à jambage, et caractères qui dépassent les deux lignes (parenthèses, « j », certains « f »...).

5.7.2 Caractéristiques structurelles pour la reconnaissance symbolique

Les caractéristiques structurelles, qui permettent d'identifier les caractères et de leur attribuer un code, doivent être le plus possible indépendantes des variations de style et de taille des fontes. Elles sont généralement constituées par des primitives de forme et la description de leurs relations.

5.7.2.1 Extraction de contour - squelettisation

Nota : ces opérations ne sont pas des extractions de caractéristiques à proprement parler, mais plutôt des traitements préliminaires qui permettent ensuite de déterminer certains types de primitives structurelles.

Les contours d'un caractère (externe et interne en cas de boucle) sont des courbes continues qui marquent ses frontières avec le fond. Le squelette est défini de manière intuitive comme suit : une courbe continue, de 1 pixel d'épaisseur, qui constitue le tracé des points équidistants des contours de l'objet (figure **15**).

Idéalement, il doit respecter les propriétés métriques et topologiques (même nombre de branches, de trous et de connexité) de la

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

forme. Nous n'entrerons pas dans le détail des algorithmes, extrêmement nombreux, qui permettent le suivi de contour ou la squelettisation [8].

La squelettisation est très utilisée pour la reconnaissance de l'écriture manuscrite, mais beaucoup moins pour celle des caractères imprimés : elle est très consommatrice en temps de calcul, et trop sensible aux distorsions des images de caractères décrites au paragraphe 5.2. De plus, certains caractères ne sont distinguables qu'en prenant en compte les variations d'épaisseur de leur tracé, qui disparaissent après squelettisation. Celle-ci peut néanmoins servir de base pour déterminer certaines caractéristiques structurelles comme les concavités, embranchements, etc.

5.7.2.2 Mesures par une grille ajustable

Une approche intermédiaire entre les caractéristiques de bas niveau et structurelles consiste à normaliser la représentation du caractère en décomposant son image selon une grille qui s'ajuste au rectangle englobant. Par exemple, pour les caractères qui s'inscrivent approximativement dans un carré (a, e, u, s, n...), on réalise une découpe par une grille de dimensions 16×16. Dans chaque pavé de la grille, on mesure la densité relative des pixels noirs. On obtient ainsi un vecteur de 256 composantes qui permet une description du caractère indépendante de sa taille. On utilisera des grilles avec plus de cellules pour les caractères d'élongation différente : 32×16 pour les caractères longs, 16×32 pour les caractères hauts. Cette approche est utilisée dans la reconnaissance par réseaux de neurones.

5.7.2.3 Descriptions topologiques en segments ou embranchements

Une méthode complètement structurelle consiste à décrire le caractère en primitives basées sur les segments, ou sur les terminaisons et embranchements. Ces primitives sont obtenues à partir du squelette, ou directement à partir de l'image du caractère.

Segments

Un caractère est décrit comme une arborescence de segments orientés. À chaque segment est attribué un code indiquant sa direction et son sens. Pour cela, on utilise généralement les codes de Freeman : huit directions orientées, numérotées de 0 à 7 dans le sens trigonométrique, correspondant aux points cardinaux (E, NE, N...), en partant de 0 = Est. On commence la description à une extrémité du caractère, et on la poursuit jusqu'à ce qu'il soit entièrement parcouru. Les bifurcations donnent lieu à la création de nouvelles branches dans l'arbre. La comparaison des caractères se ramène ensuite à une comparaison d'arbres.

Terminaisons et embranchements

Dans ce cas, on s'intéresse plutôt aux terminaisons, bifurcations et embranchements des segments. Sur la base des huit sens définis par les codes de Freeman, on définit huit types d'extrémités, $8 \times 7/2 = 28$ types de bifurcations, $28 \times 6/3 = 56$ types d'embranchements à trois branches, et $56 \times 5/4 = 70$ types d'embranchements à quatre branches. L'ensemble de ces primitives est décrit par un alphabet de $8 + 28 + 56 + 70 = 162$ symboles (en fait moins, car les embranchements à quatre branches sont rares). Chaque caractère, balayé horizontalement de gauche à droite, est décrit par la suite des symboles correspondant aux primitives trouvées lors du balayage. Plusieurs primitives détectées simultanément sont placées entre parenthèses. La comparaison des caractères se ramène dans ce cas à la comparaison de chaînes de symboles.

5.7.2.4 Projections partielles sur des axes orthogonaux

Il est également possible de décomposer un caractère en éléments obtenus par des projections partielles horizontales et verticales (figure **16**).

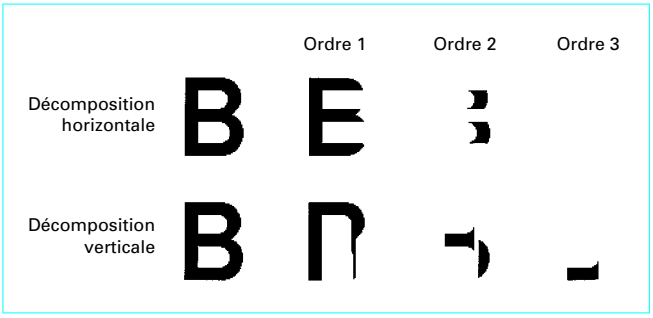

Figure 16 – Décomposition d'un B en éléments projectifs d'ordre *k*

On projette à chaque fois jusqu'à la rencontre d'un point blanc. Il y a au plus quatre projections partielles, horizontales et verticales, ce qui limite l'importance de la description. Ces projections partielles se réduisent elles-mêmes à des formes plus simples, facilement analysables, et réductibles à un nombre restreint de primitives.

5.7.2.5 Analyse des zones blanches, concavités et trous

Les primitives précédentes étaient basées sur la description des surfaces et segments noirs qui constituent les caractères. On peut au contraire fonder l'analyse sur les plages blanches qui les entourent ou qui sont contenues à l'intérieur.

Description des concavités et des trous

Les concavités et les trous sont des éléments fondamentaux de description de la topologie des caractères. Le fait que des formes présentent une concavité ou une ouverture vers le haut, le bas, la gauche ou la droite, et contiennent un ou plusieurs trous différemment situés, permet de les distinguer. Ces éléments sont codables au moyen d'un alphabet.

Analyse quantitative des zones blanches

Une manière d'affiner la description précédente consiste à préciser la position des plages blanches, ainsi que leur importance en surface. En position, on choisira les huit directions orientées des codes de Freeman. La surface de chaque plage blanche sera rapportée à la surface du rectangle englobant, ou à la surface totale de l'ensemble des plages blanches du caractère. Cela permet d'obtenir des valeurs numériques indépendantes de la taille du caractère, et dans certains cas, de la fonte.

5.7.3 Caractéristiques définies automatiquement

Les expériences multiples de reconnaissance de caractères ont permis de constater que les phénomènes de dégradation des images décrits au paragraphe 5.2 sont prépondérants, et rendent caduques les modélisations humaines. Les ensembles de primitives définis par l'intelligence humaine fonctionnent tous parfaitement pour discriminer et identifier des caractères bien formés et imprimés. Par contre, dès qu'on a affaire à des documents réels et aux caractères coupés, collés, bruités... qu'ils contiennent, aucune description formelle ne résiste. En analysant certaines images réelles de caractères, on s'aperçoit que ce qui distingue un caractère d'un autre est souvent plus une subtile accumulation de différences réparties dans toute l'image, que certains traits distinctifs précis.

Les logiciels de reconnaissance par réseaux de neurones permettent de s'affranchir de cette limitation : après normalisation des images de caractères, celles-ci sont injectées directement sur les entrées du réseau, qui utilise ses premières couches cachées (§ 5.8.6) pour extraire automatiquement, après apprentissage (§ 5.9), des caractéristiques qui ne pourraient correspondre à aucune modélisation humaine. On a donc en quelque sorte une définition et une extraction automatiques des caractéristiques.

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

5.8 Décision et classement

C'est la décision qui réalise véritablement la reconnaissance. On a vu qu'elle consiste en un classement : choix de la classe dont la représentation ou le modèle est le plus proche. Elle conduit au rejet si aucun modèle ne correspond, à la reconnaissance en cas de correspondance avec un modèle ou une classe unique, enfin à la confusion qui risque d'entraîner une substitution si plusieurs modèles conviennent. Dans ces deux derniers cas, il est possible de quantifier la décision par une mesure de vraisemblance, aussi appelée score. Les principales approches, passées en revue ci-après, sont liées au modèle de description choisi.

5.8.1 Arbres de décision

La méthode la plus naturelle pour faire un choix est celle de l'arbre de décision. Supposons qu'on soit capable de réaliser des partitions de plus en plus fines de l'ensemble des modèles, jusqu'à aboutir à celle qui comporte un seul modèle par classe, et d'autre part d'associer un test, numérique ou symbolique, à chaque niveau de partitionnement. On construit ainsi un arbre de décision, dans lequel un test précis est associé à chaque nœud, et un résultat de test à chaque branche. On effectuera par exemple les deux premiers tests sur la présence de hampe et de jambage, pour partitionner l'ensemble des caractères en quatre classes. Suivra un test sur l'élongation permettant de distinguer trois sous-classes, etc.

L'approche décrite ci-dessus consiste en une analyse intellectuelle du problème, dont nous avons souligné les limites au paragraphe (§ 5.7.3). Une méthode plus élaborée consiste, après avoir défini un jeu de caractéristiques qui permet une séparation correcte de toutes les classes de caractères, à utiliser des algorithmes de classification automatique, pour décider du regroupement hiérarchique de ces classes en partitions et sous-partitions. Cela nécessite de disposer d'une distance entre classes liée aux caractéristiques choisies. Pour réaliser une classification hiérarchique, deux types de méthodes existent :

— **ascendantes** : sur l'ensemble des N classes, on agrège les deux classes les plus proches. On obtient ainsi une partition à $N-1$ éléments, puis on réitère l'opération jusqu'à obtenir une partition qui contient toutes les classes ;

— **descendantes** : on réalise une dichotomie de l'ensemble des classes, en fonction d'un critère qui maximalise la distance globale des deux sous-partitions. On réitère ensuite l'opération sur chacune d'elles.

Ce type de classification hiérarchique permet ainsi de construire automatiquement un arbre de décision, utilisable ensuite pour le classement des objets à reconnaître.

L'avantage des arbres de décision tient à leur rapidité, car ils sont parcourus en effectuant un nombre réduit de tests : un arbre binaire permet un choix parmi N classes avec $\log_2(N)$ tests. Leur inconvénient est qu'en cas d'erreur sur un test en début d'arbre, le classement est faux et très éloigné du résultat correct.

5.8.2 Discrimination fonctionnelle linéaire

Cette approche, ancienne, est juste citée pour mémoire. Lorsque les caractéristiques extraites sont des mesures, le caractère est transformé en un ensemble de valeurs numériques. Le résultat peut être conçu comme un point dans l'espace R^m, m étant le nombre de mesures. Une classe, ensemble des caractères d'un même type, est représentée par un nuage de points inclus dans une zone délimitée. Les caractéristiques choisies sont telles que :

— les zones représentant des classes distinctes soient disjointes : si on a n classes, les vecteurs de mesure $X = (x_1, x_2 \ldots x_m)$ auront une dimension m suffisante pour réaliser cette condition ;

— les surfaces de séparation, ou fonctions de discrimination entre les classes soient simples. On fait généralement en sorte que ces surfaces soient des hyperplans : les fonctions de discrimination $d_i(X)$ sont alors linéaires, de la forme :

$$d_i(X) = w_{i\,1}x_1 + w_{i\,2}x_2 + \ldots + w_{i\,m}x_m + w_{i\,m+1}$$

Dans le cas le plus défavorable, les classes sont simplement séparables deux à deux ; il faut alors définir $k = n(n-1)/2$ fonctions linéaires de discrimination. L'application d'une telle fonction à un vecteur de mesure X, pour séparer deux classes, fournit un résultat positif en cas d'appartenance à une classe, négatif en cas d'appartenance à l'autre (représentable par un bit mis à 0 ou 1). En appliquant successivement les k fonctions de discrimination au vecteur de mesure, on obtient un mot de k bits. Une table de correspondance permet d'en déduire immédiatement la classe.

Ce procédé était utilisé dans les lecteurs optiques par masques partiels, pour la reconnaissance monofonte, en particulier celle des fontes spéciales OCR-A et OCR-B, dont la forme avait été étudiée pour permettre une séparation linéaire facile.

5.8.3 Méthodes statistiques bayesiennes

Cet ensemble de méthodes est fondé sur un modèle probabiliste. À partir des caractéristiques mesurées, les probabilités d'appartenance des formes-échantillons aux classes prédéfinies sont déduites, ce qui permet de faire le meilleur choix, et de disposer de choix alternatifs. On définit par C l'ensemble des modèles de caractères (il peut y avoir plusieurs modèles pour chaque lettre de l'alphabet), et $\Omega = \{\omega_k\}$ une partition réalisée sur C : chaque classe ω_k représente un seul caractère. À chacune, on peut associer une probabilité d'apparition de ses éléments : $p(\omega_k)$, supposée connue *a priori*. Il s'agit de la probabilité de présence de chaque caractère dans les textes, qui est connue pour chaque langue, et mesurable sur un *corpus* donné.

$P(\mathrm{O})$ désignera la probabilité de chaque observation ou mesure ; celles-ci sont considérées comme équiprobables.

On définit enfin les probabilités conditionnelles : $p(\mathrm{O}/\omega_k)$: probabilité d'une mesure O sachant qu'on a un caractère ω_k, et $p(\omega_k/\mathrm{O})$: probabilité d'avoir le caractère ω_k, après une mesure O.

Étant donné une observation O, la décision sera prise selon le critère suivant : on choisira le caractère ω_k qui maximise la probabilité $p(\omega_k/\mathrm{O})$. Le théorème de Bayes nous indique que :

$$p(\omega_k/\mathrm{O}) = p(\mathrm{O}/\omega_k) \times p(\omega_k)/p(\mathrm{O})$$

Comme les $p(\mathrm{O})$ sont équiprobables, trouver le maximum de $p(\omega_k/\mathrm{O})$ revient à trouver celui du produit $p(\mathrm{O}/\omega_k) \times p(\omega_k)$. Or ces quantités peuvent être déterminées par apprentissage sur un *corpus*, alors que $p(\omega_k/\mathrm{O})$ est inaccessible directement.

Au contraire de l'arbre de décision, la mesure est ici mise en relation avec l'ensemble des classes possibles, et le maximum de probabilité est recherché parmi les valeurs trouvées. La méthode est plus lente, mais elle a l'avantage de donner les solutions alternatives les plus probables : en cas d'erreur détectée, le meilleur choix suivant peut être proposé.

Exemple : cette approche a été utilisée pour un lecteur postal, dans lequel l'image de chaque caractère à reconnaître était envoyée en parallèle à N processus, chacun chargé de reconnaître un seul type de caractère. À l'issue de son analyse, chaque processus renvoyait à un module de décision la probabilité d'appartenance du caractère-échantillon à la classe dont il était chargé. La décision se réduisait à une recherche de maximum.

5.8.4 Programmation dynamique et parcours d'automates

La programmation dynamique est une méthode de recherche d'un chemin optimal dans un graphe simple ou pondéré (à chaque arc est associé un poids ou un coût). Elle consiste à parcourir

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

tout ou partie du graphe, et à construire un tableau dans lequel le poids total de chaque chemin est noté au fur et à mesure du parcours, somme des poids de chaque branche parcourue. En fin de parcours, le poids minimum et le chemin optimal sont obtenus. Cette méthode permet de calculer la distance entre deux chaînes de caractères ou de symboles, et le recalage entre ces chaînes. Elle permet aussi la décision dans un automate à états finis.

5.8.4.1 Distance entre chaînes

La distance d'édition entre chaînes de caractères a été définie par Levenshtein. Elle est fondée sur **trois transformations élémentaires** qui portent sur les lettres de l'alphabet, et dont la combinaison permet de transformer une chaîne en une autre. À chacune, est associé un coût individuel Γ :

— la **substitution** a $\rightarrow$ b, de coût Γ(a, b) ;

— l'**insertion** $\varnothing \rightarrow$ a, de coût $\Gamma(\varnothing, a)$ ($\varnothing$ représente l'ensemble vide) ;

— la **destruction** a $\rightarrow \varnothing$, coût $\Gamma(a, \varnothing)$.

Par exemple, la chaîne « abcd » peut être transformée en « acbe » par la suite d'opérations élémentaires : (b $\rightarrow \varnothing$), (d $\rightarrow$ b), ($\varnothing \rightarrow$ e), avec un coût total de $\Gamma(b, \varnothing) + \Gamma(d, b) + \Gamma(\varnothing, e)$.

La distance d'édition $D(x, y)$ entre deux chaînes x et y est définie comme le coût de la suite de transformations élémentaires la moins coûteuse pour transformer x en y. On prend généralement une même valeur, égale à 1, pour les coûts élémentaires.

Un algorithme de calcul de cette distance par programmation dynamique est donné par Wagner et Fisher dans [31] : après construction d'un tableau basé sur la chaîne de départ x et la chaîne d'arrivée y, la relation de récurrence suivante est appliquée :

$$D(i, j) = \min \begin{cases} D(i-1, j) + \Gamma(x_i, \varnothing), & \text{effacement} \\ D(i, j-1) + \Gamma(\varnothing, y_j), & \text{insertion} \\ D(i-1, j-1) + \Gamma(x_i, y_j) & \text{substitution} \end{cases}$$

On note dans un deuxième tableau le choix réalisé pour obtenir le coût stocké dans chaque case du premier tableau. La distance entre les deux chaînes est obtenue en fin de parcours. Le parcours inverse, aussi dit *backtrack*, décrit la suite des opérations appliquées et permet la mise en correspondance des deux chaînes, en donnant les positions des insertions, destructions et substitutions.

5.8.4.2 Parcours d'automates

Dans les cas de reconnaissance structurelle, le caractère à reconnaître est décrit par une succession de primitives appartenant à un ensemble fini, assimilable à un alphabet constitué de symboles. L'ensemble des modèles possibles est une liste de mots, plus précisément d'expressions régulières, construits avec cet alphabet. Cette liste peut être mise sous la forme d'un automate à états finis.

Exemple : la suite des neuf mots écrits avec l'alphabet {a, b, c, d, e, f} : ac - aceabf - acec - aceccef - acecdbd - acecdc - acecdceb - acecf - ad peut être représentée par l'automate de la figure **17**.

Les états cerclés plus épais sont les états finaux. Chacun d'eux représente une classe à reconnaître. Il existe des algorithmes permettant de minimiser les automates à états finis.

La vérification de l'appartenance d'une forme-échantillon à une classe peut se faire très rapidement par le parcours de l'automate : si la suite des symboles qui décrivent cette forme aboutit à un état final, cette forme appartient à la classe correspondante.

5.8.4.3 Programmation dynamique appliquée au parcours d'automate

Le parcours précédent nécessite une correspondance parfaite entre la forme à reconnaître et un des modèles décrits par l'automate. Si cette correspondance n'existe pas, un algorithme de type programmation dynamique, pour trouver le chemin le plus proche aboutissant à un état final, peut être utilisé. En attribuant des poids aux substitutions entre symboles, plusieurs parcours de l'automate avec le mot à reconnaître sont essayés, donnant lieu au calcul du poids total sur chaque chemin. La fixation d'un seuil à ne pas dépasser évite de parcourir la totalité de l'automate. L'état terminal obtenu avec le poids le plus faible donne la classe la plus proche.

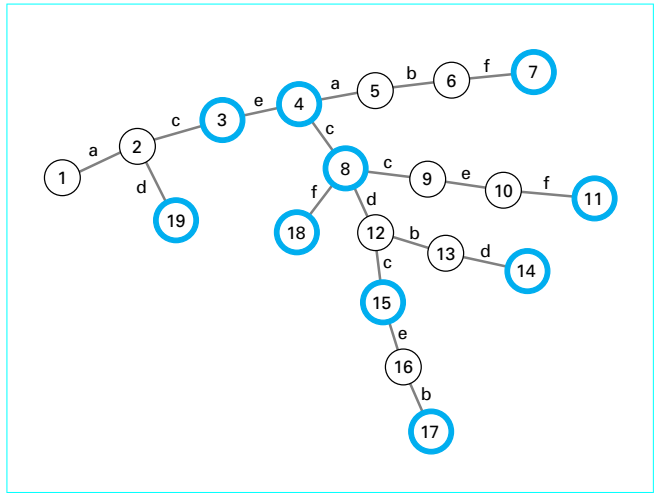

Figure 17 – Automate d'états finis représentant une liste de mots

5.8.5 Méthodes stochastiques à base de modèles de Markov

Les modèles stochastiques ont été mis au point pour décrire des processus qui évoluent au cours du temps. Ils peuvent servir également à modéliser des successions de mesures obtenues en progressant le long d'un axe, et sont donc aussi utilisés pour la reconnaissance de l'écriture, manuscrite comme imprimée [8] [40]. Une observation $O(t)$ évolue selon une succession de valeurs mesurées : O_1, $O_2 \ldots O_k$. Les successions d'états obéissent aux lois de probabilités suivantes (probabilités conditionnelles) :

$$P(O_1, O_2, \ldots O_k) = P(O_1, O_2 \ldots O_{k-1}) \times P(O_k/O_1, O_2 \ldots O_{k-1})$$

$$= P(O_1) \times P(O_2/O_1) \times P(O_3/O_1, O_2) \times \ldots$$
$$P(O_k/O_1, O_2 \ldots O_{k-1})$$

Un processus est dit de Markov et du premier ordre si un état ne dépend que de son prédécesseur immédiat ; ainsi :

$$P(O_1, O_2, \ldots O_k) = P(O_1) \times P(O_2/O_1) \times P(O_3/O_2) \times \ldots \times P(O_k/O_{k-1})$$

Il est dit stationnaire si ces lois de probabilité sont invariantes au cours du temps. Si N états observables sont possibles, une matrice $[P]$ de probabilités de transition de dimension $N \times N$ est définie, telle que $P_{ij} = P(O_i/O_j)$.

Les modèles de Markov cachés (en anglais : *hidden Markov models* - HMM) prennent en compte deux suites de variables aléatoires.

Le processus de base peut se trouver dans un ensemble fini d'états, notés $\{q_i\}$, non directement observables, d'où leur nom de *cachés*. Une première matrice [Q] donne les probabilités de transition entre les états cachés : $Q_{ij} = P(q_i/q_j)$.

Les états ne sont eux-mêmes connus qu'à partir de mesures $\{O_k\}$ qui peuvent prendre M valeurs. Une deuxième matrice $[O]$ définit

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

des probabilités d'observations : $O_{ki} = P(O_k/q_i)$. Elle représente la probabilité d'obtenir la mesure $\{O_k\}$ alors que le modèle est dans l'état $\{q_i\}$.

Le modèle de Markov caché est noté λ. Il est défini par l'ensemble $\lambda = \{Q, O, \Pi\}$, Π représentant les probabilités initiales des états cachés.

Il existe deux types principaux de modèles de Markov cachés : les modèles ergodiques, et les modèles gauche-droite. Dans les **modèles ergodiques**, toutes les transitions d'un état vers un autre sont possibles, alors que les **modèles gauche-droite** interdisent les transitions d'un état vers un autre état antérieur. Les modèles gauche-droite séquentiels sont adaptés à la description de phénomènes se déroulant dans le temps, comme la parole, ou linéaires dans l'espace, comme le texte.

Dans le cas de la reconnaissance de caractères, on associe à chaque classe de caractères un modèle de Markov caché qui génère, à travers un canal de communication complexe (impression, dégradation, échantillonnage, binarisation...), une image observable de ce caractère. L'ensemble des caractéristiques extraites de cette image donne l'observation $\{O_k\}$.

L'utilisation des HMM suppose la résolution de plusieurs problèmes :

— l'évaluation des probabilités d'observation $p(O/\lambda)$ pour chaque modèle, qui est réalisée par un algorithme nommé *forward-backward* ;

— la décision qui consiste à trouver un chemin maximisant la probabilité des états successifs, étant donné une suite d'observations ; elle est réalisée par l'algorithme de Viterbi, une variante de la programmation dynamique décrite au paragraphe 5.8.4 ;

— l'apprentissage (voir § 5.9).

Pour plus de détails, on consultera la référence [8].

5.8.6 Réseaux de neurones

Les réseaux de neurones, imaginés par analogie avec le fonctionnement des systèmes nerveux, sont des assemblages de processus élémentaires baptisés *neurones formels*, à forte connectivité, et généralement organisés en plusieurs couches [41]. Un neurone formel est un opérateur à N entrées et une sortie (figure **18**), de la forme :

$$S = f_T\left(\sum_i w_i x_i\right)$$

Les w_i sont appelés coefficients de pondération ou coefficients synaptiques. La fonction de transfert f_T est de type seuillage : fonction de Heavyside, sigmoïde... Dans le cas le plus simple (fonction de Heavyside), la sortie ne peut prendre que deux valeurs.

On montre qu'un réseau organisé en une seule couche est un classifieur linéaire, du type décrit au paragraphe 5.8.2. L'organisation la plus classique est dite *en couches* et consiste à relier les sorties d'une couche de neurones aux entrées de la couche suivante. Dans ce cas, on a une couche d'entrée, une couche de sortie, et des couches intermédiaires appelées couches cachées. Il existe aussi des réseaux dynamiques du type de Hopfield qui n'ont pas la même architecture.

Les réseaux organisés en plusieurs couches permettent de discriminer des classes d'objets qui ne sont pas linéairement séparables [41]. C'est leur premier intérêt, le second étant leur capacité d'apprentissage : les coefficients synaptiques w_i peuvent en effet être calculés à partir d'une base importante d'exemples, dans laquelle on dispose à la fois des données en entrée et du résultat de la reconnaissance en sortie.

Par rapport aux concepts définis au paragraphe 5.5, nous dirons que le réseau réalise un travail de classification pendant la phase d'apprentissage, et de classement lors de la reconnaissance. Un

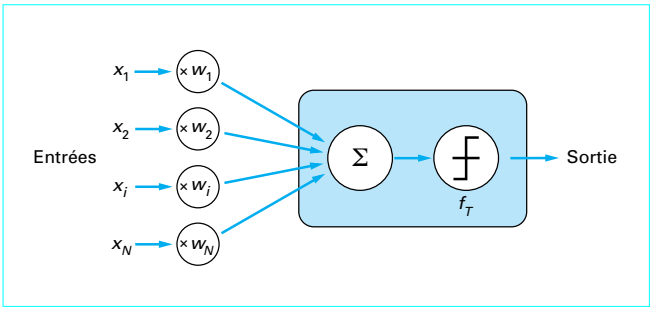

Figure 18 – Neurone formel et fonction de transfert

réseau de neurones est donc fondamentalement un classifieur. En ce qui concerne l'extraction des caractéristiques, deux options sont possibles :

— fournir l'image brute des caractères, après leur normalisation, aux entrées du réseau : dans ce cas, les premières couches cachées du réseau se chargent de l'extraction des caractéristiques ;

— extraire au préalable les caractéristiques jugées pertinentes, et utiliser le réseau uniquement comme classifieur à partir de ces éléments.

Le grand avantage des réseaux de neurones est leur capacité à prendre une décision à partir de critères non formalisables explicitement. Leur difficulté de mise en œuvre réside dans l'apprentissage, et les temps de traitement, car les applications opérationnelles nécessitent des réseaux à plusieurs milliers de neurones. Le logiciel *EasyReader* de Mimétics réalise l'OCR avec cette technique.

5.9 Classification et apprentissage

L'apprentissage est réalisé à partir d'une base de données d'images de caractères servant de référence. La mise au point de logiciels omnifontes, robustes par rapport aux variations observées dans les documents, nécessite la constitution de bases de très grande taille, qui contiennent plusieurs millions de caractères labellés. La validation nécessaire après l'apprentissage amène généralement à séparer la base en deux : une sous-base d'apprentissage et une sous-base de test. L'apprentissage consiste en deux grandes étapes :

— la **classification** manuelle ou assistée des images des caractères et des modèles qui leur sont associés, et l'**étiquetage** ou identification des classes mises en évidence ;

— le **calcul des paramètres** numériques internes aux modèles, correspondant aux différentes classes, par des méthodes algorithmiques entièrement automatiques.

Il nécessite la définition de fonctions de coût, ou de mesure d'erreur, qui peuvent être booléennes (bon, mauvais) ou numériques, pour pouvoir ajuster les paramètres internes en fonction des erreurs observées.

Il sera dit supervisé en cas d'interaction avec un « professeur » qui prend des décisions, ajuste une architecture ou un modèle en cours d'apprentissage... Il sera au contraire non supervisé s'il est entièrement automatique et que seule la base de référence est fournie (les images des caractères étant non identifiées), à un système comportant une architecture figée.

Dans le cas des modèles de Markov cachés, la deuxième étape consiste à calculer, pour des modèles déterminés à l'avance, les matrices de probabilité des états initiaux, des transitions entre états, et de distribution des observations. On utilise un algorithme dit de Baum-Welch [8].

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

Pour les réseaux de neurones, l'architecture (nombre de neurones, de couches, topologie des connexions...) est fixée, et l'apprentissage consiste à calculer les coefficients synaptiques. De nombreuses méthodes existent, la plus connue étant celle dite de rétropropagation du gradient [41].

5.10 Combinaison de plusieurs méthodes de reconnaissance

5.10.1 Utilisation de plusieurs techniques dans un même système

La diversité des documents rencontrés est telle qu'aucune méthode d'OCR ne donne de bons résultats dans tous les cas. On conçoit bien que des caractères bien formés et segmentés, ou des caractères cassés en plusieurs morceaux, ou au contraire des caractères très gras et quasiment tous collés, nécessiteront des méthodes de reconnaissance différentes. C'est ainsi que le logiciel le plus répandu sur le marché — *Omnipage* de Caere — combine en interne plusieurs moteurs de reconnaissance, basés sur des principes différents, pour pouvoir traiter la plus grande diversité de documents possible [9].

Par exemple, trois méthodes sont associées :

— la première, classique, est conforme au modèle décrit précédemment : segmentation des mots en caractères, reconnaissance des caractères isolés, post-traitements contextuels ; elle est adaptée aux caractères bien formés et facilement segmentables ;

— la deuxième vise la reconnaissance des caractères bruités ou cassés mais segmentables : elle consiste à accumuler les caractéristiques extraites sur les caractères successifs de l'ensemble du mot et à mettre le résultat en correspondance avec un ensemble de modèles qui décrivent globalement tous les mots du dictionnaire par cette méthode ;

— la troisième, adaptée aux caractères collés, consiste à extraire des caractéristiques globales à partir des mots, et à les comparer à des modèles de mots constitués sur les mêmes bases.

Certains systèmes combinent des méthodes fondées sur une modélisation structurelle préalable, avec une reconnaissance par réseaux de neurones. Il est acquis que la reprise de textes de mauvaise qualité passe par la mise en œuvre de méthodes globales de reconnaissance des mots. Celles-ci se heurtent néanmoins au problème cité au paragraphe 5.11.2.2 : il est impossible de disposer de dictionnaires complets comportant la totalité du vocabulaire rencontré dans les textes.

5.10.2 Couplage de plusieurs OCR du commerce

Une autre méthode consiste à corréler les résultats de plusieurs logiciels d'OCR du commerce, appliqués successivement et indépendamment à un même bloc de texte [32], ce qui suppose une segmentation préalable du texte indépendante de ces logiciels. Le logiciel *PRASAD* met en œuvre cette technique, basée sur un algorithme de Handley et Hickey [33], généralisation de l'algorithme de Wagner et Fischer [31]. Pour pouvoir réaliser un vote majoritaire, il utilise trois logiciels du marché : ceux des sociétés Caere, Xerox Imaging Systems et Mimetics.

La première étape consiste à normaliser les textes en éliminant les lignes vides et en réduisant les séparateurs de mots à un seul espace. La deuxième étape consiste à synchroniser les contenus des chaînes de caractères correspondantes de chacun des trois blocs. Chaque chaîne est obtenue par application d'un OCR différent sur une même ligne de texte. Les caractères identiques sont alignés à l'aide d'un algorithme de type programmation dynamique (§ 5.8.4) généralisé en trois dimensions. Les manques dans une chaîne par rapport aux deux autres sont comblés en introduisant des caractères « # » de remplissage.

La troisième étape est le vote majoritaire appliqué à chaque triplet de caractères. Chaque caractère d'un triplet appartient à une des chaînes obtenues à partir d'une même ligne de texte. Ce vote applique les règles suivantes :

— si deux au moins des trois caractères sont identiques, le caractère majoritaire est retenu ;

— si les trois caractères sont différents, celui donné par le logiciel d'OCR qui donne globalement les meilleurs résultats (Caere) est retenu. Cette technique permet de diviser le taux d'erreur par 2 ou 3, et d'atteindre des taux moyens de 0,1 % sur des originaux de bonne qualité.

5.11 Posttraitements : levées d'ambiguïtés et utilisation de lexiques

5.11.1 Levées d'ambiguïtés

Quelle que soit la sophistication des méthodes d'extraction de primitives et de décision, des confusions subsistent souvent à l'issue de ces étapes. Des agents spécialisés doivent alors intervenir pour lever certaines ambiguïtés en fonction du contexte [9] :

— la distinction entre certaines **majuscules et minuscules** ayant la même forme (comme « c » et « C ») pourra être faite en prenant en compte l'emplacement de la ligne médiane, la position du caractère en début ou milieu de mot, en début de ligne, ou derrière un point, et la présence d'autres majuscules dans le même mot ;

— la confusion entre **chiffres et lettres** (« 1 » et « l », ou « 8 » et « S » dont les boucles sont fermées) pourra être levée grâce au contexte, tout numérique, ou tout alphabétique ;

— la confusion d'un « m » cassé reconnu comme « rn » dans un mot qui comporte deux « m » (par exemple, « comment ») pourra être levée par une analyse des trigrammes de la langue. Celle-ci permettra de constater que le trigramme « rnm » est très improbable, et donc de rétablir le « m ».

Cet exemple montre l'intérêt de faire intervenir des connaissances sur la langue, d'où les approches lexicales décrites ci-après.

5.11.2 Utilisation de lexiques

Il est possible de faire intervenir un lexique à deux niveaux : en cours de reconnaissance, ou après la reconnaissance.

5.11.2.1 Validation des hypothèses sur les caractères

Tous les logiciels d'OCR du marché incorporent des dictionnaires de langue (douze langues européennes disponibles), pour faciliter la reconnaissance. Ces dictionnaires ne sont pas exhaustifs, ils comportent quelques dizaines de milliers de mots, les plus fréquents dans la langue. Ils sont utilisés dans une optique d'aide à la reconnaissance (symbolique) des caractères. On a vu que les images des caractères d'une page peuvent être préalablement regroupées en un ensemble de modèles, suite à une reconnaissance de bas niveau, avant d'être soumises à la reconnaissance symbolique. Celle-ci leur attribue des codes qui identifient les caractères, avec un certain score pour chacun. Avec ces hypothèses sur les caractères, les mots du texte sont reconstruits. Si parmi ceux-ci, un nombre suffisant appartient au dictionnaire, cela permet alors de valider les hypothèses sur les caractères qu'ils contiennent. Ainsi sont mises en place des heuristiques fondées sur une interaction entre la reconnaissance et les accès au dictionnaire, pour lever certaines ambiguïtés.

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

5.11.2.2 Contrôle *a posteriori*

Une autre approche est possible : l'utilisation d'un dictionnaire beaucoup plus complet, pour une vérification systématique du texte après la reconnaissance, sans interaction avec celle-ci. Le dictionnaire électronique *DELAF* fourni par le LADL (Laboratoire de Linguistique) comporte plus de 600 000 formes fléchies du français courant : ensemble des mots au singulier et au pluriel, toutes les formes conjuguées des verbes, etc. L'utilisation de ce genre de dictionnaire très complet permet de baliser tous les mots du texte qui n'appartiennent pas au français courant, dans une optique de détection des erreurs résiduelles et de correction par un opérateur. Cette approche est limitée par le fait que dans les textes, un tiers du vocabulaire environ n'appartient pas à la langue courante : noms propres (personnages, lieux, pays, fleuves...), abréviations, sigles et noms de sociétés, termes techniques... Cette proportion augmente encore dans les textes techniques. On peut certes envisager la constitution de dictionnaires encore plus complets, dépassant le million de mots, ou l'adjonction de dictionnaires spécialisés, qui permettraient de détecter avec plus de fiabilité les erreurs potentielles après l'OCR. Néanmoins, l'exhaustivité restera toujours inaccessible.

5.12 Reconnaissance des fontes

La reconnaissance de la fonte (type, corps, graisse, italique, stature...) est une étape supplémentaire dans la reconnaissance des caractères. Elle peut être effectuée avant ou après l'OCR, et présente des intérêts multiples :

— améliorer les performances de la reconnaissance, si la fonte peut être identifiée avant l'OCR : la reconnaissance monofonte, plus simple, donne de meilleurs taux de succès que la reconnaissance omnifonte ;

— reconstituer une image fidèle du document d'origine, pour le restituer par exemple dans un format pdf codé ;

— servir de base à la reconnaissance de la structure physique, et ultérieurement, de la structure logique.

Les logiciels d'OCR les plus courants (*TextBridge* de Xerox Imaging Systems...) sont capables d'identifier quelques caractéristiques comme la graisse, l'italique, le soulignement, mais pas de reconnaître la fonte stricto sensu. Seul *Fine Reader* possède cette capacité, et restitue du texte d'apparence et de mise en page conforme à l'original, au format rtf (BitSoft).

5.12.1 Avant OCR

Un premier groupe de méthodes [34] consiste à extraire des caractéristiques globales à partir du texte, après classification préalable des caractères en fonction de leurs dimensions : rapport entre la hauteur totale et celle de la bande centrale du texte, mesures des hauteurs et largeurs moyennes par type de classe, de l'espacement moyen des caractères, de la largeur et épaisseur moyennes des plages noires, de gradients, etc. Ces mesures (moyennes et écarts-types) sont ensuite comparées à celles d'une base de référence contenant les données de fontes connues, dans plusieurs corps. Une décision bayesienne permet d'en déduire le type de fonte, le corps, la graisse, avec des taux de succès supérieurs à 95 %.

Un deuxième groupe de méthodes se base sur la détection de certains mots courts très fréquents et caractéristiques de chaque langue (en français, « le », « la », « et », en anglais, « a », « the »...) à partir de critères à la fois morphologiques (longueur) et statistiques (fréquence d'apparition) [35] [36]. Des masques sont constitués à partir des mots courts du texte (comme décrit au paragraphe 5.6.2). Un *pattern matching* avec une base contenant des images de référence de ces mots dans différentes fontes et les corps les plus courants, permet de faire un premier tri à partir d'un seul mot, puis d'affiner avec tous les modèles de mots courts reconnus. L'ensemble qui donne le meilleur score correspond à la fonte et au corps majoritaires dans le texte. Cette méthode est moins fine que la précédente, car elle ne permet pas de déterminer la fonte à partir d'une seule ligne de texte.

5.12.2 Après OCR

Après l'OCR, une donnée supplémentaire est disponible : l'identification des caractères. Les deux types de méthodes précédents sont applicables, à ceci près que les mesures ou les comparaisons de formes peuvent être faites sur certains caractères choisis pour leurs particularités. L'inconvénient tient au fait que pour obtenir des valeurs utilisables, il faut faire des mesures sur un nombre de caractères significatif, donc répartis dans tout le texte, et que cela interdit encore la possibilité de déterminer la fonte sur une petite zone de texte.

6. Reconnaissance des zones non textuelles

Les paragraphes 4.5, 4.6, 4.7 et 4.8 traitent de la segmentation du texte et des zones non textuelles. En général seule la reconnaissance du texte offre un intérêt. Les autres types d'informations sont simplement ignorés, ou stockés sous forme image. Cependant il est parfois nécessaire de pousser les traitements plus loin, en particulier pour les tableaux, et de réaliser une véritable reconnaissance sur les zones non textuelles.

6.1 Tableaux

La reconnaissance du contenu des tableaux, après leur détection et l'analyse de leur structuration physique (§ 4.6), consiste à appliquer l'OCR sur le contenu de chacune de leurs cellules. Un problème subsiste cependant : la structure logique d'un tableau ne correspond pas toujours à sa présentation physique. Quelquefois, les textes de chacune des cellules alignées horizontalement doivent être simplement mis en correspondance. D'autres fois, il faut faire un découpage plus fin, et réaliser une correspondance ligne à ligne, ou sous-bloc à sous-bloc, pour respecter la structuration réelle de l'information. Cela a pour conséquence qu'une reconnaissance entièrement automatique des tableaux n'est pas envisageable actuellement. L'interaction avec un opérateur, vérifiant la structure trouvée, ou appliquant des modèles logiques adaptés aux documents, est encore nécessaire pour obtenir une reconnaissance correcte dans tous les cas.

Un autre problème concerne la représentation de l'information structurée : on trouve dans l'initiative CALS (*continuous acquisition and life-cycle support*, initialisé par le *Department of Defense* américain) des DTD (*document type definition*) SGML de tableaux qui permettent de représenter toutes les structures de tableaux possibles, mais aucune norme reconnue n'existe.

Notons deux progrès récents pour la reconnaissance des tableaux :

— le logiciel *PRASAD* les localise et les stocke sous forme d'images ;

— le logiciel *Fine Reader* les reconnaît et permet de les récupérer sous *Word*.

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

6.2 Formules mathématiques

La reconnaissance des formules mathématiques présente plusieurs difficultés. D'abord, elles comportent potentiellement un grand nombre de symboles : alphabet latin, alphabet grec, chiffres, opérateurs arithmétiques, algébriques, booléens..., signes diacritiques, sommations, intégrales, barres de fraction, quantificateurs... De plus, ces symboles peuvent avoir des tailles très différentes au sein d'une même formule en fonction de leur contenu ou des informations suivantes. Souvent leur structure spatiale est bidimensionnelle, contrairement au texte qui est linéaire, et il existe une infinité de combinaisons des variables et des opérateurs dans les deux dimensions.

Aucun logiciel commercial ne permet la reconnaissance des formules. Les prototypes [37] [38] [39] mettaient généralement en œuvre les étapes suivantes :

— reconnaissance des symboles (opérateurs et variables) par leur seule forme, avant toute analyse de structure ;

— association à ces symboles de leurs coordonnées pour les positionner dans le plan ;

— repérage des opérateurs (égalité, opérateurs arithmétiques, barres de fraction, sommations, intégrales...) ;

— reconnaissance des portions de formule, assistée par une grammaire associée à chaque opérateur, et prenant en compte la position des variables dans le plan par rapport à ces opérateurs ;

— reconstitution et codage de la formule globale.

En ce qui concerne le codage des formules reconnues, plusieurs solutions se présentent :

— utiliser le format *TeX* ou *LaTeX* ;

— utiliser une DTD CALS pour les formules mathématiques ;

— utiliser le langage MathML (*mathematical markup language*) en cours de définition, à partir de XML.

La première solution est la plus rapidement utilisable, la dernière est probablement celle qui s'imposera à long terme.

6.3 Graphiques et schémas

La reconnaissance des graphiques et des schémas consiste globalement à vectoriser les segments de droite qu'ils comportent, et à reconnaître (symboliquement) et coder les formes primitives qu'ils contiennent : symboles de portes logiques pour des schémas électroniques, vannes pour des schémas mécaniques, caractères alphabétiques et numériques, etc. En fait on rencontre autant de traitements spécifiques et de modes de représentation que de domaines techniques concernés. Nous sortons là du domaine d'intérêt défini au paragraphe 1.1 : cela relève de la reconnaissance des documents techniques graphiques : plans de bâtiments, cadastres, cartes, schémas électriques et mécaniques de toutes sortes...

7. Reconnaissance industrielle et voies d'évolution

L'existence sur le marché de logiciels bureautiques performants et peu onéreux, comme *Omnipage*, *TextBridge* ou *Fine Reader* peut faire croire qu'il n'y a plus rien à étudier dans le domaine de la reconnaissance des documents imprimés. En réalité, ces logiciels commerciaux ne répondent qu'à une fraction des besoins : la reconnaissance de documents assez simples, d'un nombre de pages limité (quelques dizaines), sans prise en compte de la structuration, avec un taux d'erreur non contrôlé...

La problématique de la conversion rétrospective industrielle fait ressortir des exigences bien supérieures sur plusieurs points :

— documents techniques très complexes, incluant des tableaux, des formules mathématiques et chimiques, des schémas et graphiques... ;

— applications mettant en jeu des volumes de plusieurs millions de pages ;

— documents fortement structurés, dont la conversion nécessite la prise en compte de cette structure ;

— taux d'erreurs garantis pour certaines applications : la norme en édition est de moins d'un caractère en erreur sur 10 000 ; pour certaines applications critiques (nomenclatures d'une centrale nucléaire ou d'un avion), le zéro erreur est exigé.

Un logiciel comme *PRASAD* [2] est un élément de réponse aux exigences de reprise industrielle, mais ne résoud pas encore tous les problèmes.

Les voies de recherche et d'amélioration sont les suivantes :

— garantie du taux de reconnaissance et balisage des erreurs possibles ;

— reconnaissance des fontes utilisées dans les textes ;

— reconnaissance des textes de qualité très dégradée ;

— localisation et reconnaissance des formules mathématiques ;

— reconnaissance des tableaux : structure et contenu ;

— reconnaissance de la structure logique des documents : utilisation de modèles, aide à la constitution de ces modèles...

Certains travaux de recherche montrent que des solutions existent. Néanmoins, la mise en œuvre de nouvelles techniques dans les logiciels du commerce dépend plus des analyses du marché et de critères technico-économiques, que de l'état de l'art en matière de recherche.

8. Conclusion

La tâche de reconnaissance de documents, pour simple qu'elle paraisse à un lecteur humain, est d'une complexité énorme du point de vue de sa réalisation informatique. Les difficultés qu'elle présente, passées en revue dans cet article, ainsi que la grande variété des méthodes utilisées, dont plusieurs ont été présentées, le prouvent amplement.

Néanmoins, les algorithmes mis au point pendant près de quarante années de recherches, conjointement à l'augmentation des tailles mémoires et des puissances de calcul informatiques, ont permis d'aboutir à des logiciels qui donnent satisfaction dans un grand nombre de cas. Citons pour mémoire *Omnipage*, *TextBridge* et *Fine Reader* dans le domaine de la bureautique, *PRASAD* dans celui de la conversion rétrospective industrielle. Certes, toutes les difficultés ne sont pas encore surmontées, et il serait souhaitable d'apporter à ces logiciels plusieurs améliorations citées, mais on peut néanmoins affirmer que la reconnaissance de documents imprimés est aujourd'hui arrivée à maturité. Les techniques en cours de mise au point dans les laboratoires de recherche laissent augurer pour l'avenir des performances encore bien supérieures.

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite.
© Techniques de l'Ingénieur, traité Informatique

Reconnaissance de l'imprimé

Références bibliographiques

Ouvrages et articles généraux

[1] VAN HERWIJNEN (E.). – *Practical SGML.* Kluwer Academic Publishers (1992).

[2] LEFÈVRE (P.), FELTER (C.) et LOBBRECHT (P.). – *Reconnaissance de documents : passage du document papier à l'information électronique.* Revue Epure, EDF Direction des Études et Recherches n° 58 (1998).

[3] DREYFUS (J.) et RICHAUDEAU (F.). – *La chose imprimée.* Retz (1985).

[4] INGOLD (R.). – *Structures de documents et lecture optique : une nouvelle approche.* Presses polytechniques romandes (1990).

[5] *SGML-ODA : Présentation des concepts et comparaison fonctionnelle.* Afnor (1991).

[6] JACNO (M.). – *Anatomie de la lettre.* Compagnie française d'éditions (1978).

[7] VINCENT (P.). – *La lecture automatique des pages imprimées.* Revue documentaliste *25*, n° 4-5 (1988).

[8] BELAID (A.) et BELAID (Y.). – *Reconnaissance des formes - Méthodes et applications.* Inter Éditions (1991).

[9] BOKSER (M.). – *Omnidocument Technologies.* Proceedings of the IEEE, *80*, 7 (1992).

Ouvrages, thèses et articles spécialisés

[10] LEFÈVRE (P.), FELTER (C.) et CHRISTINE (J.-Y.). – *Prototype de reconnaissance de documents.* Note interne EDF, HN-46/93/136 (déc. 1993).

[11] TRINCKLIN (J.P.). – *Conception d'un système d'analyse de documents.* Thèse, université de Franche-Comté, Besançon (1984).

[12] BAIRD (H.S.). – *The skew angle of printed documents.* 4th annual conference on hybrid imaging systems (1987).

[13] POSTL (W.). – *Detection of linear oblique structures and skew scan in digitized documents.* 8th international conference on pattern recognition (1986).

[14] SAHOO (P.K.), SOLTANI (S.), WONG (A.K.C.) et CHEN (Y.C.). – *A survey of thresholding techniques.* Computer Vision, Graphics and Image Processing (1988).

[15] WHITE (J.M.) et ROHRER (G.D.). – *Image thresholding for optical character recognition and other applications requiring character image extraction.* IBM journal of research and development, *27*, 4 (1983).

[16] CHÉHIKIAN (A.). – *Binarisation d'images : deux solutions à ce problème.* Traitement du Signal, *6*, 1 (1989).

[17] AUBERT (M.). – *Système de binarisation optimale de documents.* Thèse, Institut national polytechnique de Grenoble (1991).

[18] AUBERT (M.), CHEHIKIAN (A.) et DELAPORTE (L.). – *Vers une binarisation optimale de documents.* 1st international conference on document analysis and recognition (1991).

[19] NADLER (L.). – *A survey of document segmentation and coding techniques.* Computer Vision, Graphics and Image Processing, *28* (1984).

[20] WAHL (F.M.), WONG (K.Y.) et CASEY (R.C.). – *Block segmentation and text extraction in mixed text/image documents.* Computer Vision, Graphics and Image Processing, *20* (1982).

[21] NAGY (G.) et SETH (S.). – *Hierarchical representation of optically scanned documents.* 7th international conference on pattern recognition (1984).

[22] LEFÈVRE (P.) et PEDRON (Y.). – *Document segmentation software implemented on a Transputer network.* 1st international conference on document analysis and recognition (1991).

[23] WANG (D.) et SRIHARI (S.N.). – *Analysis of form images.* 1st international conference on document analysis and recognition (1991).

[24] GREEN (E.) et KRISNAMOORTHY (M.). – *Model-based analysis of printed tables.* 3rd international conference on document analysis and recognition (1995).

[25] POYET (P.). – *Rapport d'étude sur la conception d'une fonctionnalité de reconnaissance de tableaux pour le prototype EDF-PRASAD.* Rapport interne EDF/DER - N46/1H9861 (déc. 1994).

[26] GAILLAT (G.) et BERTHOD (M.). – *Panorama des techniques d'extraction de traits caractéristiques en lecture optique de caractères.* Revue technique Thomson-CSF, *11*, 4 (1979).

[27] GOVINDAN (V.K.) et SHIVAPRASAD (A.P.). – *Character recognition - a review.* Pattern Recognition, *23*, 7 (1990).

[28] KAHAN (S.), PAVLIDIS (T.) et BAIRD (H.). – *On the recognition of printed characters of any font and size.* IEEE Transactions on Pattern Analysis and Machine Intelligence, *9*, 2 (1987).

[29] BAIRD (H.S.). – *Feature identification for hybrid structural/statistical pattern classification.* Computer Vision, Graphics and Image Processing, *42* (1988).

[30] ANIGBOGU (J.-C.). – *Reconnaissance de textes imprimés multifontes à l'aide de modèles stochastiques et métriques.* Thèse, université de Nancy-1 (1992).

[31] WAGNER (R.A.) et FISHER (M.J.). – *The string to string correction problem.* JACM, *21*, 1 (1974).

[32] BRADFORD (R.) et NARTKER (T.). – *Error correlation in contemporary OCR systems.* 1st international conference on documents analysis and recognition (1991).

[33] HANDLEY (J.C.) et HICKEY (T.B.). – *Merging optical character recognition outputs for improved accuracy.* Conférences sur la recherche d'informations assistée par ordinateur (1991).

[34] ZRAMDINI (A.) et INGOLD (R.). – *Optical font recognition using typographical features.* IEEE Transactions on Pattern Analysis and Machine Intelligence, *20*, 8, 877-882 (1998).

[35] KHOUBYARI (S.) et HULL (J.J.). – *Font and function word identification in document recognition.* Computer Vision and Image Understanding, *63*, 1 (1996).

[36] COOPERMAN (B.). – *Producing good font attribute determination using error-prone information.* SPIE, *3027* (1997).

[37] BELAID (A.) et HATON (J.P.). – *A syntactic approach for handwritten mathematical formula recognition.* IEEE Transactions on Pattern Analysis and Machine Intelligence, *6*, 1 (1984).

[38] TWAAKYONDO (H.M.) et OKAMOTO (M.). – *Structure analysis and recognition of mathematical expressions.* 3rd international conference on document analysis and recognition (1995).

[39] LEE (H.J.) et WANG (J.S.). – *Design of a mathematical expression recognition system.* 3rd international conference on document analysis and recognition (1995).

Dans les Techniques de l'Ingénieur

[40] CRETTEZ (J.-P.) et LORETTE (G.). – *Reconnaissance de l'écriture manuscrite.* [H 1 358], Documents numériques. Gestion de contenu (1998).

[41] WEINFELD (M.). – *Réseaux de neurones.* [H 1 990]. Technologies logicielles. Architectures des systèmes (1995).

Thèses

Le catalogue du système universitaire de documentation peut être consulté en ligne : http://www.sudoc.abes.fr

BEHEIM (L.). – *Coopération entre segmentation et reconnaissance des caractères imprimés dégradés.* Université de Paris-6 (2001).

LAVIROTTE (S.). – *Reconnaissance structurelle de formules mathématiques typographiées et manuscrites.* Université de Nice (2000).

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite. – ©Techniques de l'Ingénieur

POUR EN SAVOIR PLUS

Parution : mai 1999 - Ce document a ete delivre pour le compte de 7200082594 - sorbonne universite // 195.220.213.14

POUR EN SAVOIR PLUS

Revues spécialisées

En français :

Traitement du Signal
http://www.lis.inpg.fr/revue

En anglais :

IEEE Transactions on Pattern Recognition and Machine Intelligence
Proceedings of the IEEE
http://ieeexplore.ieee.org

Pattern Recognition

Pattern Recognition Letters

International Journal of Pattern Recognition and Artificial Intelligence

Machine Vision Application

Computer Vision, Graphics and Image Processing

Normalisation

BERNERS-LEE (T.) et CONNOLLY (D.). – *HyperText Markup Language Specification - 2.0.* RFC 1866, IETF (1995).
http://www.ietf.org/rfc/rfc1866.txt
http://www.w3.org/MarkUp/html-spec

Logiciels

***Fine Reader* Pro** (Windows), ABBYY (BitSoft)
http://www.abbyy.com

Omnipage Pro (Windows), TextBridge Pro Millenium (Windows), Scansoft
http://www.scansoft.com

EasyReader Elite (Windows), Neoptec
http://www.neoptec.com

Acrobat Capture (Windows), Adobe Systems France
http://www.adobe.fr

ExperVision (Windows), ExperVision Inc
http://www.expervision.com

Organismes

Laboratoires universitaires anglophones

University of Maryland at College Park - Laboratory for Language and Media Processing (site avec *bookmarks*)
http://documents.cfar.umd.edu

University of Nevada at Las Vegas - Information Science Research Institute
http://www.isri.unlv.edu

Buffalo University - Center of Excellence for Document Analysis and Recognition
http://www.cedar.buffalo.edu

University of Washington at Seattle - Intelligent Systems Laboratory
http://www.ee.washington.edu/research/isl

Laboratoires universitaires francophones

Université Laval (Montréal) - Département de Génie électrique et de Génie informatique
http://www.gel.ulaval.ca

Université de Fribourg - Département d'Informatique
http://www.unifr.ch/informatics

Laboratoire lorrain de recherche en informatique et ses applications (LORIA, Nancy)
http://www.loria.fr

Université de Rouen - Laboratoire Perception, Systèmes, Information (PSI)
http://psiserver.insa-rouen.fr/psi

Manifestations scientifiques

Françaises et francophones

Reconnaissance des Formes et Intelligence artificielle (RFIA)

Colloque international francophone sur l'Écrit et le Document (CIFED)

Internationales

International Conference on Document Analysis and Recognition (ICDAR)

Symposium on Document Analysis and Information Retrieval (SDAIR)

International Conference on Pattern Recognition (ICPR)

International Conference on Image Processing (ICIP)

Workshop on Document Analysis Systems (DAS)

Autres

Scandinavian Conference on Image Analysis (SCIA)

Portuguese Conference on Pattern Recognition (RECPAD)

Toute reproduction sans autorisation du Centre français d'exploitation du droit de copie est strictement interdite. – © Techniques de l'Ingénieur

Parution : mai 1999 - Ce document a été délivré pour le compte de 7200082594 - sorbonne université // 195.220.213.14

TECHNIQUES DE L'INGÉNIEUR

GAGNEZ DU TEMPS ET SÉCURISEZ VOS PROJETS EN UTILISANT UNE SOURCE ACTUALISÉE ET FIABLE

Techniques de l'Ingénieur propose la plus importante collection documentaire technique et scientifique en français !

Grâce à vos droits d'accès, retrouvez l'ensemble des **articles et fiches pratiques de votre offre, leurs compléments et mises à jour,** et bénéficiez des **services inclus.**

RÉDIGÉE ET VALIDÉE PAR DES EXPERTS	MISE À JOUR PERMANENTE	100 % COMPATIBLE SUR TOUS SUPPORTS NUMÉRIQUES	SERVICES INCLUS DANS CHAQUE OFFRE

- **+ de 350 000 utilisateurs**
- **+ de 10 000 articles de référence**
- **+ de 80 offres**
- **15 domaines d'expertise**

- **Automatique - Robotique**
- **Biomédical - Pharma**
- **Construction et travaux publics**
- **Électronique - Photonique**
- **Énergies**
- **Environnement - Sécurité**
- **Génie industriel**
- **Ingénierie des transports**
- **Innovation**
- **Matériaux**
- **Mécanique**
- **Mesures - Analyses**
- **Procédés chimie - Bio - Agro**
- **Sciences fondamentales**
- **Technologies de l'information**

Pour des offres toujours plus adaptées à votre métier, découvrez les offres dédiées à votre secteur d'activité

Depuis plus de 70 ans, Techniques de l'Ingénieur est la source d'informations de référence des bureaux d'études, de la R&D et de l'innovation.

www.techniques-ingenieur.fr

CONTACT : Tél. : + 33 (0)1 53 35 20 20 - Fax : +33 (0)1 53 26 79 18 - E-mail : infos.clients@teching.com

TECHNIQUES DE L'INGÉNIEUR

LES AVANTAGES ET SERVICES compris dans les offres Techniques de l'Ingénieur

ACCÈS

Accès illimité aux articles en HTML
Enrichis et mis à jour pendant toute la durée de la souscription

Téléchargement des articles au format PDF
Pour un usage en toute liberté

Consultation sur tous les supports numériques
Des contenus optimisés pour ordinateurs, tablettes et mobiles

SERVICES ET OUTILS PRATIQUES

Questions aux experts*
Les meilleurs experts techniques et scientifiques vous répondent

Articles Découverte
La possibilité de consulter des articles en dehors de votre offre

Dictionnaire technique multilingue
45 000 termes en français, anglais, espagnol et allemand

Archives
Technologies anciennes et versions antérieures des articles

Impression à la demande
Commandez les éditions papier de vos ressources documentaires

Alertes actualisations
Recevez par email toutes les nouveautés de vos ressources documentaires

*Questions aux experts est un service réservé aux entreprises, non proposé dans les offres écoles, universités ou pour tout autre organisme de formation.

ILS NOUS FONT CONFIANCE

AgroParisTech · AIRBUS · cea · CESI Enseignement Supérieur et Formation Professionnelle · cnes Centre National d'Études Spatiales · CSTB le futur en construction · ENGIE · GE · L'ORÉAL · Leroy-Somer · École Polytechnique Université Paris-Saclay · SAFRAN Aerospace Defence Security · Sagemcom · SAINT-GOBAIN · Schneider Electric · THALES · TOTAL · Valeo · VEOLIA · Wallonie · SPW Service public de Wallonie

www.techniques-ingenieur.fr

CONTACT : Tél. : + 33 (0)1 53 35 20 20 - Fax : +33 (0)1 53 26 79 18 - E-mail : infos.clients@teching.com