Rapport "Technique" interne

LAFC Système de Lecture Automatique de Formules Chimiques par extraction et reconnaissance incrémentales d'objets

Jean-Yves Ramel - Guillaume Boissier – H. Emptoz

1. Introduction

Ce rapport décrit le travail réalisé dans le cadre d'une collaboration entre la société *XXXX* et le laboratoire RFV. L'objectif de l'interprétation automatique de documents est de faciliter l'exploitation des dessins imprimés ou manuscrits en s'appuyant sur un ensemble d'étapes permettant la transformation du schéma papier en un document numérique qui puisse être interprété automatiquement. Beaucoup de travaux concernent l'analyse de documents imprimés tels que les sommaires de périodiques, les pages de journaux, les partitions musicales, ... mais très peu traitent de l'interprétation des documents manuscrits. Les difficultés qui proviennent de la diversité des types d'informations (texte, traits, formes, ...) constituant ces documents, de leur manque de normalisation (grande liberté du dessinateur), expliquent probablement l'intérêt limité porté à ces documents jusqu'à aujourd'hui.

Cette remarque doit permettre de prendre conscience, dès à présent, de l'originalité et des difficultés auxquels va se confronter cette étude.

Le résultat de ce travail est une maquette permettant la lecture automatique de formules chimiques. Ce système opère selon 2 phases distinctes (figure 1) : l'une dite de **perception globale** permettant l'acquisition d'une représentation du document, l'autre dite de **lecture et de compréhension**.

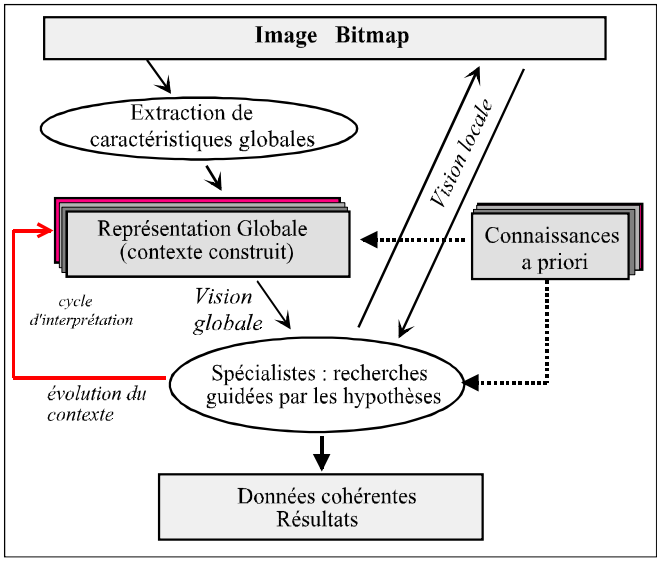


Figure 1 : Architecture de la maquette proposée

Ce rapport fournit, dans les deux parties suivantes, une description détaillée des deux phases : extraction des caractéristiques globales et interprétation du document que comporte cette approche.

Il nous paraît nécessaire, avant de commencer tout développement ou mise en place de méthodologie de lecture automatique, de décrire d'abord qu'elle est la structure d'un dessin et de montrer ensuite ce qui a déjà été réalisé concernant ce type de problème. La question étant par ailleurs encore loin d'être complètement résolue, nous nous attacherons à mettre en lumière les problèmes existants.

1. Du dessin à l'image

Un document a normalement pour but de transmettre un ensemble d'informations à des destinataires. Traditionnellement le support de communication de ces informations est le papier. Les documents peuvent présenter différents aspects : formulaires, documents composites, dessins techniques, ... Chaque type possède une structure qui lui est propre, choisie de manière à ce que les informations qu'il véhicule soient aisément compréhensibles et réutilisables par les destinataires.

Parmi les documents, les dessins forment une catégorie particulière; ils sont constitués de lignes, de régions pleines, de régions hachurées, de texte, ... Ils contiennent toujours une quantité importante d'informations et peuvent être d'une grande complexité et leur domaine d'application est très étendu.

Les organigrammes et autres dessins

Cette catégorie englobe les schémas représentant une organisation (organigrammes de programmes, modélisation de données, représentation d'une hiérarchie d'entreprises, ...) et les documents de type tableau (figure 3).

Les organigrammes sont caractérisés par la présence de nombreuses zones de texte. Les formes à reconnaître sont généralement des polygones, lignes, cercles, et autres objets géométriques élémentaires.

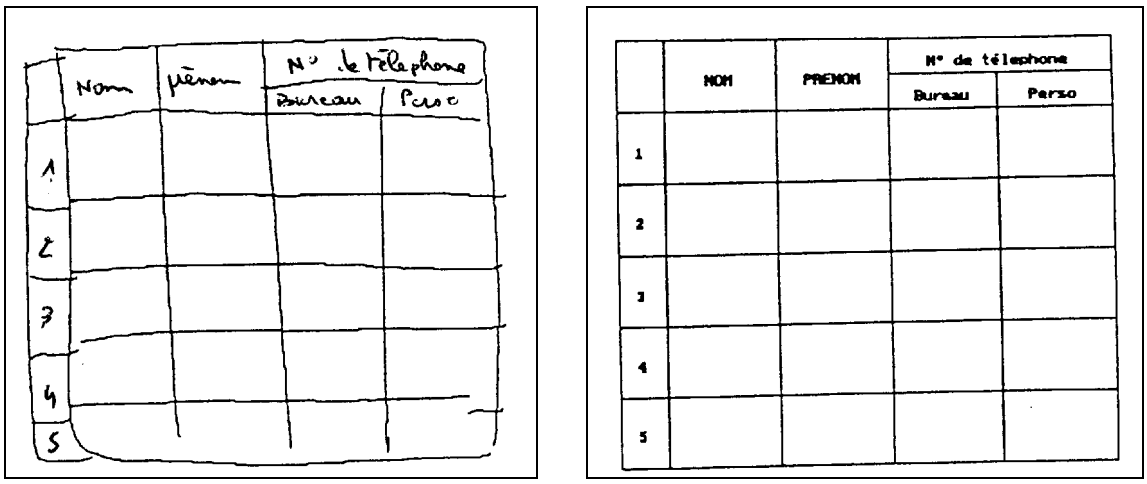


Figure 3 : Exemple d'esquisse de tableau et son idéalisation

Les objectifs sont donc l'extraction du texte et l'extraction d'attributs relatifs aux objets élémentaires afin de pouvoir fournir une description des relations spatiales existant entre les composants (inclusion, lien, intersection, ...). Il s'agit parfois de dessins faits à main levée qui constituent, après remise en forme, l'entrée d'un système d'édition de graphiques (dessins ou tableaux).

Il est à noter que tous ces dessins peuvent se présenter sous différents aspects. Ils peuvent avoir été *dessinés à main levée* comme cela est le cas pour les **formules chimiques** ou, au contraire, à l'aide d'instruments graphiques (règles, normographes, ...) et peuvent avoir ou non subi des dégradations (photocopies successives, ...). *La qualité* d'une image se définit alors sur des critères d'orientation, de bruits (taches noires parasites ou blancs induisant des discontinuités), et de distorsion des formes. Face à ces défauts, dus aux techniques d'acquisition, mais aussi à la mauvaise qualité du document original lui-même, les méthodes choisies devront être plus ou moins robustes et il conviendra donc de bien définir le type et la nature des documents traités par notre système (grâce à un ensemble d'images tests).

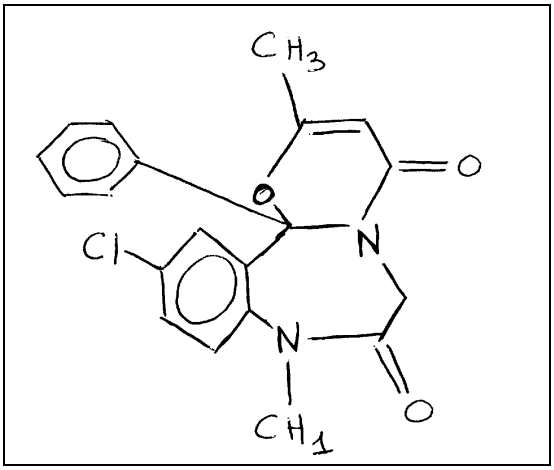
2. Spécificité de la lecture de formules chimiques

La complexité des images de symboles chimiques n'atteint pas celle des projections orthogonales, des schémas cinématiques ou des schémas électriques. Elles ne comportent généralement que des segments de droite, des cercles, et du texte en quantité "raisonnable". Les difficultés de reconnaissance automatique proviennent plutôt du fait de la réalisation à main levée de ces dessins (sur une tablette graphique dans notre cas). Cependant, l'aspect temporelle du tracé est un paramètre important (obtenu si on utilise une tablette graphique - on line) qui pourra aussi être utilisé durant la phase d'analyse et de reconnaissance.

Les symboles chimiques obéissent à des règles syntaxiques et sémantiques qui pourront aussi être utilisées pendant l'analyse de l'image et qui permettront la vérification des résultats fournis par le système de lecture automatique.

Les architectures classiques de lecture de symboles chimiques [Casey93] comportent donc généralement un module de vectorisation (extraction des segments de droite), un module d'extraction du texte et un module d'extraction des courbes ou cercles. Après localisation des zones de texte, celles-ci sont soumises à un module d'OCR. Au niveau supérieur, on trouve un module d'agrégation/reconnaissance (des symboles) et un module de vérification de la cohérence des données extraites.

La qualité fluctuante des tracés oblige la mise en place d'algorithmes de redressement et d'amélioration des tracés avant ou pendant la phase de reconnaissance. Les travaux réalisés dans ce domaine [Poirier93, Galindo97] soulignent d'ailleurs la difficulté de cette tâche même pour un lecteur humain !



Exemple de formule chimiques

Les techniques classiques de vectorisation, de localisation du texte et de localisation des courbes (décrites ci dessous) pourront être utilisées seulement après leurs modifications afin qu'elles prennent en compte l'aspect fluctuant des tracés à main levée.

3. Les techniques d'analyse

3.1. Analyse d'un dessin sur la base de sa structure physique

L'autre point commun entre ces différents dessins concerne les formes des objets à extraire des images. Tout système d'analyse de documents techniques manipule **des entités de type structurel** formées à partir de segments de droite et d'arcs tels que lignes, polygones, cercles, courbes. D'ailleurs, dans la plupart des systèmes, une étape de vectorisation permet d'obtenir une description de l'image sous forme de suites de vecteurs plutôt que sous forme de pixels. Le premier pas vers une description, puis vers une interprétation de type structurel, est ainsi réalisé. Il peut être suivi d'une phase d'identification d'entités géométriques simples, obtenues par combinaison des vecteurs de base, telles que les polygones.

Le problème de la lecture automatique de dessins semble donc résider dans la phase de séparation puis de reconnaissance de chacune de ces couches, et dans le choix de l'ordre d'extraction le plus approprié. Cela ne va pas sans difficulté puisque, comme nous venons de le voir, toutes les couches sont constituées d'objets géométriques eux-mêmes formés à partir de primitives d'un même type (arcs et segments).

La plupart des systèmes d'analyse de documents techniques réalisent, en premier lieu, une identification des formes (composantes connexes, formes fines, formes pleines, ...) présentes dans l'image [Kasturi92]. Les objets constituant les dessins sont ainsi regroupés suivant les méthodes de traitement que l'on pourra leur appliquer par la suite. Une fois cette séparation réalisée, chaque couche continue à être traitée indépendamment, par l'intermédiaire de méthodes variées (vectorisation, détection de contours, maillage, ...), chacune étant choisie en fonction du type de forme à extraire (figure 4).

Cette façon d'aborder le problème peut être suffisante pour certaines applications, notamment pour les organigrammes ou pour tout autre document comportant simplement une couche Texte et une couche Formes Fines (ou un nombre limité de couches). La reconstruction des entités (géométriques) est alors réalisée assez simplement si chacune de ces couches peut être analysée indépendamment et si les formes à regrouper ne sont pas dispersées dans différentes couches.

Au plus bas niveau se trouvent les algorithmes opérant directement sur les pixels de l'image pour en extraire des informations plus structurées, telles que suivi de contours, vectorisation, filtrage, ... Des outils spécifiques, dédiés à cette *phase d'analyse lexicale*, ont été développés de façon à traiter chaque forme de la manière la plus adéquate possible.

Analyse morphologique

Certains proposent de privilégier l'extraction des zones d'intérêt au moyen de traitements de type morphologique (ouverture, érosion/dilatation) pour éliminer les formes fines d'un dessin. Kasturi utilise ce principe pour localiser et extraire les régions pleines d'un document [Kasturi90]. Une succession d'érosions permet d'éliminer les segments, arcs et symboles de l'image. Reste, ensuite, à redonner aux objets encore présents dans l'image leur forme initiale par dilatations successives. La dilatation n'étant pas exactement l'opération inverse de l'érosion, des traitements supplémentaires sont encore nécessaires pour obtenir une restitution exacte des symboles pleins originels.

La vectorisation

En règle générale le recours au codage du contenu de l'image est indispensable afin de structurer, pour ensuite traiter et décrire l'image. Cette transformation doit posséder les caractéristiques suivantes :

- conservation de l'information intéressante contenue dans l'image,
- réduction de la place nécessaire au stockage,
- simplification et adaptabilité du nouvel espace de représentation au traitement à réaliser.

Il est difficile de trouver une représentation donnant pleine satisfaction à ces trois contraintes. De manière pratique on choisit un compromis en privilégiant les aspects importants du problème traité. Comme nous l'avons vu précédemment, l'une des caractéristiques principales des documents techniques est qu'ils sont constitués essentiellement à partir de deux primitives élémentaires : le *segment de droite* et l'*arc*.

Nous allons donc voir les différentes méthodes permettant d'exploiter cette structure spécifique pour passer de la structure matricielle de l'image à une représentation plus adaptée aux dessins étudiés.

La représentation de l'image sous forme de vecteurs est la plus fréquemment utilisée, même si l'on a parfois fait appel à d'autres techniques. Ce passage de la forme matricielle de l'image à une description sous forme de vecteurs se nomme *vectorisation.*

De nombreux systèmes effectuent d'abord une squelettisation pour obtenir des segments, des arcs et des courbes (primitives élémentaires) ayant une épaisseur de 1 pixel. Les traitements suivants se trouvent ainsi simplifiés. Cependant, la squelettisation pose certains problèmes comme l'apparition de barbules, la perte d'information (épaisseur du trait), ou l'inadaptabilité au traitement des formes pleines.

Abe [Abe86] propose une méthode de suivi de contours adaptée à l'analyse des organigrammes. Grâce à cette technique, les boucles fermées, utiles par la suite, sont extraites durant la vectorisation. Cugini [Cugini84] emploie une technique d'appariement de contours pour obtenir une représentation, sous forme de vecteurs, des projections orthogonales. D'autres systèmes couplent l'information contours avec l'information squelette pour améliorer la qualité des résultats [Tanigawa94]. Il ressort de ces études que la mise en correspondance des contours est moins sensible au bruit que la squelettisation mais que sa mise en place est beaucoup plus complexe.

Analyse par les composantes connexes

De nombreux systèmes d'analyse de documents techniques effectuent un étiquetage des **composantes connexes**. L'analyse des composantes détectées procure d'importantes informations pour la suite du traitement, plus particulièrement pour l'extraction des composantes texte ou l'extraction de certains symboles électriques. La Transformée de Hough permet de regrouper en mots ou phrases les composantes connexes alignées correspondant (parfois!) aux caractères. Les caractères imprimés correspondent, lorsqu'ils ne touchent pas une autre partie du dessin, aux petites composantes connexes. Des traitements complexes et variés peuvent alors être réalisés pour, d'une part supprimer les composantes ne correspondant pas à des caractères (pointillés, ...), et d'autre part localiser les caractères manquants. Desseilligny [Desseilligny95] étudie très précisément tous les voisinages des composantes connexes de petites tailles pour les regrouper en mots. Pour cela il utilise des critères de taille, de voisinage, d'orientation et effectue même une reconnaissance de la police.

Des techniques ont été développées pour extraire les caractères attachés aux graphiques en minimisant à la fois l'altération du caractère et celle du graphique [Joseph91].

Analyse syntaxique

Interviennent ensuite, durant une *phase d'analyse syntaxique*, les algorithmes qui effectuent une étude des données fournies par le niveau inférieur : segmentation (regroupement/séparation), calcul d'attributs, ... en vue de la reconnaissance. Pour cette phase, les travaux réalisés sur les schémas électriques sont à rapprocher de ceux effectués sur les plans cadastraux. La localisation des entités à reconnaître se base sur l'étude des vecteurs (issus de la vectorisation) : recherche des boucles fermées (polygones) et sur l'étude de la position des zones de texte. Pour traduire les relations entre les primitives de bas niveau, Antoine [Antoine92] préconise l'utilisation de modèles hiérarchiques d'objets. Après la localisation, de nombreuses méthodes à base de graphes structurels permettent de mettre en correspondance les modèles d'une base de données avec les formes détectées. Il se pose malheureusement rapidement des problèmes d'explosion combinatoire. L'utilisation d'heuristiques permet, dans certains cas, de résoudre ces problèmes par génération d'hypothèses (boucles fermées = symboles dans les schémas électriques) [Habacha93b].

3.2. Interprétation se basant sur la structure logique

Peu à peu, les techniques d'interprétation de documents techniques ont évolué et la décomposition en couches n'a plus été réalisée en fonction de critères géométriques (couches des traits forts, couches des traits fins, ...), c'est à dire des traitements à mettre en place, mais plutôt sur des *critères sémantiques*, c'est à dire en fonction du sens des informations fournies par les regroupements judicieux des formes de l'image, en utilisant sa structure logique.
Il est à noter qu'avec ce principe, seules les méthodes d'extraction des primitives de base et les modèles de description des objets constituant les dessins peuvent être communs à tous les systèmes d'interprétation. L'ordre d'extraction des entités dépend lui du domaine d'application. Ainsi, il est possible de décrire la structure logique des différents types de documents :

- Pour les documents de type projections orthogonales, on distingue généralement une couche Cotation (Texte + lignes de référence + flèches), une couche Contours visibles, une couche Contours cachés, une couche Axes de symétrie, ...

- Pour les schémas de principe (et les schémas cinématiques) on distingue le plus souvent une couche Légendes/cotations, une couche Symboles, et une couche Liens de connexion.

- Pour les documents de type plans, les couches à extraire sont, par exemple, la couche Réseaux routiers, la couche Immeubles, la couche Canalisations, ... pour obtenir des cartes dites "géocodées".

La stratégie d'interprétation doit se baser sur ces connaissances et, de plus, les exploiter dès le bas niveau, aussi bien que dans tous les niveaux de l'analyse, afin d'employer des outils spécifiques et adaptés durant tout le traitement.
C'est en tenant compte de ce type d'information qu'il faut définir, par exemple, l'ordre d'extraction des différentes entités constituant un dessin, ou essayer de lever certaines ambiguïtés concernant l'appartenance d'une primitive à une entité plutôt qu'à une autre. On limite ainsi les incohérences et leur propagation dans les niveaux supérieurs.

Les différents documents obéissent tous à des règles strictes, définies par différentes normes, qui précisent quelles entités et quelles relations entre entités sont susceptibles d'apparaître. On ne parle plus alors de structure mais plutôt de syntaxe propre au schéma. Ces règles varient selon le type de documents et d'applications auxquels on s'intéresse : les bases de données d'un système de CAO mécanique sont totalement différentes de celles d'un système de simulations de circuits électriques ou même de celles d'un système de CAO d'un autre type. La connaissance de ces différentes règles permet de vérifier la cohérence de la représentation obtenue.

De nombreuses méthodes de reconnaissance (structurelles, syntaxiques, statistiques, ...) peuvent être appliquées. Cependant, il est préférable de leur adjoindre des programmes d'Intelligence Artificielle, fondés sur des bases de connaissances spécifiques au domaine considéré. L'*analyse sémantique* peut en effet ainsi venir en aide à l'analyse syntaxique.

Le système ANON, développé au sein de l'équipe de Joseph [Joseph92] utilise une grammaire de type LR1 pour gérer les connaissances. Les règles de cette dernière ont pour but de contrôler l'activation des mécanismes de détection des primitives de bas niveau. Dans le même esprit, Habacha [Habacha93a] définit des liens contextuels entre le texte, les symboles, et les liaisons constituant les schémas électriques. La détection d'une entité peut alors déclencher des recherches d'entités d'un autre type dans son voisinage.

4. Les problèmes qui subsistent

Même si ces travaux ont permis, comme on vient de le voir, de nombreuses avancées en ce qui concerne les traitements de bas niveau et l'extraction de primitives, de nombreuses difficultés subsistent encore :

- Les formes pleines ainsi que certaines jonctions ou extrémités sont altérées durant la squelettisation (préalable à la vectorisation).

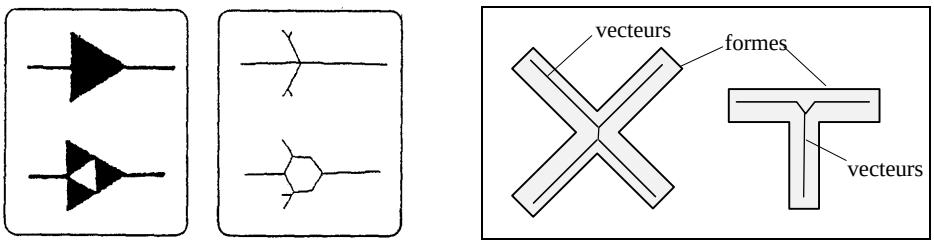


Figure 6 : Problèmes lors de la squelettisation

• Quelle que soit la technique de vectorisation, le résultat de l'approximation polygonale n'est pas toujours celui escompté. Les points critiques (points de contrôle) fournis ne sont pas toujours représentatifs de la forme initiale comme on peut le voir sur la figure suivante.

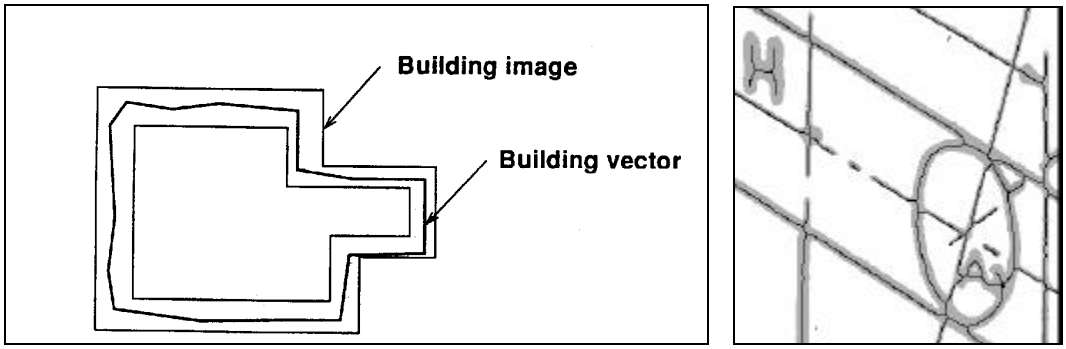


Figure 7 : Problèmes lors de l'approximation

• La séparation des caractères du graphique pose des difficultés lorsque la taille des composantes connexes n'est plus significative : c'est à dire quand un caractère est connecté à une autre partie du dessin ou quand il s'agit de manuscrit.

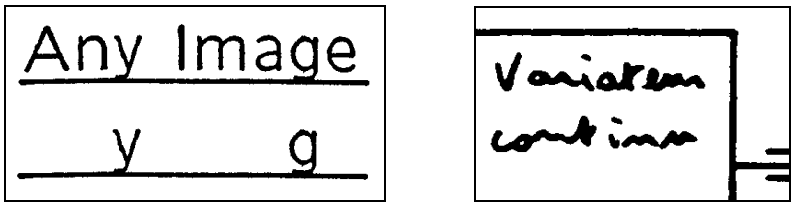


Figure 8 : Problèmes lors de la localisation

• Les caractères isolés sont également très difficilement localisables et sont souvent confondus avec d'autres petits symboles. Les pointillés ne doivent pas être confondus avec les caractères.

• Peu de travaux s'étendent sur l'extraction et le codage des courbes, cercles et autres formes non rectilignes si nombreuses dans le manuscrit ainsi que sur leur mise en relation avec les autres parties du document (liaison entre une droite et une courbe).

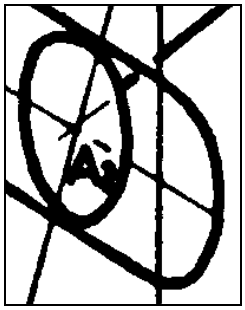


Figure 10 : Problème de représentation

De même, une fois les primitives de base extraites, les systèmes d'interprétation de documents complexes se heurtent à d'autres difficultés.

• Après la séparation des différentes couches que comporte un schéma, il devient très difficile d'effectuer la reconstruction et la reconnaissance d'entités de haut niveau puisque leurs composants peuvent être dispersés dans plusieurs des couches. Les représentations engendrées sont très différentes les unes des autres, la comparaison et la mise en correspondance des résultats obtenus pour chaque couche deviennent quasiment impossibles. La plupart des réalisations ne fournissent pas des solutions satisfaisantes; les recherches s'interrompent donc à ce niveau en raison d'une interprétation isolée de chacune des couches.
Pourtant, beaucoup de travaux indiquent que cette segmentation en couches doit obligatoirement être réalisée avant toute autre analyse puisque les traitements d'extraction sont spécifiques à chaque type de forme qu'il est possible de rencontrer dans de tels documents; ils estiment qu'aucun traitement universel n'est applicable.

• Il me semble que les retours en arrière sont pratiquement irréalisables sans remise en cause de la totalité de l'analyse. Les travaux de Ogier [Ogier94b] sur l'analyse de la cohèrence vont aussi dans ce sens; les vérifications sont réalisées après chaque construction d'objet élément du cadastre pour éviter la propagation des erreurs.

• La soustraction des entités reconnues (hachures, textes, ...) de l'image initiale ou de son modèle de représentation pose encore beaucoup de problèmes. Ces entités rendent pourtant plus difficile la détection des autres données.

• La prise en compte des données extraites est ardue : la gestion du contexte doit permettre de générer des hypothèses c'est-à-dire de faciliter l'utilisation des connaissances obtenues et

d’augmenter les possibilités de coopération entre les processus d’analyse. Les techniques de reconnaissance pure deviennent alors insuffisantes et doivent être complétées. L'adjonction d'artifices doit permettre la prise en compte du *contexte* de l'image et du *but* poursuivi.

2. L'approche globale du dessin

Les vecteurs

Une forme binaire est décrite de manière équivalente par son contour ou par les pixels de la région noire, qui la constituent. Une information sur sa direction peut être obtenue par une étude locale de ses contours. Quant à l’épaisseur, qui correspond à la distance entre les deux frontières du trait, elle est surtout intéressante pour l'étude des traits. Ces raisons nous ont conduits à choisir la description des formes par les contours au moyen de la primitive Vecteur (figure 2). Ainsi, une approximation polygonale des contours des formes constituant l'image initiale fournira une suite ordonnée (par le suivi des contours) de vecteurs. La suite est notée *(SV)*.

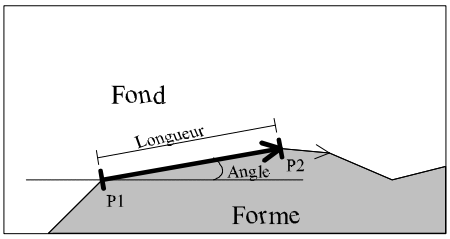


Figure 2 : Objet Vecteur

Les segments obtenus sont stockés dans une liste chaînée en respectant l’ordre fourni par le suivi des contours. Ensuite, pour améliorer les performances de sélection des points de contrôle et diminuer la variabilité suivant le seuil choisi, on étudie les frontières à différents niveaux de détails. Pour cela, la phase d'approximation polygonale est opérée de manière itérative (en utilisant toujours le même algorithme d’approximation) sur les points de contrôle nouvellement obtenus.

Le nombre de points de contrôle est ainsi réduit par fusion de certains des segments obtenus lors des étapes successives. Ce procédé est répété jusqu'à stabilisation, c'est-à-dire jusqu’à ce que plus aucune fusion ne soit possible. Le nombre d’itérations avant stabilisation varie généralement entre 2 et 5 suivant les caractéristiques du contour (courbure, manuscrit ou imprimé, ...).

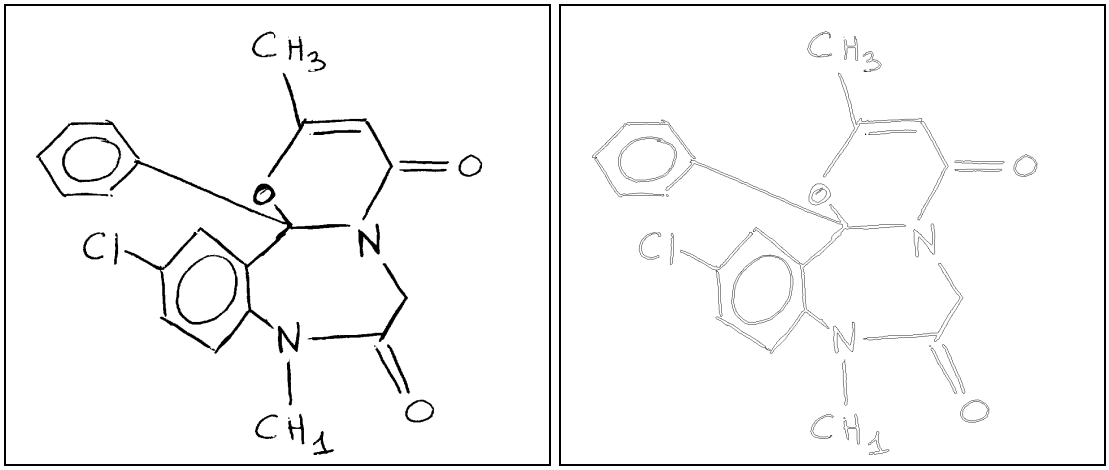


Figure 3 : Approximation des contours par chaînes de Vecteurs

La représentation de l'image sous forme de chaînes de *Vecteurs* correspondant aux contours des formes permet de limiter la perte d'information et donne une approximation très fidèle du dessin (figure 3), mais elle n'est pas facilement exploitable et ne fournit pas suffisamment d'information sur la structure du document. On obtiendrait alors seulement une vectorisation des contours. La structuration supplémentaire décrite ci-dessous permet de faire évoluer cette représentation de manière hiérarchique et d'acquérir des renseignements intéressants.

Les Quadrilatères

Pour obtenir et gérer aisément des données supplémentaires, il a fallu définir un outil de description plus évolué. La nature des formes à extraire nous a incité à choisir le *Quadrilatère.* Une étape d'appariement des *Vecteurs* de la suite SV permet la mise en place des primitives *Quadrilatères.* Chaque *Quadrilatère* est défini par une paire de *Vecteurs* (chacun appartenant à l'une des deux frontières opposées d'une forme fine). Le Quadrilatère étant une primitive plus souple, il n'est pas nécessaire, dans notre cas, d'effectuer, au préalable, l'extraction du texte contenu dans l'image. Afin d'obtenir une description plus robuste des formes fines, la construction des Quadrilatères s'effectue en plusieurs étapes :

- Appariement des *Vecteurs* (éléments de SV) pour construire les *Quadrilatères* et mise à jour de SV (ensemble des vecteurs non appariés).
- Tri des *Quadrilatères* suivant leur proximité de manière à reconstruire une « chronologie » du tracé
- Fusion de certains *Quadrilatères* voisins

Pour augmenter la robustesse de l'algorithme d'appariement, on commence par sélectionner le vecteur V1 de longueur maximum dans la liste des vecteurs encore non appariés, puis on recherche le vecteur le plus proche (suivant la distance euclidienne entre points) de chacune des extrémités de V1 et qui vérifie certains critères. Pour que la mise en correspondance de 2 vecteurs ait lieu, des conditions (critères d'appariement), correspondant chacune à une propriété physique des traits, doivent être remplies. Ces critères ont été choisis "empiriquement" mais en respectant la définition intuitive d'un trait :

- un vecteur ne peut être apparié qu'avec le vecteur le plus proche de lui (la distance entre les extrémités des 2 vecteurs doit être inférieure à 60 pixels)
- la différence entre les directions (pentes) des 2 vecteurs doit être inférieure à Π/8
- présence de pixels noirs entre les extrémités (P1Q2 et P2'Q1 dans la figure 8)
- les vecteurs doivent être de sens opposé
- la longueur du 1er vecteur doit être supérieure aux épaisseurs (e1 et e2 calculées à partir des extrémités)

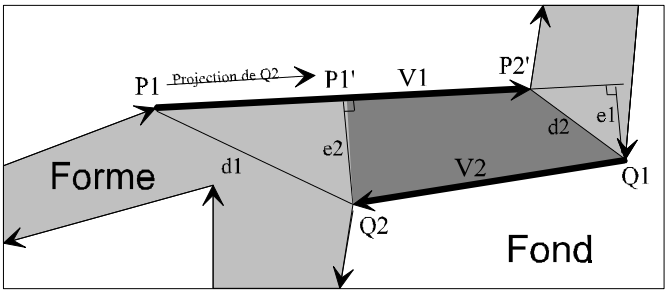


Figure 4 : Appariement des Vecteurs

Pour améliorer l'appariement, certains vecteurs peuvent être décomposés en 2 vecteurs fils, le point de coupure est obtenu par projection d'une extrémité d'un vecteur sur le support de l'autre (figure 4). Les longueurs des deux vecteurs constituant le *Quadrilatère* sont, de cette manière, plus semblables. Le second vecteur fils reste dans la liste des vecteurs à étudier. Le processus d'appariement s'arrête lorsque plus aucune mise en correspondance n'est réalisable (selon les critères d'appariement imposés).

Du fait de la localisation des points de contrôle (provenant de l'approximation polygonale) et de la méthode d'appariement, une phase de fusion des *Quadrilatères* est nécessaire pour obtenir des objets se rapprochant au mieux des traits effectifs sur l'image (*Quadrilatères* de dimensions optimales et en nombre réduit). Cette fusion sera effectuée après une étape de tri des *Quadrilatères* selon leur proximité.

L’ensemble des objets *Quadrilatères* finalement obtenu est représentatif des formes fines présentes dans le document initial ; il ne faut cependant pas voir cette étape comme une véritable étape de vectorisation du document mais plutôt comme la méthode que nous avons choisie pour obtenir une description initiale des formes fines d’un document à l'aide d'une primitive très simple à manipuler.

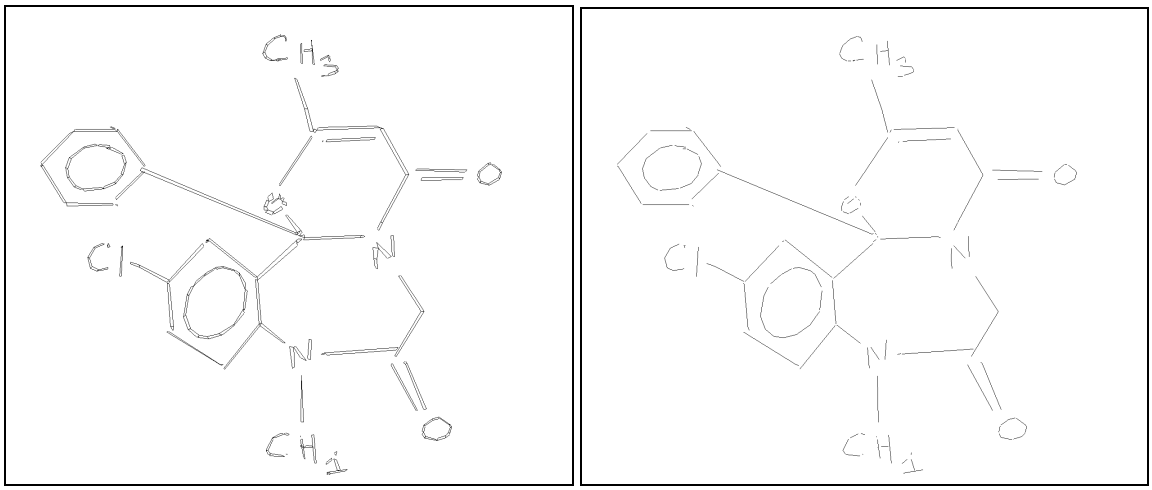


Figure 5 : Image des Quadrilatères

Les vecteurs ne pouvant être mis en correspondance avec aucun autre, correspondent aux contours des formes pleines.

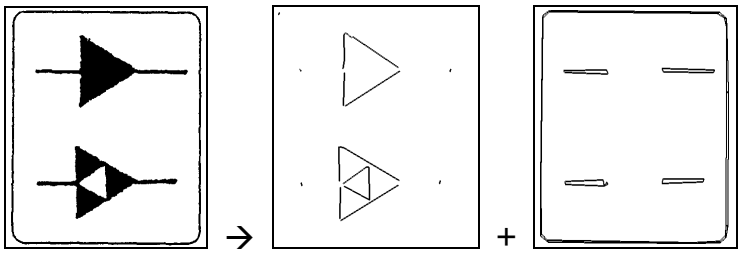


Image initiale → Vecteurs + Quadrilatères

Figure 6 : Représentation d’une image

3. Mécanisme d’interprétation

L'interprétation du document est réalisée par différents **spécialistes**. Tous utilisent la représentation construite pour focaliser leur attention sur des zones précises de l’image (génération d'hypothèses) qu’ils étudient en détail (vérification) pour faire avancer l’analyse.

Pour effectuer ce travail d'exploration-confrontation, les spécialistes possèdent des compétences élémentaires, précises et suffisantes dans des domaines particuliers comme l'extraction du texte, des liaisons, des polygones, ...

Les différents spécialistes

Du fait des types de tâches à effectuer et des connaissances utiles pour réaliser correctement chacune d'elles, nous avons, dans le cas des formules chimiques, eu recours aux spécialistes cités ci-dessous.

- Le premier spécialiste est chargé d'extraire les zones de **texte**. Dans notre représentation, la taille, la forme et la disposition des quadrilatères sont représentatives de la texture de l'image initiale. Les zones de texte sont caractérisées par des regroupements de quadrilatères courts. Ils sont détectés par leur proximité physique et sans tenir compte de leur orientation. Bien que d'autres formes produisent des quadrilatères ayant des caractéristiques semblables, une analyse locale de ces groupes de quadrilatères et de leur voisinage permet la construction de zones de focalisation (susceptibles de contenir du texte manuscrit ou imprimé), ceci sans avoir recours aux composantes connexes. A partir de chaque amas de quadrilatères, il est possible de générer une **composante de texte virtuelle** (zone de focalisation) constituée par le rectangle englobant l'ensemble des quadrilatères regroupés pour former l'amas ; la figure 7 montre quelques exemples de zones de focalisation.

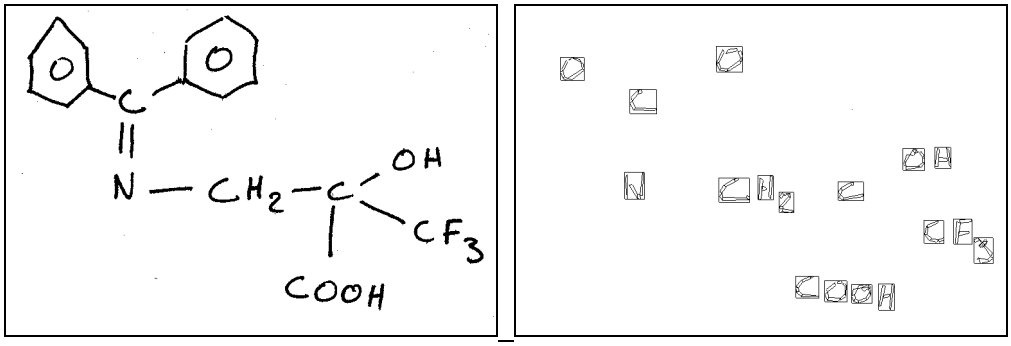


Figure 7 : Composantes de texte virtuelles

En couplant cette information avec celle fournie par l'étude des composantes connexes présentes dans le document initial, le spécialiste prend une décision et classe chaque zone dans l'une des catégories suivantes : texte, non texte, indécision.

Dans les zones d'indécision, en étudiant la taille et la position des composantes de texte virtuelles par rapport aux zones de texte déjà localisées, on procède à une confrontation avec le contexte, et certaines ambiguïtés peuvent alors être levées. Enfin dans les cas où le doute subsiste, on peut faire appel au module d'OCR ou, dans un système interactif, interroger l'utilisateur pour prendre d'autres décisions.

- Le second spécialiste est chargé de **reconnaître** le contenu des zones de texte préalablement extraites. Le fonctionnement de ce spécialiste étant complexe, sa description est faite dans le chapitre 4 (qui lui est entièrement consacré).

- Un spécialiste peut être chargé de rechercher les **formes pleines**. Dès que les quadrilatères ont été construits, la suite (SV) ne contient plus que les vecteurs pour lesquels la mise en correspondance a échoué ; la plupart de ces vecteurs (de longueurs significatives) appartient alors au contour d'une forme pleine. Ainsi, les formes pleines peuvent être localisées par recherche de chaînes de contour quasi-fermées dans la suite (SV) et par vérification dans l'image initiale (figure 8). La position de chaque forme est sauvegardée.

Ce spécialiste n'a pas été inclus dans la maquette car aucune forme pleine ne peut être dessinée on-line (sur tablette).

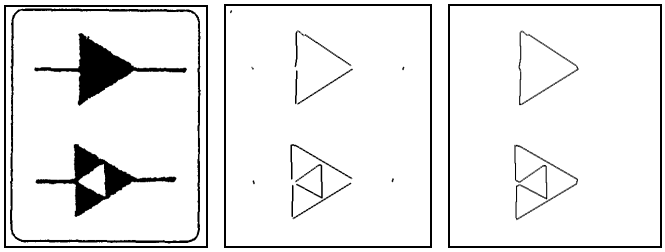


Image initiale Vecteurs Formes Pleines

Figure 8 : Détection des contours des formes pleines

- Pour reconnaître **les éléments chimiques (liaisons, polygones et chaines)**, il nous a semblé nécessaire d'organiser les informations extraites de manière à mieux utiliser le contexte lors de l'interprétation ; une représentation sous la forme d'un graphe est la plus adaptée pour lier les différents objets qui constituent le dessin en fonction de leur voisinage, c'est aussi la plus apte à traduire les relations structurelles entre primitives. C'est le travail réalisé par la fonction **«Redessine» :**

Les traits (quadrilatères) extraits constituent les noeuds du graphe ; les arcs reliant ces noeuds traduisent les relations qui existent entre les primitives ; le graphe représente la structure globale du document. Pour le construire, on calcule pour chaque primitive une zone d'influence. Des arcs décriront la nature des liens existant entre la primitive en cours d'étude et celles appartenant à sa zone d'influence. Le tableau ci-dessous (figure 9) répertorie les différents types de relations qui servent à la construction du graphe.

Type de liaison entre deux primitives	Exemple correspondant
T (Liaison orientée en T)	
X (Intersection)	
P (Parallèles)	
L (Liaison en L)	
S (Successifs)	

Figure 9 : Types d'interaction

La figure 10 décrit le processus de construction de la partie du graphe correspondant au Quadrilatère 0 (élément de SQ). La figure 11 montre un graphe structurel obtenu et représentatif d'une formule chimique.

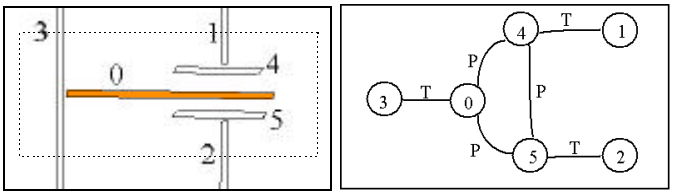


Figure 10 : Zone d'influence du Quadrilatère 0 et graphe correspondant

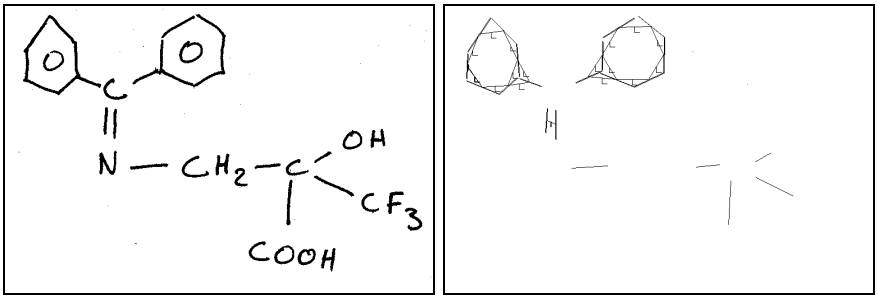


Figure 11 : Exemple de graphe obtenu

- L'étude des relations entre les traits constituant la formule chimique obtenue à l'aide du graphe construit permet à différents spécialistes : le spécialiste **Liaisons**, le spécialiste **Polygones** et le spécialiste **Chaînes** de localiser et reconnaître les différents constituants des **éléments chimiques**.

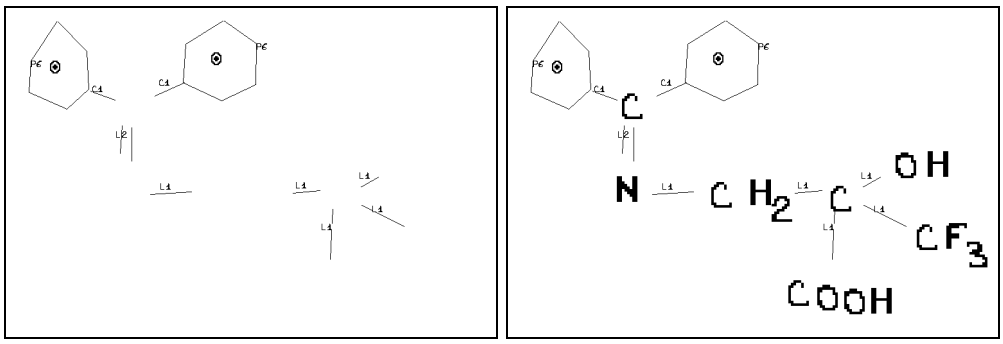


Figure 12 : Polygones (P), Liaisons (L), et Chaînes (C) détectés et **résultat final complet**

4. La Reconnaissance des Caractères Manuscrits

Auteur : Guillaume Boissier

Cette étude a été effectuée dans le cadre du développement d'un programme de reconnaissance de formules chimiques. Ce document concerne l'étude de la reconnaissance des caractères manuscrits contenus par le schéma à traiter. Grâce à un module extérieur à cette étude, tous les caractères ont été au préalable isolés. Notre objectif est alors de les "étiquetés".

4.I. Algorithme de reconnaissance des caractères isolés

A. Choix des paramètres et méthodes de comparaison

Notre choix a été orienté par les directives suivantes:

- Le module de reconnaissance de caractères devra être facilement adaptable de la reconnaissance off-line vers la reconnaissance on-line.
- Le module de reconnaissance de caractères devra s'appuyer sur la reconnaissance de caractères isolés (et non sur l'analyse de lignes, blocs, mots, ... et caractères).

Les méthodes envisagées:

1. Comparaison directe avec des modèles de références.

Cette technique est la plus basique que l'on puisse imaginer. Elle consiste tout simplement en la comparaison d'un modèle avec le caractère localisé. Aucun traitement n'est effectué sur le caractère à reconnaître ce qui lui permet de conservé toutes ces caractéristiques. Toutefois, le principal inconvénient de cette méthode tien au fait de la non-stabilité de la taille des caractères, cela nous obligerait dès lors à disposer d'une base de modèles pour toutes les tailles de caractères, ce qui nous a semblé peu réaliste.

2. Méthode préconisée par F. Le Bourgeois dans son rapport de DEA

Cette technique s'appuie sur l'analyse des concavités et connexités du contour intérieur et extérieur de chaque caractère. De plus, cette méthode implique la nécessité de distinguer profil haut, bas, gauche et droite d'un caractère, ce qui ne nous a pas semblé aisé à réaliser. Par ailleurs, nous ne pensons pas que cette technique puisse être adaptée à la reconnaissance on-line. Enfin la complexité de cette méthode nous laisse présager une certaine lenteur dans son exécution.

3. Méthode de graphes et arbres étiquetés - Utilisation des codes de Freeman

Cette technique se base sur une description du squelette de chaque caractère, par une succession de codes. Un code exprime la direction suivie pour passer d'un point remarquable à un autre au sein du squelette. Nous avons choisi d'écarter cette méthode pour les raisons suivantes:

- L'obtention du squelette d'un caractère est très sensible aux variations et perturbations.
- Pour parcourir un caractère, il est nécessaire de déterminer un point de départ et un point d'arrivée qui soient toujours identiques quel que soit le scripteur. Si cette manipulation semble évidente sur un "C", elle l'est beaucoup moins sur un "H" ou un "O".

4.Méthode d'analyse des concavités, des boucles, ainsi que de la courbure

Ces différentes méthodes servent à mettre en évidence les concavités, les courbes de chacun des caractères. Ces méthodes sont très souvent utilisées en complément parce qu'elles sont

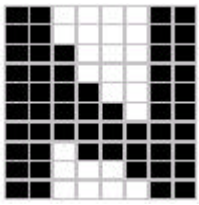
trop compliquées à mettre en place comme unique moyen de différenciation. Dans le cadre de notre projet ces méthodes ne nous ont pas paru suffisamment déterminantes pour constituer en soit le moyen de différencier les caractères manuscrits. Elles pourront toute fois être utilisées ultérieurement en complément de méthodes plus discriminatoires.

5. Méthode choisie

Notre méthode combine trois techniques. Tout d'abord elle s'appuie sur une méthode à base de masques. Pour cela nous commençons par normaliser le caractère à reconnaître, puis nous lui superposons un "masque de référence". Par calcul de la distance entre le caractère et les masques de la base de référence, nous arrivons à déterminer quels sont les deux masques les plus proches de la forme à reconnaître. Ensuite nous nous appuyons sur l'étude des directions principales des quadrilatères (traits) composant le caractère. Finalement nous corroborons ceci avec l'étude des demis profils droits du caractère normalisé. Toutes ces méthodes sont expliquées en détail dans la suite de ce document.

B. Normalisation

Cette technique sert à ramener le caractère à une taille inférieure définie au préalable, tout en conservant au maximum les caractéristiques.
Pour cela, nous avons choisi de "découper" le caractère en un damier, puis au sein de chaque case, compter le nombre de pixels noirs (le damier, empiriquement choisi, est de 8 cases en largeur, 10 cases en hauteur). Si le taux de noir est supérieur à un seuil (ici 25%) alors la case correspondante dans le caractère normalisé est considérée comme noire, sinon elle est considéré comme blanche.
Lors de la découpe, si le nombre de lignes (respectivement de colonnes) n'est pas un multiple de notre division, nous avons pris le parti de compter autant de lignes (respectivement de colonnes) en double qu'il est nécessaire pour obtenir le multiple supérieur. Les lignes (respectivement les colonnes) comptées deux fois correspondent à celles situées en bas (respectivement à gauche) d'une case. Elles apparaissent alors en haut (respectivement à droite) dans la case suivante.
Le seuil, déterminant si la case normalisée doit être blanche ou noire, a été choisi empiriquement suite à une étude des résultats obtenus.



Caractère N normalisé en 8*10.

C. Technique de comparaison (calcul de distance)

1. Distance principale

Pour bien comprendre le calcul de distance, il est nécessaire de bien comprendre comment fonctionne notre base de modèles.

Chaque modèle de référence est composé d'un nom (représentant le caractère), ainsi que d'un tableau de "cases". Chaque case peu se trouver dans les états suivants:

- Noir (ou 1): le caractère comparé doit comporter une case noire au même endroit.
- Blanc (ou 0): le caractère comparé doit comporter une case blanche au même endroit.
- Bleu (ou 2): Le caractère comparé peut comporter une case dont la couleur n'a pas d'importance.

Modèle de référence du caractère m.

Ainsi il existe un certain nombre de cases auxquelles nous n'accordons pas d'importance. Toute fois il est important de noter que leur nombre est limité pour chaque modèle (ici le maximum a été fixé à 8 cases sur un total de 80 soit 10%). Cet état se justifie lors de l'augmentation du nombre de modèles de référence dans la base. En effet deux modèles d'un même caractère suffisamment proche peuvent être fusionnés en un seul, leurs différences seront alors qualifiées de cases non significatives.

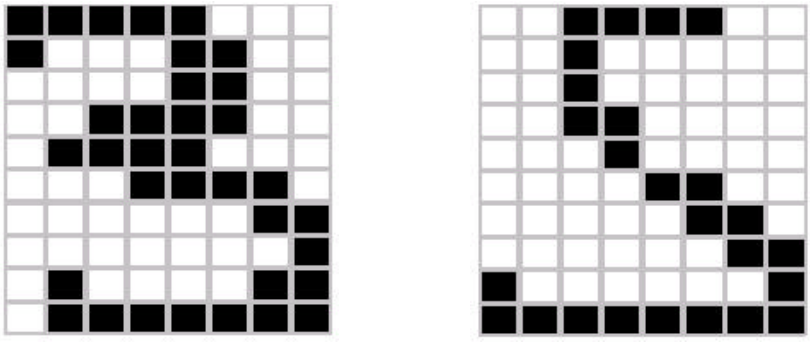
Le calcul de la distance entre deux modèles (ou entre un modèle et un caractère normalisé) se fait alors très simplement. Pour chaque case, on regarde sa valeur dans les deux damiers. Si elles sont différentes et qu'aucune d'elles ne correspond au code d'une case indifférente (ou bleu) alors on ajoute 1 à la distance. Le résultat ainsi obtenu est donc le nombre de cases "significatives" différentes entre deux damiers. Après chaque ajout d'un nouveau modèle de référence, nous appliquons notre algorithme de fusion à la base des modèles. Le mécanisme de fusion est expliqué plus loin dans ce document.

Pour reconnaître un caractère nous normalisons celui-ci, puis nous cherchons parmi notre base de modèles, quels sont les plus proches, de celui-ci. Si la distance entre le caractère et le modèle le plus proche est inférieure à un seuil (ici nous avons choisi 17, ce qui correspond a une différence maximal de 17/80 soit environ 21%) nous pensons avoir reconnu le caractère, sinon nous déclarons que ce caractère est inconnu mais qu'il est proche de ... à une distance de ... .
Conserver les deux modèles les plus proches nous permet de pouvoir ultérieurement changer d'avis sur le caractère reconnu.

2. Autres paramètres / critères de comparaison

a)Etude des demis-profils droits

Afin d'augmenter la distance entre certaines lettres dont les dessins sont très proches (F et P, C et O, m et n, S et 3, etc. ...) nous utilisons une deuxième méthode. Elle met plus particulièrement en évidence les différences présentes sur la moitié droite des caractères. Cette méthode se base sur le caractère normalisé ainsi que sur les 2 modèles de référence les plus proches. Nous comptons pour chacun le nombre d'intersections entre une droite horizontale et le demi dessin normalisé droit d'un modèle (un F sera du genre 1,0,0,0,1,0,0,0,0,0 alors qu'un P sera 1,1,1,1,1,0,0,0,0,0 ; etc. ...). Comme nous travaillons sur les caractères normalisés, nous obtenons deux listes de nombres de même taille (même nombre d'éléments). Pour calculer la distance entre deux modèles, il suffit de faire la somme des valeurs absolues des différences entre chaque nombre de même rang. Lorsqu'il y a hésitation entre deux modèles, le système choisi celui qui est le plus proche en terme de nombre d'intersections.



Modèles de 3 et de S.

Le calcul des demis-profils droits donne sur ces deux exemples:

- pour le 3: 1;1;1;1;1;1;1;1;1;1.
- pour le S: 1;0;0;0;0;1;1;1;1;1.

Si ce même calcul avait été effectué sur le caractère entier nous aurions obtenu:

- pour le 3: 1;2;1;1,1,1,1,1,2,1.
- pour le S: 1;1;1;1;1;1;1;1;2;1.

Le premier calcul permet donc de mieux mettre en évidence les différences des deux caractères.

b) Etude des quadrilatères

De façon à augmenter encore la discrimination entre deux caractères nous combinons les méthodes décrites précédemment à l'étude des quadrilatères (traits) "significatifs" qui composent un caractère. Pour cela nous étudions les directions de ces quadrilatères, et nous comparons avec les règles définies dans une base de règles. Ceci nous sert plus à différencier deux caractères, qu'à réellement reconnaître le caractère.

Nous appelons quadrilatères "significatifs" ceux qui sont inclus dans la zone texte étudiée, et dont la longueur est supérieur ou égale à 1/3 (seuil fixé empiriquement) de la longueur ou de la hauteur (suivant son inclinaison) de la zone texte. Nous avons défini un "dictionnaire de règles" qui, pour les caractères ou cela est possible, détermine le nombre de quadrilatères devant avoir une orientation parmi les 4 suivantes: horizontale, verticale, diagonale droit, diagonale gauche. Les orientations des quadrilatères "significatifs" ont été déterminés à partir de l'angle moyen (compris entre 0 et 180°) qu'il forme avec l'horizontale. Un quadrilatère formant un angle appartenant à:

- [0;27[U] 153;180 [a une direction horizontale (-).

- [27;63] a une direction diagonale droit (/).
-]63;117[a une direction verticale (-).
- [117;153] a une direction diagonale gauche (\).

Cette méthode agit de la façon suivante:

Si le caractère reconnu dispose de règles dans le dictionnaire et qu'il ne les vérifie pas, alors nous regardons si le caractère le plus proche a des règles vérifiées. Si oui, nous permutons caractère reconnu et caractère proche. Dans tous les autres cas aucun changement n'est effectué.

A gauche caractères N et H, à droite quadrilatères "significatifs" correspondants.

c) Résumé du processus de reconnaissance

Notre procédé est donc le suivant:
- Nous appliquons la méthode des masques (ou modèles).
- Si la différence entre les deux distances des modèles retenus est inférieure à un seuil (ici 6) ou si le caractère n'a pas été reconnu, alors nous faisons appel à la méthodes des "demis profils droits" et enfin à la méthode des quadrilatères. Cet appel à la méthode des quadrilatères nous permet de vérifier la validité des changements effectués par l'étude des demis profils droits.

D. Améliorations possibles

Les principaux inconvénients de notre méthode sont les suivants:

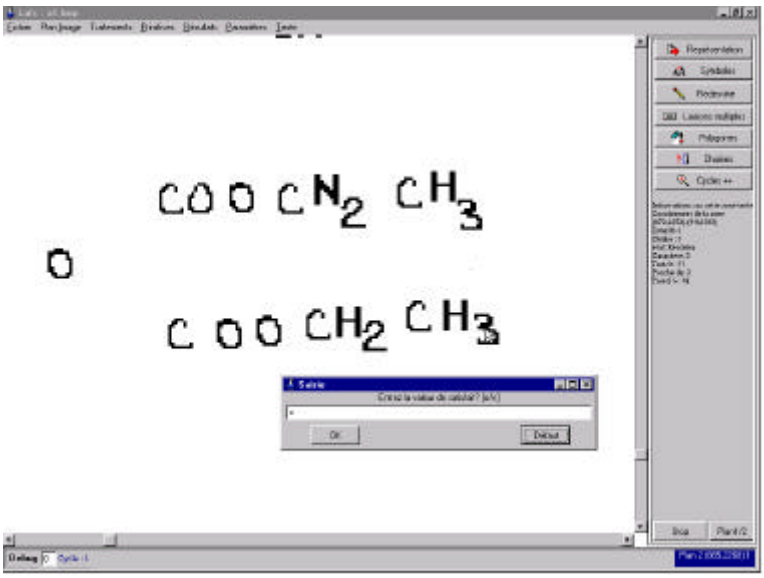
- Si un scripteur attache deux caractères, ils sont alors reconnus comme un seul caractère. Malgré une de nos méthodes pour tenter de séparer les zones ou deux caractères peuvent se trouver collés, ce problème persiste (peu de résultats acceptables) et nous semble très difficile à traiter.

- Si une des lettres est anormalement terminée (boucle d'un trois, d'un s ... qui se prolonge) le caractère n'est pas reconnu..
- Si deux zones de caractères se chevauchent, il ne nous est pas possible de différencier les pixels qui appartiennent à l'un ou à l'autre dans la partie commune.
- La zone délimitant un caractère étant un rectangle ceci ne permet pas de reconnaître certain type d'écriture "trop" penché tel l'italique.

Nous n'avons à ce jour développé aucune méthode pour palier à ces problèmes.

E. Interface utilisateur / Correction

Afin de laisser à l'utilisateur la possibilité de corriger facilement les erreurs de reconnaissance commises par l'ordinateur, nous avons élaboré une méthode qui permet, par simple click sur une zone texte, de pouvoir en visualiser les données et le cas échéant les modifier. Grâce a cette méthode l'utilisateur peut ajouter des modèles à la base.

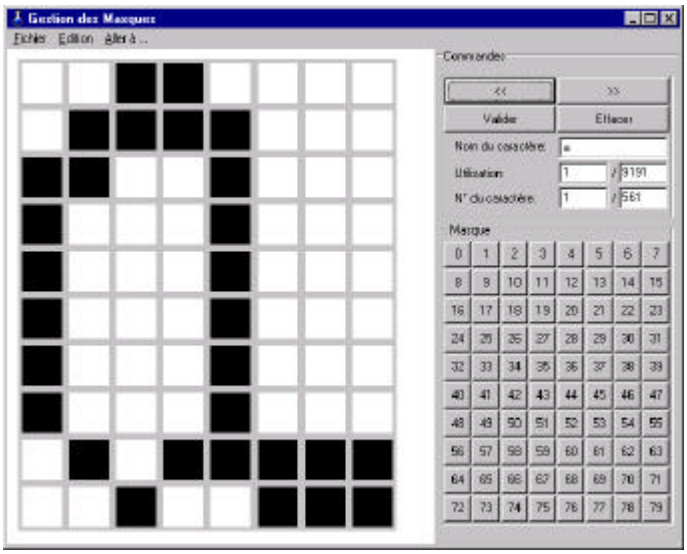


Interface de correction.

II. Apprentissage.

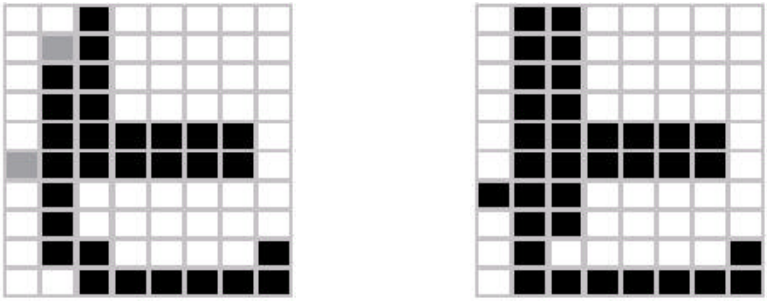
A. Construction / Evolution et Gestion des modèles

Afin de gérer et de construire notre base de modèles nous avons développé plusieurs outils. Tout d'abords nous avons créé une interface graphique, qui nous permet de visualiser les modèles, mais aussi de les modifier, supprimer ou encore d'en ajouter de nouveaux.

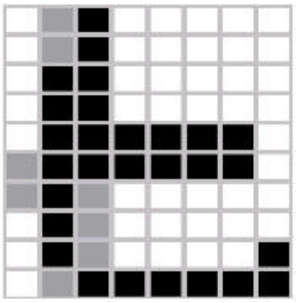


Interface de gestion de la base des modèles.

Afin de créer des modèles, nous avons également développé un algorithme de reconnaissance qui demande à l'utilisateur s'il souhaite ajouter un caractère non reconnu comme modèle. De même, notre interface de correction autorise l'ajout de modèles. Toutefois, pour un non expert, il est très difficile de choisir s'il faut ou non ajouter un modèle. En effet de ce choix découle le bon fonctionnent de la reconnaissance. Il faut donc que l'utilisateur se demande si le caractère qu'il veux ajouter à la base correspond bien a un caractère souvent utilisé et dont le dessin a été effectué proprement. L'ajout de lettres mal formées, ayant un dessin proche d'autres lettres, pourrait avoir des conséquences néfastes.



Deux modèles de t pouvant fusionner.



Résultat de la fusion.

Dans le but de limiter le nombre de modèles, nous avons mis au point un algorithme qui fusionne tous les modèles d'un même caractère suffisamment proche. Notre algorithme fonctionne de la façon suivante. Pour chaque modèle de référence contenu dans notre base, nous cherchons le modèle suivant codant le même caractère. Nous cherchons alors à les fusionner: si leur différence n'est pas supérieure au nombre de cases indifférentes (ou bleu) autorisées alors nous modifions le premier modèle de façon à ce que toutes les différences entre les deux caractères soit codées par une case indifférente (ou bleu) , le second modèle est alors supprimé. Nous cherchons ainsi à fusionner un maximum de modèles entre eux afin de limiter la prolifération de modèles pour un même caractère. Ceci permet également de minimiser l'effet de faibles différences entre deux occurrences d'un même caractère.
Il est important de noter que notre base n'est ni triée ni organisée. Le dernier modèle ajouté se trouve en fin de fichier.
Après chaque changement, la base est sauvegardée sur disque dans le fichier "bmdl.dat".

B. Adaptation aux scripteurs

Pour que la base puisse s'adapter au scripteur nous avons laissé à l'utilisateur la possibilité d'ajouter les modèles des lettres qui n'ont pas été reconnues (cf. I. E. Interface utilisateur / Correction). Ainsi par ajouts successifs de modèles, la base s'adapte petit à petit, aux nouveaux types de caractères. Afin d'éviter un accroissement constant de la base nous l'avons dotée d'une fonction "d'oubli". Ceci nous permet de nous débarrasser des modèles qui ne sont jamais utilisés. En effet notre procédé gère un nombre d'utilisation de chacun des modèles. Lorsqu'un modèle permet l'identification d'un caractère son utilisation est incrémentée. Nous gérons également le nombre total d'utilisation de la base (égal au nombre de zones caractères tentées d'identifier). L'algorithme d'oubli est très simple. Tous les modèles dont le nombre d'utilisation est égal à zéro sont effacés, et tous les compteurs sont

remis à zéro. L'utilisateur prendra donc soin de n'effectuer cette opération que lorsque le nombre total d'utilisation est suffisamment important (2000 environs...).

III. Post Traitement

A. Reconstruction des chaînes de caractères

Notre algorithme de reconstruction des chaînes de caractères est simple, il provient de l'observation de la disposition spatiale des caractères les uns par rapport aux autres. Notre méthode fonctionne sur le principe suivant. Nous recherchons le premier caractère non répertorié dans une zone (c'est à dire dont le numéro de zone est nul), puis nous cherchons tous les caractères se trouvant à sa gauche à sa droite, dans son coin supérieur gauche, supérieur droit, inférieur gauche, inférieur droit. Nous limitons, évidement, cette recherche à une distance établie à l'aide du calcul de la taille moyenne des zones caractère. Chaque zone ainsi trouvée se voit attribuer le même numéro de bloc. Pour chaque voisin trouvé, nous réitérons la procédure de recherche des voisins. Ainsi nous regroupons par bloc les différentes zones testées. Au sein de chaque bloc nous trions par ordre d'abscisses croissantes les différentes zones textes.

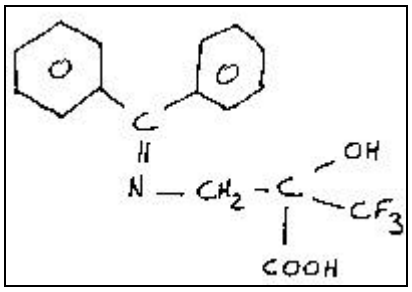
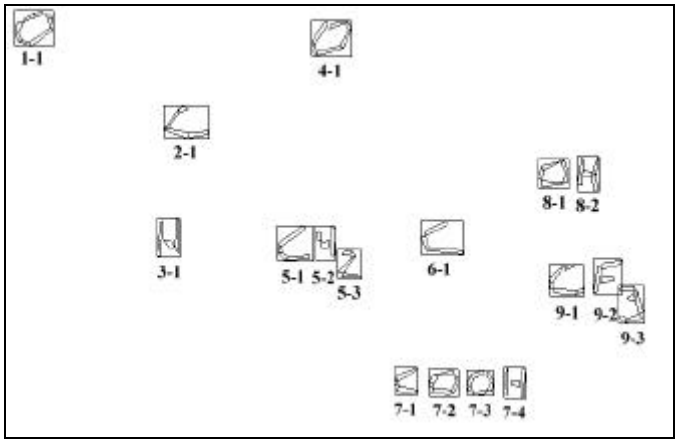


Image à traiter.



Découpage en bloc texte.

B. Utilisation du Contexte

Une fois les caractères regroupés en blocs, nous pouvons analyser le contenu de ces zones, et notamment vérifier l'existence des symboles formés. Pour cela nous cherchons dans chaque bloc, à former des mots de longueur maximum. Un mot est un ensemble de lettres pouvant correspondre à un symbole de notre dictionnaire. La fin d'un mot est trouvée lorsque:

- le caractère suivant n'est pas une minuscule (donc un chiffre, une zone complexe, une majuscule ou un signe)
- la fin du bloc a été atteint.
- le mot composé comprend plus de 4 caractères (en effet le symbole le plus long de notre dictionnaire est Naph soit 4 caractères au maximum).

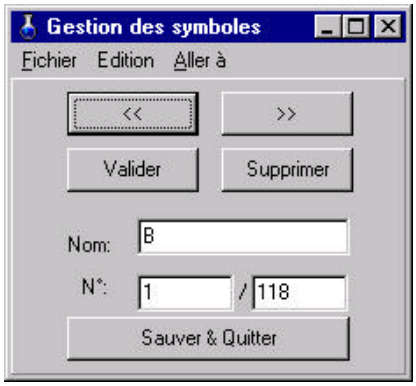
Attention un mot commence forcément par une majuscule (ou un t minuscule pour autoriser tBu tMe ...).
Ensuite nous testons la validité du mot trouvé. S'il appartient au dictionnaire alors il est valide et on passe au suivant, sinon on essaye toutes les permutations possibles entre caractère retenu et caractère proche de celui retenu. Si aucune permutation ne donne de résultats probants, alors on essaye de vérifier si le mot existe avec une lettre en moins (la dernière). l'opération est ainsi réitérée jusqu'à arriver à un mot de longueur 1 caractère. Si aucun caractère de cette zone texte n'appartient au dictionnaire, alors la zone est marquée comme problématique, et vraisemblablement erronée.

C. Dictionnaire (Gestion - Utilisation)

Le dictionnaire est divisé en deux. Un premier comporte l'ensemble des symboles composés d'un seul caractère. Le second contient tous les autres symboles existants. Ils se présentent sous la forme de deux fichiers textes "éditables". Leur structure est la suivante:

Nombre de symboles
Symbole numéro 1
...

Pour facilité l'utilisation de notre logiciel nous avons développé une interface qui permet a l'utilisateur d'éditer la liste des symboles contenus dans les deux fichiers dictionnaires. Lors de la sauvegarde des modifications effectuées, le tri par ordre alphabétique ainsi que la répartition dans les deux fichiers sont effectués automatiquement.



Interface de gestion du dictionnaire de symboles chimiques

D. Apport (sur le taux de reconnaissance)

Ici réside un point clé de notre programme. Plus le dictionnaire est petit, plus la reconnaissance sera fiable. La suppression des lettres peu utilisées comme G D Y et de certains symboles tels Dy Sm Uuu ... améliorerait grandement les résultats.

IV. Segmentation

A. Caractères collés

Ce type de problème n'est que partiellement résolu par notre programme. En effet nous avons pris le parti d'essayer de découper un nombre très réduit de zones à savoir celles pour lesquelles l'algorithme de reconnaissance (normalisation plus calcul de distance) n'a rien donné. Ainsi nous ne travaillons que sur des zones déjà qualifiées de peu sûr. La principale difficulté est de localiser correctement les débuts réels des caractères ainsi que leur fin. Cette difficulté est de plus accentuée par le fait que nous utilisons des zones rectangulaires (la césure entre deux lettres liées n'étant pas forcement verticale, mais plutôt à tendance diagonale (d'un côté pour les droitiers de l'autre pour les gauches)).

Le principe de séparation que nous avons choisi d'adopter est le suivant:
Le début de la zone constitue le début du premier caractère. Les caractères souvent liés aux autres étant essentiellement C et O, nous avons centré notre travail sur ces derniers. Nous considérons donc qu'un premier caractère d'une zone double n'est débuté que par des lettres telles que le nombre d'intersections avec des droites verticales se limite dans un premier temps à 1. Nous parcourons donc la zone suivant les x croissants, de la façon suivante. Tant que le nombre d'intersections avec des droites verticales est 1 , nous comptons le nombre de pixels de couleur noire faisant intersection avec chaque droite. Ceci nous permettra d'obtenir ultérieurement l'épaisseur moyenne (en nombre de pixels) d'un trait. Nous comptons également le nombre de fois qu'il nous est possible d'obtenir une seule et unique intersection. Ceci nous permet de déterminer une largeur maximum à laisser en fin de caractère. Ensuite nous continuons de balayer cette zone tant que nous ne retrouvons pas un nombre d'intersection de nouveau égal à 1 (ce qui signifie alors que nous entamons la fin d'un caractère ou le début du trait qui lit deux caractères entre eux). Nous progressons alors selon x tant que le nombre d'intersections avec des droites verticales est égal a 1 et que le nombre de pas effectués n'est pas supérieur à la largeur déterminée précédemment. Nous supposons dès lors que nous avons atteint la fin du premier caractère. Nous recherchons ensuite le début du caractère suivant. Pour cela nous nous basons sur l'étude de l'épaisseur moyenne d'un trait effectuée précédemment. Tant que nous n'avons pas obtenu une épaisseur de trait 1.5 fois (cette valeur est arbitraire et obtenue après étude des résultats) supérieure à l'épaisseur normale d'un trait nous décrétons qu'il ne s'agit que de la liaison entre deux caractère. Ainsi nous pouvons déterminer le début du deuxième caractère. La fin de celui ci correspond à la fin de la zone texte. Reste alors à resserrer la zone contenant un caractère selon y. Nous appliquons ensuite à chacune des deux zones notre algorithme de comparaison avec les

modèles de référence. Si cette comparaison donne des résultats probants (différents de non-reconnu) alors nous validons le découpage, sinon la zone reste “non-reconnu” et peu fiable.

Image Initiale.

Localisation des zones textes.

Tentative de dédoublement des zones textes non reconnues.

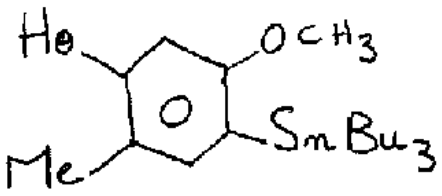
Résultat final de la reconnaissance.

V. Test / Evaluation

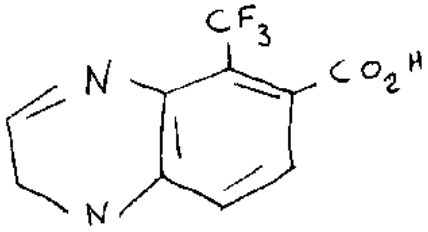
A. Exemples - Taux de reconnaissance

Pour évaluer correctement le taux de reconnaissance nous avons utilisé une base de modèles nouvellement créée, comportant 328 masques, ainsi que 4 documents n’ayant pas servi à la création de la base. Bien évidement une base plus fournie présenterait de meilleurs résultats. Le taux de reconnaissance est fonction de la base des modèles, de l’application du scripteur, de la taille du document, du nombre de symboles autorisés dans notre dictionnaire.

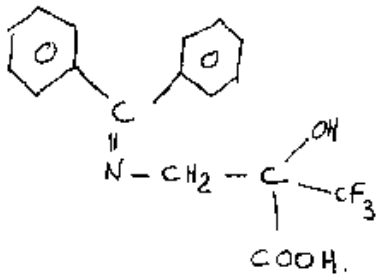
Document 1 :



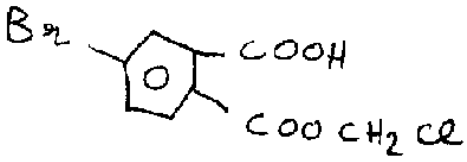
Document 2 :



Document 3 :



Document 4 :



1. Reconnaissance pure

Document	% Reconnaissance	% indécision	% Inconnu	% Erreur
1	64,29%	28,57%	7,14%	0,00%
2	22,22%	55,56%	22,22%	11,11%
3	58,82%	35,29%	5,88%	0,00%
4	64,29%	28,57%	28,57%	0,00%

2. Après découpage et application des autres méthodes

Document	% Reconnaissance	% indécision	% Inconnu	% Erreur
1	78,57%	21,43%	0,00%	0,00%
2	22,22%	66,67%	0,00%	11,11%
3	58,82%	41,18%	0,00%	0,00%
4	60,00%	40,00%	0,00%	0,00%

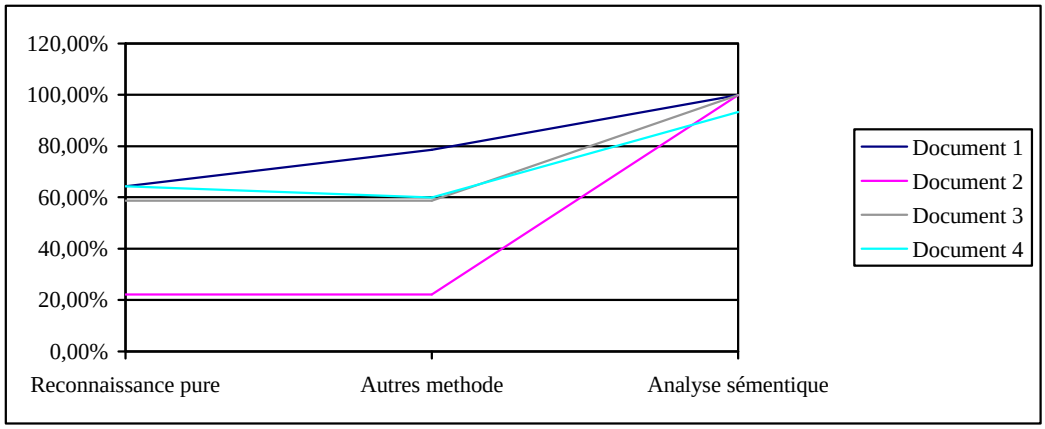
Notons que le 4^{ième} document comporte une zone qui a été dédoublée (Cl).

3. Après analyse sémantique (fin du processus)

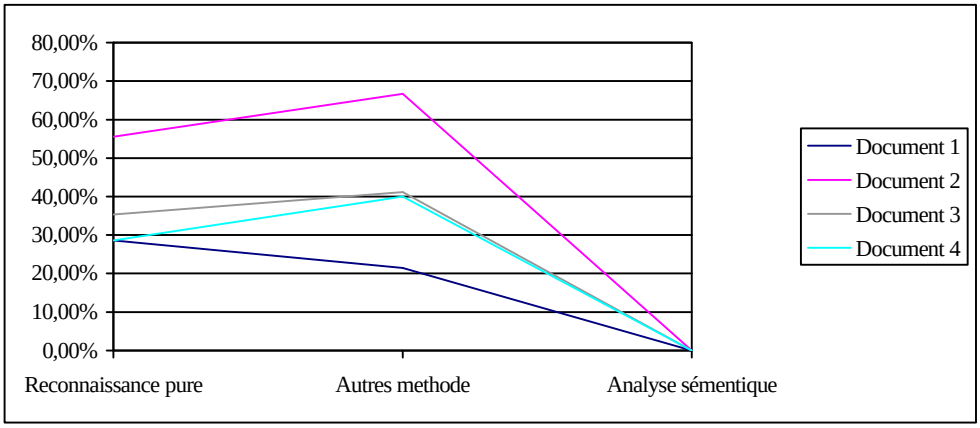
Document	% Reconnaissance	% indécision	% Inconnu	% Erreur
1	100,00%	0,00%	0,00%	0,00%
2	100,00%	0,00%	0,00%	0,00%
3	100,00%	0,00%	0,00%	0,00%
4	93,33%	0,00%	0,00%	6,67%

La seule erreur qui persiste est dû au mauvais découpage d'une zone dans le 4^{ième} document.
Le Cl a été reconnu comme C2.

4. Evolution du taux de reconnaissance au sein du processus



5. Evolution du taux d'indécision au sein du processus



B. Evolution de la base des modèles - Nombres d'images utilisées pour apprendre

Nous avons tout d'abord longuement hésité entre deux méthode :

- Soit nous utilisions une base unique pour tous les scripteurs
- Soit nous utilisions une base pour chaque scripteur

Nous avons préféré la deuxième méthode qui permet une meilleure reconnaissance à long terme, même si à cours terme la première idée semblait mieux fonctionner. En effet chacun a sa façon de former les caractères, même si beaucoup de caractéristiques se retrouvent d'une écriture à l'autre.
Un nouveau scripteur pourrait bien entendu débuter avec une base de modèles entraînée sur une autre écriture que la sienne, mais nous pensons que le temps mis pour se débarrasser des modèles qui diffèrent de son écriture est bien plus important que le temps d'un nouvel apprentissage.

Pour évaluer le nombre de documents moyens à produire pour effectuer un apprentissage correct, nous avons décidé d'effectuer nous mêmes cette expérience. Les documents utilisés pour ce procédé sont tous identiques. Ils se composent de la classification périodique des éléments (document comportant l'ensemble des symboles autorisés par notre dictionnaire)

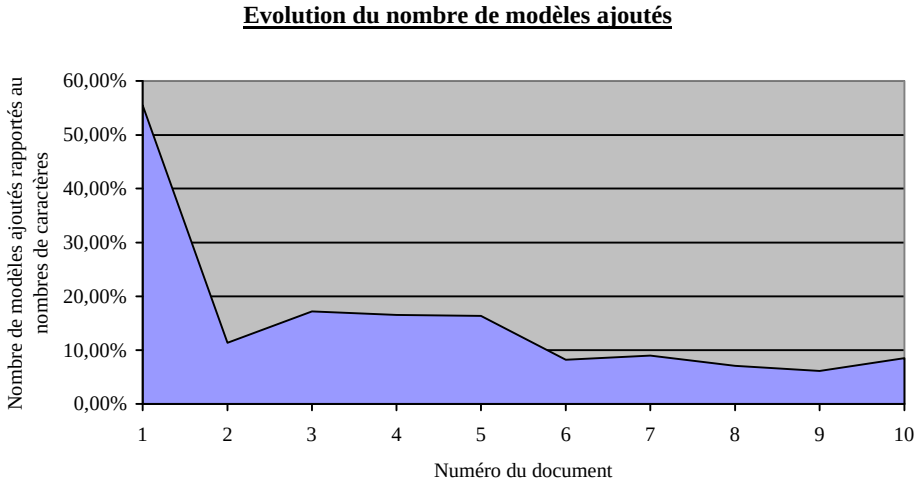
ainsi que de la liste des symboles supplémentaires (tels que tBu, tMe, Naph, Phi, Ph, Et, φ, ...) et la liste des chiffres (0 à 9) recopié par le nouveau scripteur.

Le premier document a été traité en utilisant une méthode qui permet de demander à l'utilisateur de confirmer la reconnaissance effectuée par l'ordinateur (cette méthode n'utilise pas le post traitement). Tous les documents suivants ont été traités en appliquant l'algorithme classique de reconnaissance (post traitement inclus), puis une correction, au moyen de l'interface développée à cet effet, a été effectuée par l'utilisateur.

Les résultats sont les suivants:

Document n°	Nombre de caractères détectés	Nombre de modèles ajoutés	rapport ajouts / nombre de caractères (en %)	nombre de caractère dans la base en fin de traitement
1	227	126	55,51%	126
2	220	25	11,36%	151
3	221	38	17,19%	189
4	218	36	16,51%	225
5	220	36	16,36%	261
6	220	18	8,18%	279
7	223	20	8,97%	299
8	226	16	7,08%	315
9	229	14	6,11%	329
10	223	19	8,52%	348

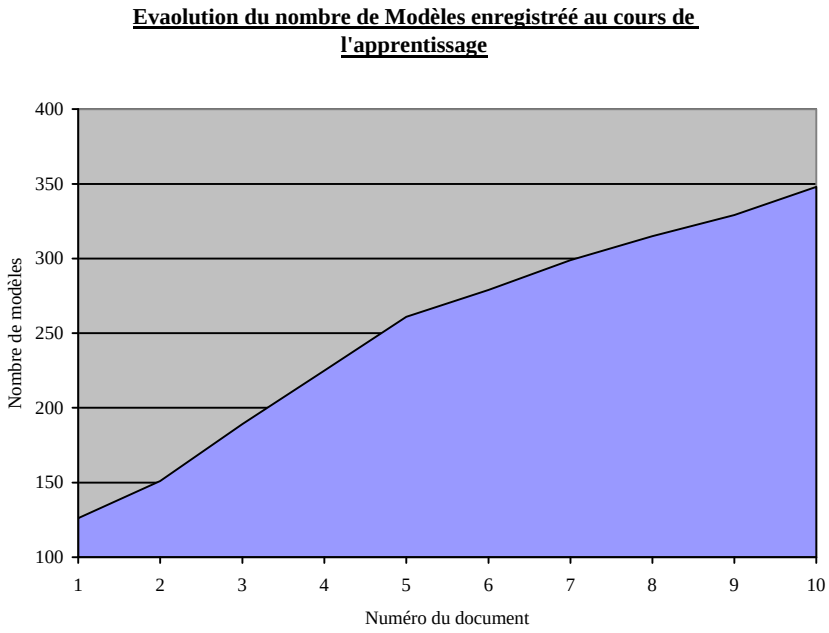
Le graphique suivant montre bien l'évolution de l'apprentissage :



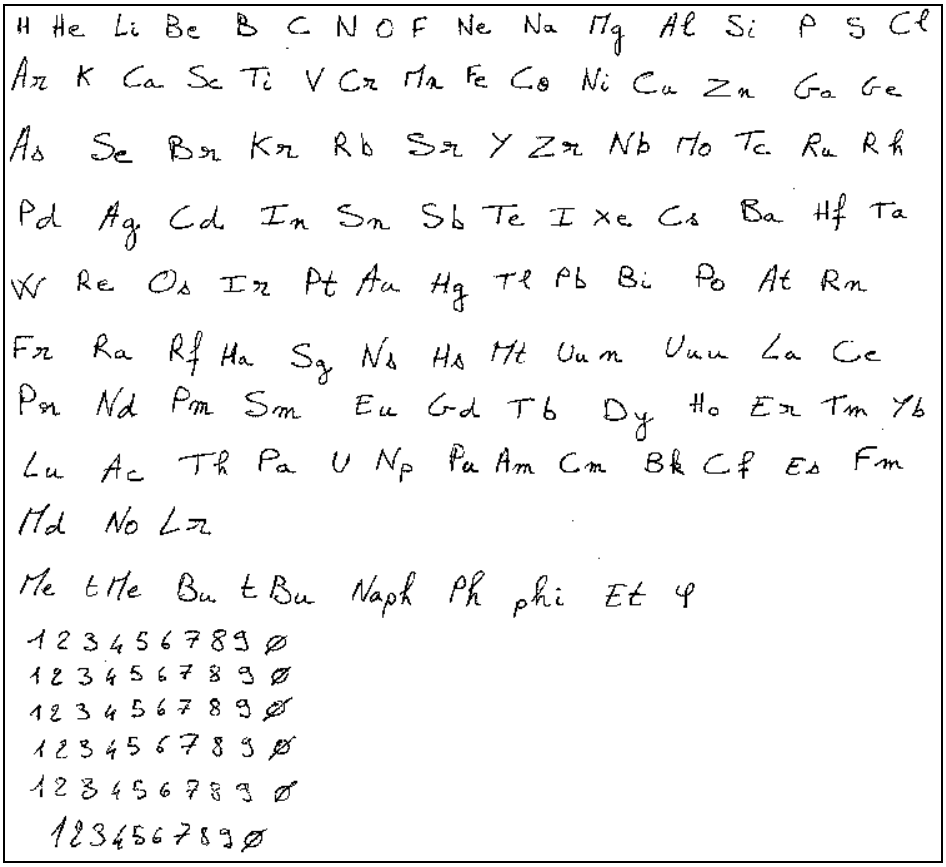
On peut distinguer 3 phases :

- Le premier temps concerne uniquement le premier document. Pour la première fois des modèles sont ajoutés à la base. Ceci est le tout début de l'apprentissage.
- La deuxième phase concerne les documents 2 à 5. Pendant cette période différents caractères ont été ajoutés, ce sont aussi bien les plus usités (O, N, H, ...) que les plus folkloriques (Y, D, U, ...) qui sont alors appris.
- La dernière partie concerne les documents 6 à 10. Ici les caractères les plus usités ont été enregistrés et seul des caractères peu utilisés sont rajoutés à notre base de modèles.

L'évolution de la taille de la base est représentée sur le graphique suivant :



Nous constatons donc la croissance de la base des modèles au cours de l'apprentissage. Toute fois nous pouvons remarquer que cette croissance se ralentit à partir du 5ème document.
Ainsi donc, nous recommandons pour un apprentissage correct un minimum de 5 documents (6 seraient préférable). Bien évidement cet apprentissage ne suffit pas pour obtenir un taux de reconnaissance très élevé. Beaucoup de pratique, non plus sur la classification périodique, mais aux contraire sur des exemples concrets permet de biens meilleurs résultats. Toutefois cet exercice donne une base non négligeable de modèles de référence.
Le document suivant présente un exemple du type de fichier utilisé lors de l'apprentissage.



VI. Résumé de la méthode utilisée pour la reconnaissance des caractères

Les grandes étapes de notre méthode sont les suivantes:

Pour chaque zone texte

- Normalisation
- Comparaison avec les modèles de références et sélection des deux modèles les plus proches
- Si la zone n'a pas été identifiée, alors tentative de séparation de la zone en 2.
- Vérification du choix effectué au moyen de l'étude des "demis-profils".
- Vérification du choix effectué au moyen de l'étude des quadrilatères.

Analyse de l'ensemble des zone textes.

- Si une zone est placée en position d'indice (coin bas droit de la zone précédente) et s'il existe un doute sur le contenu de cette zone mettant en jeu un chiffre, alors nous donnons priorité à ce chiffre.
- Découpage en mots et analyse sémantique grâce au dictionnaire.

5. Conclusion

L'étude de l'existant a montré que très peu de travaux dans ce domaine avaient abouti à un résultat commercialisable. Aucun logiciel de ce type n'a été recensé sur le marché actuel.

La maquette réalisée permet :

- la lecture de fichiers BMP binaires : elle travaille donc sur des documents off-line et non on-line (tablette). La résolution choisie est de 300 dpi.
- le passage de l'image (un amas de pixels noirs et blancs) à une représentation structurée de l'image (vecteurs et quadrilatères).
- la localisation des zones de Texte dans l'image
- la reconnaissance du texte manuscrit localisé
- la construction d'un graphe représentatif pour la partie non textuelle (les traits)
- la localisation des liaisons multiples
- la localisation des polygones
- la localisation des chaînes
- la sauvegarde des résultats dans un fichier texte.

Les points les plus sensibles semblent être :

- la localisation des zones de Texte dans l'image : un seuil défini la taille maximale autorisée pour une zone textuelle. Si la formule comporte des caractères de taille trop élevée ou connectés à une autre partie du schéma ceux-ci ne sont pas localisés.

- la reconnaissance du texte manuscrit : comme prévu, le taux de reconnaissance des caractères manuscrits n'est pas impressionnant. Il a donc été nécessaire de définir certaines hypothèses afin d'améliorer les performances à ce niveau :

    - ◊ chaque scripteur potentiel devra posséder une base de modèles personnelle chargée au « login »

◊ les caractères collés les uns aux autres ou aux graphiques sont à éviter

◊ l'écriture devra être « lisible »

- la localisation des polygones : celle-ci est correcte lorsque les polygones sont fermés ou quasi fermés. Lorsque les espaces entre sommets sont trop importants, le polygone est localisé sous forme de plusieurs chaînes. Là aussi, un seuil définissant la distance maximum entre sommets a été mise en place.

Ce qu'il reste à faire :

- certaines étapes mises en place rapidement lors de ce « maquettage » peuvent être améliorées : localisation du texte, ...

- les formes reconnues sont, pour l'instant, des formes géométriques constituées de segments de droites (liaisons, polygones, chaînes).

Pour poursuivre, il est nécessaire d'étudier, d'associer ces différentes formes géométriques (entre elles et avec le texte reconnu) afin de reconstruire et reconnaître la formule chimique dessinée. Durant cette étape, il est nécessaire d'utiliser des techniques d'intelligence artificielle (des règles sémantiques sur la constitution des formules chimiques) pour obtenir une formule chimique cohérente et si nécessaire corriger automatiquement les erreurs dues aux étapes antérieures. Cette étape nécessite donc une collaboration étroite entre chimistes et informaticiens.

- Passage du off-line au on-line. Les seuils et techniques utilisés dans la maquette sont adaptés à un type d'images particulier : images binaires numérisées à 300 dpi. Il faudra en tenir compte pour réutiliser ce qui à été fait avec une tablette graphique.

- Elaboration de l'interface utilisateur finale.

Pour conclure et à notre avis, le résultat de cette étude est encourageant et démontre la faisabilité d'un tel système de lecture automatique de formules chimiques.

La seule difficulté nous parait être la reconnaissance du texte manuscrit. Le passage au on-line peut permettre d'obtenir des résultats bien meilleurs puisque une information supplémentaire devient accessible : le temps. Des techniques de reconnaissance plus évoluées et mieux maîtrisées sont alors utilisables.

Enfin, il est important de noter que les scripteurs (dessinateurs) devront faire un effort d'application pour qu'un système ait une chance d'interpréter le dessin réalisé. Il s'agit ici d'un point essentiel à étudier avec les futurs utilisateurs potentiels ! Une formule telle que celle présentée figure 5.1 n'a aucune chance d'être reconnue automatiquement (même en on-line) ; une formule telle que celle présentée figure 5.2 a toutes les chances d'être reconnue correctement (même en off-line).

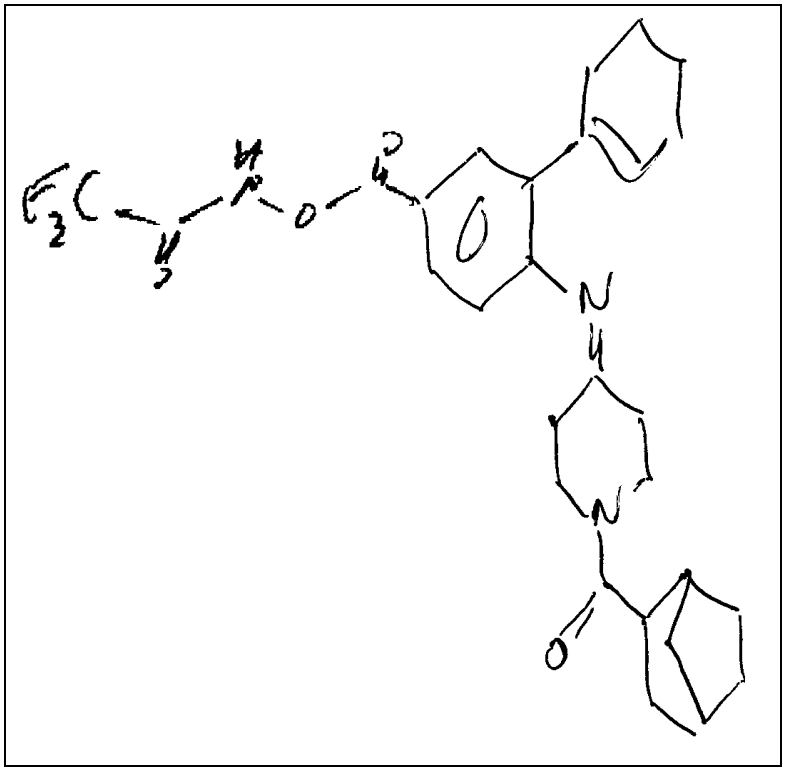


Figure 5.1 : Exemple d'image

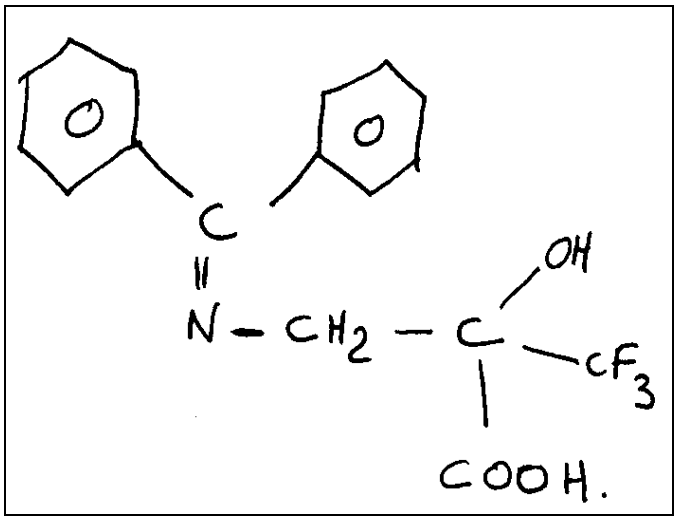


Figure 5.2 : Exemple d'image

Références Bibliographiques

[Abe86] **Abe, K., Azumatani, Y., Mukouda, M. And Suzuky, S.** Discrimination of symbols, lines, and characters in flowchart recognition. In : *Proceedings of the 8th International Conference on Pattern Recognition, Paris (France), october 27-31, 1986.* Vol. 2, p. 1071-1074.

[Antoine92] **Antoine, D.** Techniques spécialisées et représentation de la connaissance pour l'interprétation des plans cadastraux. *Bigre*, 1992, N° 80, p. 65-73.

[Casey93] **Casey, R., & Al.** Optical recognition of chemical graphics *Proceedings of the 2rd International Conference on Document Analysis and Recognition, 1993*. p. 628-631.

[Desseilligny95] **Pierrot-Desseilligny, M., Le Men, H. and Stamon, G.** Characters string recognition on maps, a method for high level reconstruction. In : *Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal (Canada), august 14-16, 1995*. Vol. 1, p. 249-252.

[Galindo97] **Galindo, D., Faure, C.** Perceptually based representation of network diagrams *Proceedings of the International Conference on Document Analysis and Recognition, Ulm, 1997*. Vol. 2. p. 352-356.

[Habacha93b] **Habacha-Hamada, A.** *Reconnaissance de symboles techniques et analyse contextuelle de schémas.* Thèse de doctorat : Institut National Polytechnique de Lorraine : Nancy (France), 1993. 169 p.

[Joseph91] **Joseph, S.H.** On the extraction of text connected to linework in document images. In : *Proceedings of the First International Conference on Document Analysis and Recognition, Saint-Malo (France), september 30-october 2,* 1991. p. 993-999.

[Kasturi90] **Kasturi, R.** A system for interpretation of line drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 1990, Vol. 12, N°10, p. 978-991.

[Ogier94b] **Ogier, J.M.** *Contribution à l'analyse automatique de documents cartographiques : Interprétation de données cadastrales.* Thèse de doctorat : Université de Rouen, 1994. 374 p.

[Pavlidis86] **Pavlidis, T.** A vectorizer and feature extractor for document recognition. *Computer Vision, Graphics and Image Processing*, 1986, Vol. 35, p. 111-127.

[Poirier93] **Poirier, F., Julia, L., & Faure, C..** Tapage : édition de tableaux sur ordinateur à stylo. Vers une désignation naturelle. *Proceedings of IHM93, 1993*. p. 45-49.

[Ramachandra80] **Ramachandran, K.** Coding method for vector representation of engineering drawings. In : *Proceedings of the IEEE*, 1980, Vol. 68, N° 7, p. 813-817.

[Roosli95] **Roosli, M. and Monagan, G.** A high quality vectorisation combining local quality measures and global constraints. In : *Proceedings of the 3rd International Conference on Document*

Analysis and Recognition, Montreal (Canada), august 14-16, 1995. Vol. 1, p. 243-248.

[Rosenfeld70] **Rosenfeld, A.** Connectivity in digital picture. *Journal of the Association for Computing Machinery*, 1970, Vol. 17, N°1, p. 146-160.

[Tanigawa94] **Tanigawa, S., Hori, O. and Shimotsuji, S.** Precise line detection from an engineering drawing using a figure fitting method based on contours and skeletons. In : *Proceedings of the 12th International Conference on Pattern Recognition, Jérusalem (Israël), 9-13 octobre, 1994*. Vol. 2, p. 356-360.

[Tombre92] **Tombre, K.** Technical drawing Recognition and understanding : From Pixel to semantics. In : *Proceedings of the IAPR Workshop on Machine Vision and Application. Tokyo (Japon), 7-9 decembre, 1992*. p. 393-401.