Lecture automatique des écritures sur cartes scannées

Automatic Strings Interpretation on Scanned Maps

par Pierrot DESEILLIGNY *, H. LE MEN, G. STAMON **

* *IGN/DT/SR/MATIS – 2, av. Pasteur, F-94160 St Mandé*
** *UFR Math et Info – Rue des Saint Pères, F-75005 Paris*

Résumé

Nous décrivons une méthode de lecture automatique des chaînes de caractères sur cartes scannées. Cette méthode prend en compte les chaînes orientées qui sont fréquentes sur les cartes. L'application se décompose en trois modules principaux : analyse des particules connexes, chaînage des hypothèses, recherche des caractères connectés.
Les résultats obtenus semblent suffisants pour passer au stade opérationnel à condition d'utiliser quelques règles syntaxiques de haut niveau pour améliorer la détection des cas douteux.

Mots clés : Lecture automatique de cartes, systèmes d'informations géographiques, Reconnaissance optique de caractères, invariance par rotation, théorie des graphes.

Abstract

We describe a method for an automatic character string understanding from scanned maps. This method takes into account oriented strings that occur frequently on maps. The application is split in three main modules : connected components analysis, string construction, connected character detection.
Results seem sufficient to allow an operational application if syntactic rules are added to ameliorate the warning process.

Key words : *Map understanding, GIS, OCR, Rotation invariance, Graph theory.*

1. Introduction

1.1. PRÉSENTATION

Cet article décrit une méthode d'extraction et d'identification de chaînes de caractères dans le contexte cartographique. Cette méthode a été développée à l'IGN (Institut Géographique National) dans le cadre d'une étude sur la lecture automatique de cartes scannées. L'objectif général de cette étude est de convertir automatiquement le fonds cartographique existant, sous forme papier, en une base de données d'objets géographiques directement utilisable dans un SIG (Système d'Informations Géographiques).

L'IGN étant le producteur des cartes analysées, la reconnaissance s'effectue directement à partir des planches–mères (éléments d'impression en « noir et blanc ») et on évite ainsi les problèmes de séparation des couleurs (décrits, par exemple, dans [Yan], [Lefrère], [Pierrot]).

De nombreux travaux ont été menés sur l'analyse de documents graphiques tels que, par exemple : analyse du cadastre [Ogier], analyse de partitions musicales [Leplumey], analyse de plans d'ingénieur [Vaxivière]. De manière générale, ce qui différencie la carte des autres documents graphiques est, d'une part, une grande densité d'information et une légende très riche, d'autre part, une très forte normalisation des règles de rédaction.

La démarche générale, pour résoudre les difficultés induites par le premier type de spécificités, est alors de tirer un parti optimal de la connaissance a priori donnée par les règles de rédaction. Ces spécifités cartographiques, et la démarche induite, sont instanciées, au cas particulier de la lecture des écritures, dans les paragraphes 2.3 et 2.4.

1.2. POSITION PAR RAPPORT À D'AUTRES TRAVAUX

La reconnaissance optique de caractères est un des thèmes d'application les plus anciens de la reconnaissance des formes et le chapitre 6 de [Belaïd] constitue une synthèse assez complète des nombreuses méthodes proposées.

Les méthodes de reconnaissance, par appariement direct entre formes et références, sont parmi les plus anciennes utilisées dans la littérature sur la reconnaissance de caractères [Han]. Elles ont été longtemps abandonnées au profit de méthodes d'analyse ou de classification (Bayésienne [Duda], Connexionniste [Cun], Syntaxique [Fu],[Miclet]) après changement d'espace de représentation [Simon], [Nieman] constitue une présentation générale de ces techniques. On attend en général de ces méthodes « indirectes » un meilleur degré de généralisation (l'appariement direct donne effectivement de mauvais résultats avec l'écriture manuscrite) et, parfois, une plus grande rapidité. Cependant, avec l'augmentation de la puissance de calcul des ordinateurs, les méthodes par appariement direct commencent à réapparaître dans un certain nombre de contextes adaptés. [Carrosio], par exemple, décrit une méthode opérationnelle de reconnaissance des points de cote, sur les cartes suisses, basée sur ce principe.

La solution que nous utilisons, pour la normalisation aux rotations, est une variante de la méthode par normalisation des axes d'inertie, décrite par exemple dans [Lefrère], qui constitue une des plus anciennes techniques de reconnaissance invariante par rotation. L'originalité de notre approche réside dans le système de normalisations multiples décrit au 3.1.4.

La construction du vecteur de forme que nous utilisons (normalisation d'histogramme), pour la reconnaissance invariante par « homothétie–translation », est voisine de celle décrite dans [Lee].

Plusieurs auteurs effectuent le chaînage avant la reconnaissance des caractères par détection globale des alignements. Il s'agit en général de contextes où l'on ne dispose pas des fontes. [Fletcher] décrit une « méthode de séparation du texte et des graphiques », applicable à différents types de documents, qui reconnaît les alignements de particules connexes par transformée de Hough. [Likforman] décrit, dans le contexte de l'écriture manuscrite, une méthode de chaînage des composantes connexes inspirée de la « Gestalt ». Ces deux auteurs effectuent le chaînage sur les boîtes d'encombrement des particules connexes, indépendamment de toute reconnaissance.

La solution que nous utilisons, pour le problème du chaînage et la recherche de caractères connectés multiples (non décrite ici), utilise la programmation dynamique. La formulation de ce type de problèmes en « recherche de plus court chemin dans un graphe d'hypothèses » est décrite dans [Rocha] qui l'utilise aussi pour les caractères connectés multiples.

L'organisation générale de notre application, est assez voisine de celle de [Nakamura] qui travaille sur la lecture automatique de cartes japonaises. Cependant, les cartes françaises et japonaises étant peu semblables (surtout pour les écritures), les architectures des différents sous–modules sont assez éloignées.

2. Contexte et choix d'organisation

2.1. CONTEXTE INDUSTRIEL

Le contenu des cartes IGN est insuffisant (précision géométrique et actualité) pour satisfaire aux spécifications d'une base de données géographiques comme la BD–TOPO (base de données topographiques de l'IGN). Il serait néanmoins intéressant de disposer de données structurées, issues de cartes, pour créer des bases de données d'attente (échéance de fin de saisie de la BD–TOPO : 30 ans) ou pour guider l'interprétation automatique à partir de photos.

L'intérêt « secondaire » de l'utilisation envisagée, implique qu'une application de lecture de carte ne sera économiquement rentable que si l'on arrive à une lecture entièrement automatique avec vérification sur les cas douteux alertés par le programme.

2.2. OBJECTIF

Dans l'objectif d'une interprétation complète de la carte, il est normal de commencer par les thèmes les plus « faciles », afin de pouvoir, pas à pas, simplifier la carte. A priori, il n'est pas évident que les caractères constituent le thème le plus facile à extraire. Cependant, il s'agit du seul thème pour lequel on dispose de règles de cohérence de haut niveau (organisation en chaînes) permettant une vérification intra–couche. Il s'agit donc du premier thème extrait, et un des buts de l'extraction des chaînes est la simplification de l'image de la carte.

Cependant, le but essentiel de cette extraction est de pouvoir ultérieurement créer les liens entre objets géographiques et chaînes les qualifiant (route et nom de routes, symboles conventionnels et mots clés...) afin de valider et d'enrichir les différents niveaux d'extraction.

2.3. SPÉCIFICITÉS CARTOGRAPHIQUES

Nous allons décrire les principales spécificités de l'extraction de caractères dans le contexte cartographique.

2.3.1. Facilité

On peut disposer de manière exhaustive des polices présentes sur une facture de carte donnée. Ce nombre de polices varie entre 3 et 10.

Par ailleurs, le temps de calcul n'est pas un critère prépondérant. On peut considérer qu'une nuit de calcul, pour une coupure de carte (200 giga octets à 500 dpi), est un ordre de grandeur acceptable.

2.3.2. *Difficultés*

Les caractères ne constituent qu'une petite partie de l'information présente. Ceux qui ont des formes simples (« I » – « l » – « j »...) peuvent se confondre facilement avec d'autres morceaux de la carte (traits de routes...). Par ailleurs, les caractères sont souvent connectés à d'autres éléments.

La difficulté la plus spécifique au contexte est la présence de chaînes obliques et courbes. Les connaissances dont on dispose sur l'orientation des chaînes sont que :

- les chaînes sont régulièrement courbes à l'intérieur d'un même mot (c.à.d. à peu près disposées sur un arc de cercle), cette propriété est difficile à utiliser dans les phases d'analyse;
- l'angle entre deux caractères consécutifs d'une même chaîne ne dépasse pas 30°;
- la direction « moyenne » des chaînes doit être comprise entre $-90°$ et $+90°$; compte tenu des chaînes courbes, ceci conduit à accepter des « hypothèses–caractères » entre $-120°$ et $+120°$.

Les polices sont présentes à des corps très variables; on compte jusqu'à huit corps différents pour la même police. Compte tenu de l'orientation, la reconnaissance doit donc être invariante aux similitudes.

Il n'y a pas de régularité dans la disposition des chaînes. Ceci exclut les techniques de reconnaissance globale des blocs de caractères (« layout analysis »). Par ailleurs, l'espacement entre les caractères d'un même mot est très variable; pour certaines chaînes « à disposition » (chaînes qui sont disposées de manière à recouvrir au mieux les éléments surfaciques qu'elles qualifient) l'espace entre les caractères d'un même mot peut atteindre 4 fois la largeur du « e ».

2.4. MÉTHODOLOGIE

Les difficultés propres au contexte cartographique empêchent que l'on puisse utiliser des logiciels commerciaux. On cherche donc à utiliser au maximum les facilités liées au contexte.

Pour ceci on s'oriente, pour les parties de reconnaissance de formes, vers des méthodes d'appariement direct des références. Ces méthodes présentent les inconvénients d'être un peu lentes et inadaptées à la reconnaissance omni–fontes mais nous avons vu que ceci n'est pas un réel inconvénient dans notre contexte.

Cependant il faudra effectuer un certain nombre de normalisations et de présélections rapides avant d'obtenir des temps de calcul d'un ordre de grandeur acceptable.

2.5. ARCHITECTURE DE L'APPLICATION

Nous effectuons d'abord les remarques qui nous ont conduit à retenir une architecture.

Il est très rare que dans une même chaîne tous les caractères soient connectés au reste de l'environnement; on a donc intérêt à terminer par la recherche des caractères connectés en s'appuyant sur les caractères isolés que l'on aura reconnus.

Il est souvent difficile de prendre une décision de reconnaissance non ambiguë au niveau de la particule connexe; citons, parmi beaucoup d'exemples : confusion entre « s » et « S » à un facteur d'homothétie près, confusion entre « n » et « u » à π près, confusion entre « I » et un bâtiment, confusion, parfois, entre les mêmes caractères de différentes polices...

Par contre, la plupart de ces ambiguïtés peuvent être levées si on sélectionne, pour chaque particule, celle des différentes hypothèses qui permet de former une chaîne cohérente, du point de vue de la police et des caractéristiques géométriques (position, échelle, orientation), avec les particules voisines.

Il existe quelques ambiguïtés résiduelles, telles que la confusion entre « 0 » et « O » (zéro et « O » majuscule), qui ne peuvent être résolues sur de simples critères de formes (même à police, position, échelle et orientation fixées). Ces ambiguïtés doivent être résolues en utilisant des connaissances syntaxiques; pour utiliser ces connaissances, il est souhaitable de raisonner sur des chaînes complètes, elles ne sont donc utilisées qu'à la fin.

Ces remarques nous ont conduits à choisir l'architecture suivante :

1) analyse des particules connexes; pour chaque particule, on renvoie un ensemble, éventuellement vide, d'hypothèses–caractères;

2) chaînage des hypothèses; on recherche la sélection d'hypothèses qui permet les meilleurs regroupements;

3) recherche des caractères connectés à partir des résultats, fiabilisés, de l'étape précédente;

4) nouveau chaînage des hypothèses, avec utilisation de connaissances syntaxiques élémentaires.

Nous allons maintenant décrire ces différents modules en insistant plus particulièrement sur l'analyse des particules connexes.

3. Analyse des particules connexes

Dans toute cette partie on définit une forme comme une application de R^2 dans R. Le calcul des particules connexes au sens de la 4–connexité ([Belaïd] pp. 31–34) de l'image s'effectue sur une image binarisée avec un seuil absolu. Le module d'analyse des particules connexes se décompose en 5 étapes.

Après une normalisation aux rotations (3.1), on calcule un vecteur de forme invariant par homothétie–translation et on présélectionne un certain nombre d'hypothèses caractères (3.2). Compte tenu de la normalisation aux rotations, cette présélection est globalement invariante aux similitudes. On estime ensuite, pour chaque référence présélectionnée, les paramètres de transformation géométrique (similitude) entre la forme et la référence (3.3). Après un affinement de ces paramètres (3.4), on calcule une image

de la forme appariable avec chaque référence présélectionnée qui permet la sélection finale (3.5).

3.1. ROTATIONS

3.1.1. Formalisation

On décide d'aborder le problème de la reconnaissance invariante par rotation par une normalisation sans changement d'espace de représentation. On définit une normalisation (aux rotations) par un opérateur qui, à chaque forme ϕ, associe une rotation $\theta(\phi)$ telle que, pour toute forme ϕ et toute rotation β :

$$\theta(\phi o\beta) = \theta(\phi) o\beta^{-1}$$

Avec un tel opérateur, on peut associer à chaque forme ϕ une forme normale $\tilde{\phi}$, invariante aux rotations de ϕ, en posant :

$$\tilde{\phi} = \phi o\theta(\phi)$$

Si une telle normalisation existe, avec de bonnes propriétés de stabilité, il suffit, pour décider si deux formes ϕ et φ sont semblables, à une rotation près, de comparer $\tilde{\phi}$ et $\tilde{\varphi}$; si on décide qu'elles sont semblables alors $\theta(\phi)^{-1} o\theta(\varphi)$ donne une estimation de la rotation permettant de passer de ϕ à φ.

Plusieurs difficultés empêchent que l'on puisse suivre, telle quelle, une procédure aussi simple.

3.1.2. Symétries

On définit qu'une forme ϕ admet une symétrie d'ordre k si $\phi = \phi o\beta_k$, où β_k désigne la rotation d'angle $2\pi/k$; par exemple, $k = 2$ pour « S », $k = 3$ pour « Δ »...

Un premier problème est dû à l'existence de formes invariantes par rotation. L'opérateur θ ne peut pas exister s'il existe des formes ayant des ordres de symétrie supérieurs à 1 car, si cela était, on aurait :

$$\theta(\phi) = \theta(\phi o\beta_k) = \theta(\phi) o\beta_k^{-1}$$

Soit n le degré de symétrie maximal de l'ensemble des formes à reconnaître, on peut contourner ce problème en décidant que l'opérateur θ renvoie des rotations à $k2\pi/n$ près. Cela allonge un peu les calculs car il ne suffit plus de comparer $\tilde{\phi}$ et $\tilde{\varphi}$; il faut en fait comparer $\tilde{\phi}$ et tous les $\tilde{\varphi} o\beta_k$ pour k dans $[0, n[$.

Dans l'alphabet latin, il existe couramment des lettres ayant un degré de symétrie d'ordre 2 (« S », « 0 », « N », « I »...). Inversement il n'existe pas de formes admettant des degrés de symétrie supérieurs (du moins sur les polices des cartes IGN où le « o » est assez allongé). On décide donc de rechercher une normalisation à π près.

3.1.3. Axes d'inertie

Une normalisation, couramment décrite dans la littérature et répondant aux critères précédents, est la normalisation par axes d'inertie. On définit $\theta(\phi)$ comme étant l'angle (au sens angle de droite) que fait l'axe principal d'inertie de la forme avec un axe de référence, par exemple la deuxième diagonale. La forme normalisée $\tilde{\phi}$ est alors la forme ϕ tournée d'un angle qui aligne son axe principal sur la deuxième diagonale.

La figure 1 montre différents exemplaires de « a » et, en dessous, leurs formes normalisées. On peut vérifier que, à π près, on obtient bien une forme invariante par rotation.

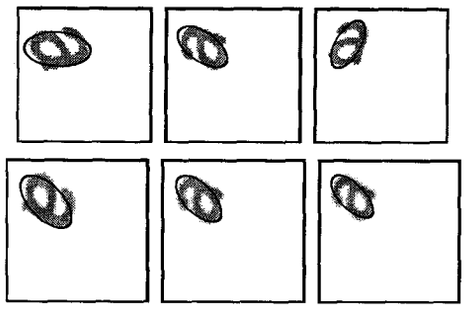


Figure 1. – «a» extraits de la carte et, en dessous, leurs formes normalisées aux rotations.

3.1.4. Instabilité

La normalisation par axes d'inertie est d'autant plus stable que l'ellipse d'inertie est allongée. La majorité des formes de caractères possèdent un allongement vertical et cette normalisation ne pose, en général, pas de problème. Cependant, pour les caractères ayant une ellipse d'inertie presque circulaire, on obtient une normalisation très instable. La figure 2 représente différents « M » de la carte et, en dessous, leurs formes « normalisées »; on s'aperçoit que, alors que les « M » initiaux sont tous horizontaux, les formes normalisées sont très différentes.

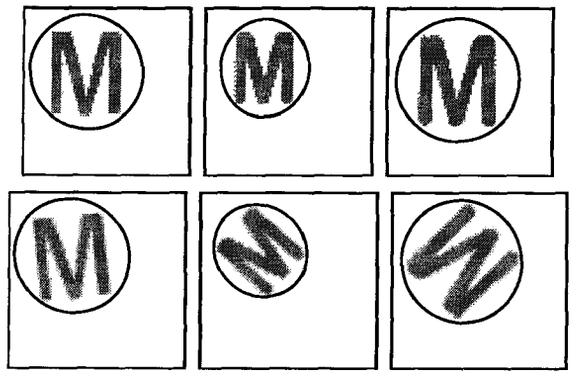


Figure 2. – Instabilité de la normalisation aux rotations avec des « M » extraits de la carte.

On pourrait essayer de résoudre ce problème en cherchant à définir une autre normalisation aux rotations qui n'aurait pas ces propriétés d'instabilité. En fait cette voie de recherche est sans issue; on peut montrer, de manière formelle, que, indépendamment des problèmes de symétrie, toute normalisation aux rotations, d'un espace de formes non trivial, admet nécessairement des points de discontinuité. La démonstration rigoureuse de ce résultat est assez longue et est donnée dans [PIERROT 94]; la démonstration suit le schéma suivant :

• *on exhibe un sous espace des formes, sans forme symétrique, invariant par rotation et isomorphe à la sphère de* R^4*;*

• *on démontre que sur ce sous espace, si la normalisation continue existe, les applications* $\phi \rightarrow \tilde{\phi}$ *et* $\phi \rightarrow \theta(\phi)$ *sont continues;*

• *on démontre que « l'application réciproque »* $(\tilde{\phi}, \theta) \rightarrow \tilde{\phi} o \theta$ *est aussi continue;*

• *on en déduit que la sphère de* R^4 *est homéomorphe au produit cartésien du cercle et d'un certain espace ce qui est topologiquement impossible (car la sphère de* R^4 *est simplement connexe).*

Une solution, pour contourner cette absence d'opérateurs de normalisation continus, est de remarquer que l'allongement de l'ellipse d'inertie est un estimateur a priori de la stabilité de la normalisation que l'on va effectuer. Si l'on dispose d'autres opérateurs de normalisation, on peut utiliser la normalisation par axes d'inertie pour les formes ayant un allongement suffisant et utiliser un de ces autres opérateurs pour les formes ayant une ellipse presque circulaire. De manière plus formelle, on décide de reformuler le procédé de normalisation qui a été défini au 3.1.1. On définit une normalisation par la donnée de n couples d'opérateurs θ_k, F_k, tels que :

• θ_k *est un opérateur allant de l'espace des formes vers les rotations;*

• $\theta_k(\phi o \beta) = \theta_k(\phi) o \beta^{-1}$ *pour toute forme* ϕ *et toute rotation* β*;*

• $F_k(\phi)$ *est un estimateur de la stabilité de* θ_k *au point* ϕ *de l'espace des formes.*

Par ailleurs, on pose $\tilde{\phi_k} = \phi o \theta_k(\phi)$ et ainsi, la comparaison entre une forme ϕ et une référence φ est effectuée en comparant $\tilde{\phi_m}$ et $\tilde{\varphi_m}$, où m est l'entier réalisant le maximum des $F_k(\phi)$. Si ces formes sont jugées semblables, on estime alors la rotation permettant de passer de ϕ à φ par $\theta_m(\phi)^{-1} o \theta_m(\varphi)$.

3.1.5. Normalisation par les contours

Pour utiliser le schéma formel que nous venons de décrire, nous devons définir d'autres opérateurs de normalisation qui ne soient pas instables pour les mêmes formes que la normalisation par axes d'inertie. Pour ceci, on remarque que les contours du « M » sont orientés de manière privilégiée dans la direction verticale. On décide donc d'utiliser un opérateur qui quantifie la notion intuitive de direction principale des contours.

De manière formelle, on considère la distribution « champ de vecteur gradient » de la forme. La figure 3 représente cette distribution pour le deuxième « M » de la figure 2.

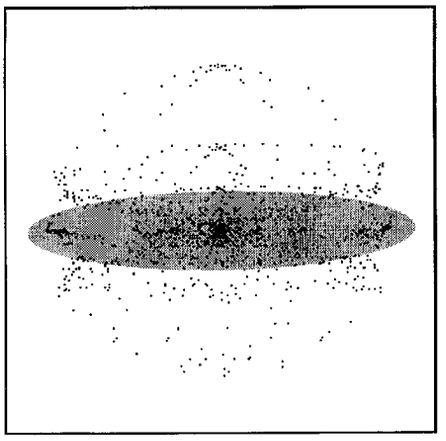


Figure 3. – Distribution du champ de vecteur gradient d'un «M» et son ellipse d'inertie.

On définit alors le deuxième opérateur de normalisation comme l'axe principal d'inertie de cette distribution. L'estimateur de stabilité choisi est encore l'allongement de l'ellipse d'inertie; ce choix présente l'avantage de fournir un estimateur de stabilité « naturellement » comparable avec le précédent.

En théorie, il peut exister des formes qui sont instables pour les deux opérateurs que nous avons définis. C'est pourquoi, dans un souci de généralité, nous avons présenté le schéma formel avec un nombre arbitraire d'opérateurs. En pratique ces deux opérateurs sont suffisants pour que, sur toutes les références utilisées, un au moins de ces deux opérateurs soit stable. Sur la figure 3 on a représenté, en sous–impression, l'ellipse d'inertie de la distribution de gradient du « M »; elle possède un allongement suffisant pour assurer une bonne normalisation. Comme la définition de nouveaux opérateurs n'est pas évidente, nous nous sommes limités à ce jeu de deux opérateurs.

3.2. PRÉSÉLECTION

Une fois les formes normalisées aux rotations, on souhaite définir un vecteur de forme, invariant par homothétie–translation, qui permette d'effectuer une présélection rapide des références. Ces vecteurs sont, évidemment, calculés sur les formes normalisées des références et des formes à identifier.

Une façon très simple de définir un tel vecteur est de calculer une imagette réduite de la forme, de taille $n \times n$, où n est un petit entier (ordre de grandeur 5 à 10). Le facteur d'homothétie entre la forme et l'imagette est alors calculé sur la forme avec un opérateur tel que : moyenne quadratique de la distance au centre de gravité, plus grande dimension de la boîte englobante...

En partant de ce principe, nous avons préféré utiliser une construction un peu plus complexe mais, expérimentalement, plus robuste : l'égalisation d'histogramme. La forme est assimilée à une distribution de points, et on effectue sur cette distribution le

seul changement de coordonnées, séparable en x et y, qui permette d'obtenir des probabilités projetées constantes sur l'intervalle $[0 \ldots 1]$ et nulles en dehors. Nous détaillons maintenant le procédé de construction dans le cas d'images continues.

Soit une forme, on pose

$$\phi = \frac{\varphi}{\iint \varphi \partial u \, \partial v}$$

afin d'assimiler la forme à une probabilité. On calcule alors les histogrammes projetés :

$$fx(u) = \int \phi(u,v)\, \partial v$$
$$fy(v) = \int \phi(u,v)\, \partial u$$

puis les histogrammes cumulés :

$$Fx(a) = \int_{u<a} fx(u)\, \partial u$$
$$Fy(b) = \int_{v<b} fy(v)\, \partial v$$

On définit alors les fonctions inverses de ces histogrammes cumulés :

$$\xi x = Fx^{-1}$$
$$\xi y = Fy^{-1}$$

$(\xi x, \xi y)$ fournit alors le changement de variable définissant l'histogramme égalisé EG :

$$EG(\alpha,\beta) = \xi' x(\alpha) \xi' y(\beta) \phi(\xi x(\alpha), \xi y(\beta));$$

Ce procédé doit être étendu aux images discrètes. Le calcul des histogrammes projetés et cumulés ne pose pas de problème. On définit l'imagette discrète $EG(i,j)$ par une valeur intégrale :

$$\begin{aligned} EG(i,j) &= \int_{\frac{i}{n}}^{\frac{i+1}{n}} \int_{\frac{j}{n}}^{\frac{j+1}{n}} EG(\alpha,\beta)\, \partial\alpha\, \partial\beta \\ &= \int_{\frac{i}{n}}^{\frac{i+1}{n}} \int_{\frac{j}{n}}^{\frac{j+1}{n}} \xi' x(\alpha)\, \xi' y(\beta) \phi(\xi x(\alpha), \xi y(\beta)) \partial\alpha\, \partial\beta \end{aligned}$$

En refaisant un changement de variables en $(\xi x, \xi y)$, on obtient :

$$EG(i,j) = \int_{\xi x\left(\frac{i}{n}\right)}^{\xi x\left(\frac{i+1}{n}\right)} \int_{\xi x\left(\frac{j}{n}\right)}^{\xi x\left(\frac{j+1}{n}\right)} \phi(x,y)\, \partial x\, \partial y$$

On peut donc calculer $EG(i,i)$ directement à partir de l'intégrale de ϕ; pour effectuer le calcul des $\xi x\left(\frac{k}{n}\right)$ et $\xi y\left(\frac{k}{n}\right)$ à partir de l'histogramme cumulé discret, on procède par interpolation linéaire et, pour le calcul de l'intégrale de $\phi(x,y)$ sur des rectangles à coordonnés réelles, on répartit la contribution de pixels traversés par les $\xi x\left(\frac{k}{n}\right)$ et $\xi y\left(\frac{k}{n}\right)$ au prorata des surfaces.

La figure 4 illustre, sur un « e » de la carte la procédure de construction de l'imagette. Au–dessus et à droite de l'image, on a représenté les histogrammes cumulés. Les barres blanches verticales (resp. horizontales) sont positionnées aux abscisses (resp. ordonnées) des $\xi x\left(\frac{k}{n}\right)$ et $\xi y\left(\frac{k}{n}\right)$. Sur cette figure, on a choisi $n = 13$, afin d'améliorer la lisibilité.

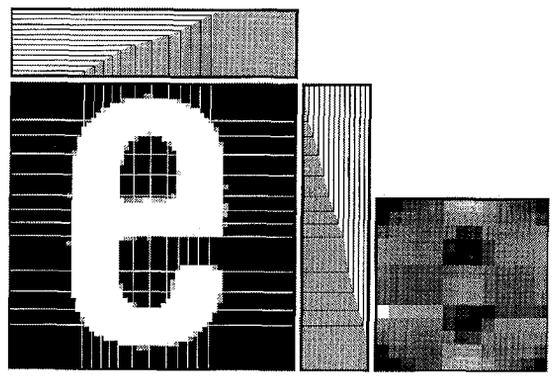


Figure 4. – Un «e» et la construction de la normalisation d'histogramme avec $n = 13$.

Le vecteur de forme obtenu est invariant aux « homothéties–translations » mais aussi à tous les changements de coordonnées séparables en x et y. Comme il s'agit d'une phase de présélection, cet aspect sur–normalisateur n'est pas très pénalisant. Pour mesurer un écart entre les références et la forme, on utilise le cœfficient de corrélation centré entre les vecteurs de forme.

3.3. ESTIMATION DES PARAMÈTRES DE SIMILITUDE

Soit $n\ x\ n$ la dimension du vecteur de forme, soient $T_x[i]$ et $T_y[i]$ les tableaux contenant les $\xi x\left(\frac{k}{n}\right)$ et $\xi y\left(\frac{k}{n}\right)$, soient $T'_x[i]$ et $T'_y[i]$ les tableaux équivalents pour les références, si la forme se déduit de la référence par une homothétie–translation de paramètres $(\rho, \Delta_x, \Delta_y)$, on doit avoir les relations :

$$\begin{cases} T_x[i] = \rho * T'_x[i] + \Delta_x \\ T_y[i] = \rho * T'_y[i] + \Delta_y \end{cases}$$

Ces relations doivent être vérifiées pour i dans $]0, n[$. La connaissance de T_x, T'_x, T_y, T'_y permet alors d'estimer $(\rho, \Delta_x, \Delta_y)$ par résolution aux moindres carrés du système précédent.

On obtient ainsi une estimation des paramètres d'homothétie–translation entre les formes normalisées aux rotations. Par ailleurs, on a une estimation des paramètres de translation–rotation entre

les formes initiales et les formes normalisées aux rotations. En composant ces estimations des transformations géométriques, on obtient une estimation des paramètres de similitude permettant de passer de la forme à la référence.

3.4. AFFINEMENT DES PARAMÈTRES DE SIMILITUDE

En général, l'estimation des paramètres de similitude est assez bonne. Cependant il peut arriver, lorsque la forme est légèrement bruitée, que les paramètres obtenus ne permettent pas un appariement fin. La partie gauche de la figure 5 représente l'image de gradient d'un « E » de la carte, sur cette image on a surimposé le contour du « E » de référence obtenu en tenant compte des paramètres de la similitude estimée.

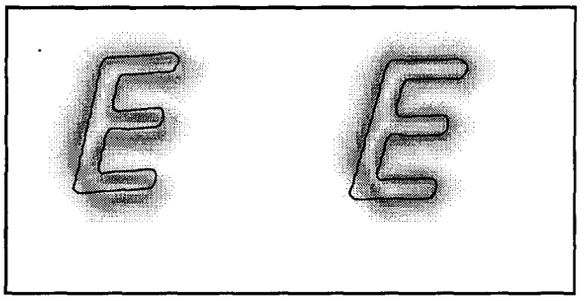


Figure 5. – Appariement des contours de la référence sur la forme avant, et après, optimisation.

Afin d'obtenir une reconnaissance résistante à ce type de configuration, on va chercher à améliorer les paramètres de la similitude. On formalise ce problème comme un problème d'optimisation dans l'espace des paramètres de la similitude : « trouver la similitude qui maximise la corrélation entre le champ de vecteurs gradients observé (celui de la forme) et celui prédit par transformation (par la similitude) du champ de vecteur de la référence ». L'optimisation porte sur tous les paramètres de la similitude, y compris les paramètres de translation (qui permettent de fixer le centre de l'homothétie–rotation).

Pour résoudre ce problème d'optimisation, on dispose d'une solution « proche » de l'optimum grâce aux paramètres estimés. On peut donc utiliser des algorithmes d'optimisation locale, soit, en notant $C(S)$ la fonction de qualité d'une similitude :

1) poser S_0 = similitude estimée;

2) soit S_i la solution courante, recherche d'une direction d'optimisation, par linéarisation du problème (calcul de la défférentielle de C en S_i) et résolution exacte du problème linéarisé qui se ramène à un problème de moindres carrés;

3) soit ΔS_i la direction choisie, résoudre le problème de l'optimisation à une dimension $f_i(\lambda) = C(S_i + \lambda \Delta S_i)$, on utilise une recherche dichotomique (par exemple, Golden Search [Press]);

4) soit $\lambda_i^{\max}$ la solution optimale, si condition d'arrêt fin de l'algorithme sinon poser $S_{i+1} = S_i + \lambda_i^{\max} \Delta S_i$ et retour en 2.

La condition d'arrêt est vérifiée lorsque l'amélioration de la corrélation par rapport à l'étape précédente, est inférieure à un seuil ou lorsque le nombre d'itérations est supérieur à un seuil.

Comme cette étape est un peu longue, le calcul n'est effectué que sur quelques points (de 20 à 30) de la référence; ces points ont été pré–calculés automatiquement de manière à être régulièrement espacés (au sens de l'abscisse curviligne) le long des contours des références. Sur la figure 5, l'image de droite représente la solution obtenue après amélioration.

3.5. SÉLECTION

On utilise les paramètres estimés pour calculer une image de la référence appariable avec la forme. La sélection finale est effectuée en calculant le coefficient de corrélation centré sur les champs de vecteurs gradients.

4. Chaînage

4.1. FORMALISATION GÉNÉRALE

La figure 6.b représente les données dont on dispose à la fin de l'étape d'analyse des particules connexes pour l'image de la figure 6.a. Pour chaque particule, on dispose d'un ensemble d'hypothèses–caractères (code ASCII, police, échelle, position, orientation, taux de confiance). On note $\{P_0, \ldots, P_n\}$ les particules; on note $P_i = \{h_j^i\}$ où les h_j^i sont les hypothèses–caractères générées par la particule P_i.

L'objectif de la phase de chaînage est à la fois de lever l'ambiguïté d'interprétation et de structurer les « hypothèses caractères » sélectionnées en chaînes. La structure de données représentant un chaînage est un ensemble de listes d'hypothèses $\{C_1, \ldots, C_k\}$ (k est la cardinalité du chaînage). Un tel chaînage vérifie que,

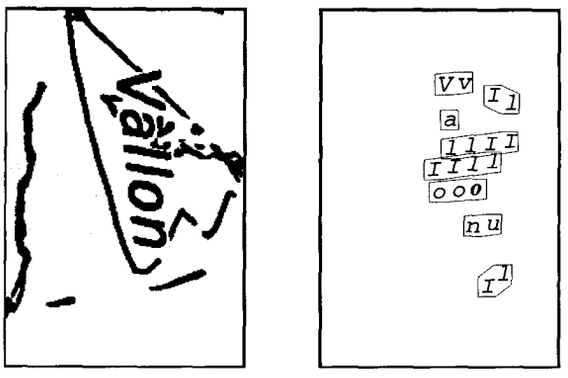


Figure 6.a et 6.b. – Un extrait de la carte (6.a) et les hypothèses générées (6.b).

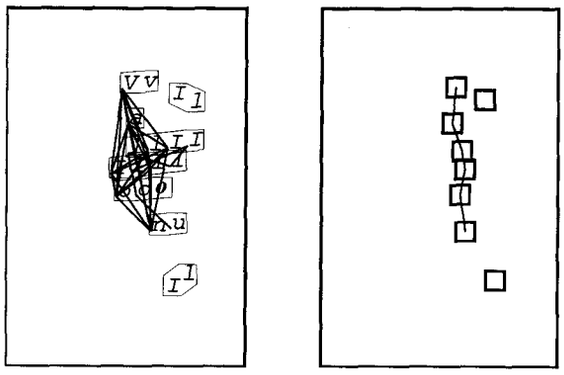


Figure 6.c et 6.d. – Le graphe des hypothèses (6.c) et la forêt de poids minimum sur les particules (6.d).

pour chaque particule P_i une, et une seule, des hypothèses qu'elle a générées appartient à une des listes C_k. C'est cette propriété qui assure qu'un chaînage lève l'ambiguïté d'interprétation.

Chaque liste d'hypothèses C_i représente une chaîne éventuellement incomplète; l'ordre de représentation des hypothèses a donc une importance.

La figure 6.e représente un chaînage des données de la figure 6.b. Afin de pouvoir calculer un « bon » chaînage à partir d'un ensemble d'ensembles d'hypothèses, on doit introduire des connaissances; ces connaissances doivent, d'une part, permettre de spécifier que les C_k vérifient un certain nombre de règles de cohérence (hypothèses de la même police, paramètres géométriques compatibles, règles syntaxiques sur les codes ASCII...); d'autre part, ces connaissances doivent permettre de quantifier la qualité des chaînages cohérents (ceux qui contiennent les hypothèses les plus probables et les plus voisines du point de vue des paramètres géométriques).

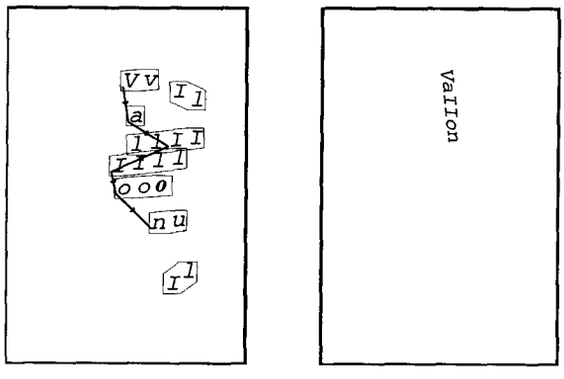


Figure 6.e et 6.f. – Un chaînage (6.e) et son interprétation (6.f).

On formalise alors le problème comme la recherche du chaînage cohérent de cardinalité minimale et de moindre coût (ces deux conditions sont en général contradictoires).

4.2. HYPOTHÈSES SIMPLIFICATRICES

On admet que les notions de cohérence et de qualité du chaînage peuvent se définir à partir de relations élémentaires définies sur les couples d'hypothèses. On dispose des couples d'hypothèses cohérentes (mêmes polices, paramètres géométriques voisins...); cet ensemble permet de définir un graphe, généralement assez dense (figure 6.c). Chaque couple d'hypothèses cohérentes est affecté d'un coût.

On définit alors qu'un chaînage est cohérent si, pour toutes les listes Ci, tous les couples d'hypothèses consécutifs de ces listes sont cohérents. Le coût d'un chaînage est égal à la somme des coûts des couples d'hypothèses.

On peut maintenant formaliser le problème du chaînage optimal comme la recherche du chaînage qui, par ordre de priorité (afin de résoudre le conflit entre cardinalité et coût), est cohérent, de cardinalité minimale et de coût minimal.

4.3. ALGORITHME

L'algorithme que nous utilisons pour résoudre le problème ainsi formalisé est décrit entièrement dans [Pierrot].

On remarque d'abord que le problème admet une solution exacte par programmation dynamique quand on peut ordonner les particules de telle manière que tous les couples cohérents soient de la forme $(h^i_j, h^{i+1}_{j'})$ ou $(h^{i+1}_j, h^i_{j'})$. On calcule une forêt de poids minimum (pondérée par la distance euclidienne) sur les couples de particules ayant généré au moins un couple d'hypothèses compatibles, (cette forêt est représentée sur la figure 6.d). On ne conserve que les couples d'hypothèses qui correspondent à des couples de particules de cette forêt; on obtient un graphe avec lequel on peut, sur « presque toutes » les composantes, utiliser l'algorithme exact de programmation dynamique. Enfin, on explore combinatoirement toutes les décompositions possibles de la forêt en un ensemble maximal de chaînes; en général, comme sur la figure 6.d, la forêt est déjà un ensemble de chaînes.

L'image 6.f représente le type de données dont on dispose à la fin de l'étape de chaînage. Nous allons maintenant détailler quelques points de cet algorithme.

4.4. FONCTIONS DE COHÉRENCE ET DE COÛT

Deux hypothèses $h1$ et $h2$ sont cohérentes lorsqu'elles ont la même police $(R1)$, la même échelle $(R2)$, la même orientation $(R3)$ et qu'elles définissent un couple compatible de caractères horizontaux dans « leur repère commun » $(R4)$. $R1$ est une règle stricte, $R2$ aussi aux incertitudes d'estimation près. $R3$ est une règle souple à cause des chaînes courbes. Enfin $R4$ signifie que, dans le « repère commun », $h1$ et $h2$ ont même abscisse $(R4a)$, même ordonnée $(R4b)$ et que $h1$ est à droite de $h2$ $(R4c)$. $R4a$

est une règle stricte aux erreurs d'estimation près, $R4b$ est une règle très souple à cause de l'espacement variable des chaînes et $R4c$ est une règle stricte.

La règle $R1$ ne pose pas de problème. Soient $(x_1, y_1, \rho_1, \theta_1)$ et $(x_2, y_2, \rho_2, \theta_2)$ les paramètres géométriques des deux hypothèses, pour les règles $R2$ et $R3$ on définit deux seuils $\Delta\rho^{\max}$ et $\Delta\theta^{\max}$ max et on impose $\Delta\theta = |\theta_1 - \theta_2| < \Delta\theta^{\max}$ et $\Delta\rho = |\rho_1 - \rho_2| < \Delta\rho^{\max}$.

Pour la règle $R4$, on doit définir précisément les abscisses et ordonnées dans un repère commun. On définit d'abord une orientation moyenne $\theta = \frac{(\theta_1+\theta_2)}{2}$ et une échelle moyenne $\rho = \sqrt{\rho_1\rho_2}$.

On définit alors Δx et Δy : les différences d'abscisses et ordonnées des repères des hypothèses vues dans un repère ayant subi une homothétie–rotation de paramètre (ρ, θ). Les figures 7.a et 7.c illustrent ces définitions.

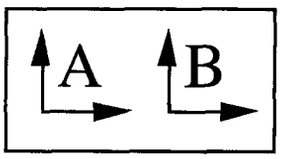


Figure 7.a. – Références dans leurs repères.

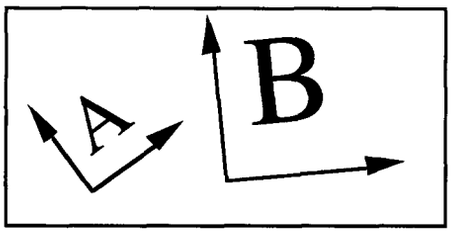


Figure 7.b. – Particules connexes et leurs repères associés.

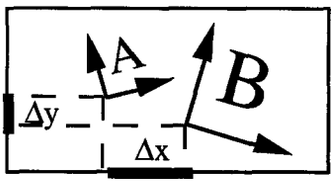


Figure 7.c. – Particules connexes vues dans leur repère commun.

On définit deux seuils $\Delta x^{\max}$ et $\Delta y^{\max}$ max et la règle $R4$ devient :

$$\bullet\ 0 < \Delta x < \Delta x^{\max}$$

$$\bullet\ |\Delta y| < \Delta y^{\max}$$

Enfin, pour la fonction de coût, on choisit :

$$\text{Coût}(h1, h2) = K * (2 - \text{Corr1} - \text{Corr2}) + \frac{\Delta\theta}{\Delta\theta^{\max}} + \frac{\Delta\rho}{\Delta\rho^{\max}} + \frac{\Delta x}{\Delta x^{\max}} + \frac{|\Delta y|}{\Delta y^{\max}}$$

Le premier terme permet de privilégier les hypothèses les plus fiables; Corr1 et Corr2 sont les cœfficients de corrélations calculés au 3.5 et K est une constante permettant de pondérer l'importance relative de cette confiance accordée sur les critères de forme par rapport aux écarts géométriques.

4.5. PROGRAMMATION DYNAMIQUE DANS LE CAS LINÉAIRE

Afin d'alléger l'exposé, nous décrivons le fonctionnement de l'algorithme sur un exemple artificiel. Soient l'image de la figure 8.a, et le graphe, résultant du filtrage par arbre, de la figure 8.b .

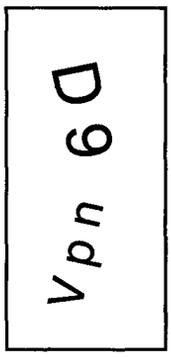


Figure 8.a. – Une image artificielle.

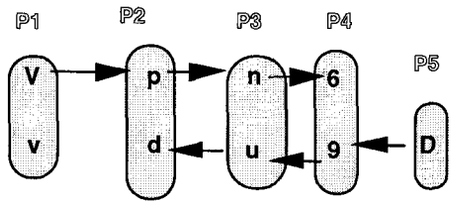


Figure 8.b. – Un graphe d'hypothèses linéaire.

A partir de ce graphe, on définit un nouveau graphe (illustré sur la figure 9) de la manière suivante :

• chaque hypothèse caractère $h^i j$ génère deux sommets $[h^i j]^+$ et $[h^i j]^-$; intuitivement, le sommet $[h^i j]^+$ correspond à « Pi s'interprète comme $h^i j$ et elle appartient à une chaîne lue de gauche à droite »; symétriquement $[h^i_j]^-$ correspond à une lecture de droite à gauche (l'ordre est, a priori, arbitraire);

• chaque arc $(h^i j, h^{i+1} j')$ génère un arc $([h^i j]^+, [h^{i+1} j']^+$ dans le nouveau graphe;

• symétriquement $(h^{i+1} j, h j')$ génère $([h^i j]^-, [h^{i+1} j']^-)$;

• on rajoute tous les arcs, non encore présents, entre hypothèses consécutives; ces arcs correspondront à des coupures de chaîne; afin de privilégier la cardinalité, on leur affecte un poids « infini ».

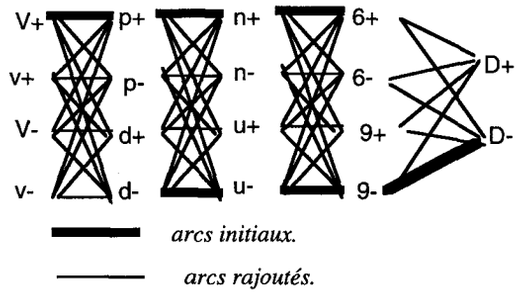


Figure 9. – Le graphe construit.

Pour trouver la solution optimale du chaînage avec ce graphe, on procède alors ainsi :

• on calcule le plus court chemin entre un sommet de $P1$ et de $P5$ dans ce nouveau graphe;

• supposons que ce plus court chemin soit $(V+, p+, n+, 9-, D-)$; il sera, par exemple, meilleur que $(V+, p+, n+, 6+, D-)$ parce que $(n+, 6+)$ aura une moins bonne cohérence géométrique que $(9-, D-)$;

• on coupe le plus court chemin au niveau des arcs de poids infini;

• ici l'arc $(n+, 9-)$ a un poids infini, on obtient donc les deux chemins $(V+, p+, n+)$ et $(9-, D-)$;

• enfin, on inverse les chemins formés de sommets $[h^i j]^-$;

• ici $(9-, D-)$ donne $(D-9-)$;

• on obtient donc les deux chemins (Vpn) et $(D9)$ qui constituent la solution du chaînage optimal.

4.6. EXPLORATION COMBINATOIRE

Pour les cas rares où une composante du graphe résultant de l'arbre de poids minimal n'est pas une chaîne, on explore combinatoirement la décomposition de ce graphe en un minimum de chaînes.

Soit un arbre qui n'est pas une chaîne, il possède au moins trois sommets de degré un. Alors, pour chaque paire possible de sommets de degré un, on calcule l'unique chemin P entre ces deux sommets. Comme P obéit au cas linéaire, on peut lui appliquer l'algorithme de programmation dynamique. Pour le complémentaire de P, on appelle récursivement l'algorithme d'exploration combinatoire.

4.7. COMPLEXITÉ

Bien qu'il soit assez complexe à définir, cet algorithme de chaînage obtient, en pratique, des temps de calcul acceptables pour trois raisons. D'abord, l'algorithme de programmation dynamique a un temps linéaire en fonction du nombre d'arêtes (plus court chemin dans un graphe sans circuit, [Gondran] pp. 45-46). Ensuite, le calcul de l'arbre de poids minimum est en $N \log(N)$, [Gondran] pp. 112-117. Enfin, bien que la phase d'exploration combinatoire ait une mauvaise complexité théorique (elle est approximativement, pour chaque composante, en « factorielle de $N-2$ où N est le nombre de sommets de degré différent de 2 ») , en pratique, nous n'avons jamais rencontré de composante à plus de 4 sommets de degré différent de 2.

Actuellement, la phase de chaînage représente 4% du temps de calcul total.

5. Recherche des caractères connectés

5.1. PRINCIPE

L'étape de chaînage a permis de sélectionner des hypothèses fiables. Il existe encore quelques ambiguïtés qui ne peuvent être résolues que sur des critères syntaxiques, mais ces ambiguïtés ne sont pas gênantes pour la recherche des caractères connectés. Les informations nécessaires, pour limiter cette recherche, sont la police, l'échelle, l'orientation et la position géométrique approximative des caractères recherchés. Le code ASCII des caractères voisins serait de peu d'utilité.

On va alors rechercher des caractères connectés au voisinage des hypothèses sélectionnées. Cette recherche se fera de manière systématique à gauche et à droite des extrémités des chaînes, elle se fera entre les couples de caractères consécutifs s'il y a l'espace suffisant pour intercaler un caractère.

On renvoie encore des ensembles d'hypothèses puisqu'il existe des ambiguïtés (telles que « 0 » – « O ») qui ne peuvent être levées que sur des critères syntaxiques.

La recherche s'effectue par appariement direct des références sur l'image. L'appariement des champs de vecteur gradient donne un critère très sélectif mais très sensible aux variations des paramètres géométriques. On effectue donc la recherche en deux temps.

Une présélection rapide des références possibles est d'abord effectuée par appariement des fonctions d'extinction (c.à.d les fonctions « distance au complémentaire ») en se limitant au squelette des références. L'appariement de ces primitives ne permet pas une sélection sévère mais, par contre, il est très résistant aux imprécisions sur l'évaluation des paramètres géométriques des références cherchées.

Puis on vérifie et on améliore la détermination des positions par appariement des champs de vecteur gradient, à partir des hypothèses sélectionnées précédemment. Cette corrélation est effectuée en se limitant aux points de la référence, afin de ne pas être influencé par l'environnement du caractère.

On obtient ainsi un certain nombre d'« hypothèses–caractères connectés ».

5.2. CONFUSIONS

Pour sélectionner les caractères connectés et leur attribuer un taux de confiance, il n'est pas possible de seuiller simplement le cœfficient de corrélation de la référence avec l'image. En effet il est fréquent qu'un caractère en inclue un autre comme ceci est illustré par la figure 10.

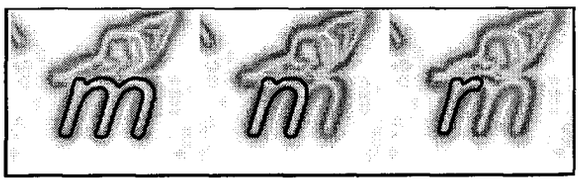


Figure 10. – Un «m» connecté générant des hypothèses «m» «n» et «r».

Dans la configuration précédente, le « m » corrèle aussi bien que le « n » ou le « r ». En fait le « m » pourrait très bien être vu comme un « n » connecté à un environnement particulier. Le choix entre « m » et « n » doit être pris (au moins partiellement) sur des critères de formes : si on conserve les deux hypothèses avec des taux de vraisemblance semblables il n'y aura en général aucune règle syntaxique permettant de lever de manière fiable l'ambiguïté.

La raison qui doit faire choisir le « m » est que le « m » contient le « n » et non l'inverse. Pour quantifier le fait que l'on souhaite privilégier les caractères qui incluent les autres (car ce sont ceux qui expliquent le mieux les contours présents), on décide d'attribuer à chaque hypothèse sélectionnée un taux de confiance défini par la corrélation de chaque référence sur le masque de l'ensemble des références présélectionnées.

6. Traitements syntaxiques

Une fois les chaînes complètes, elles sont décomposées en mots par seuillage adaptatif de l'espacement entre lettres consécutives. Le seuil est calculé, pour chaque chaîne, à partir de l'ensemble des espaces entre lettres consécutives de la chaîne.

Quelques connaissances syntaxiques élémentaires sont ensuite ajoutées pour lever les dernières ambiguïtés. Ce sont par exemple des connaissances sur les chaînes alphanumériques valides, sur les mots de une lettre valides, sur les compositions possibles de majuscules et minuscules dans un même mot...

Ces connaissances élémentaires sont utilisées afin de trouver la meilleure solution syntaxiquement correcte. On sélectionne les connaissances qui peuvent s'exprimer, au moins partiellement, en terme de règles sur les couples de lettres valides et on relance l'algorithme de programmation dynamique en interdisant les couples invalides.

Donnons un exemple de connaissance s'exprimant en terme de couples de lettres valides. La connaissance sur les majuscules et minuscules est : « un mot est formé soit uniquement de majuscules, soit uniquement de minuscules, soit d'une majuscule en première lettre et de minuscules ensuite ». Soit alors $c1$, $c2$ un couple de caractères envisagé :

- si $c1$ et $c2$ sont tous les deux des majuscules ou des minuscules, le couple est accepté (pour cette règle);
- sinon, si $c2$ est le premier caractère d'un mot (ils ont été étiquetés après le découpage), le couple est accepté car $c1$ et $c2$ n'appartiennent pas au même mot;
- sinon, si $c1$ est une majuscule, $c2$ est une minuscule et $c1$ le premier caractère d'un mot, le couple est accepté;
- dans tous les autres cas, le couple est refusé.

Les connaissances syntaxiques, trop complexes pour être implantées sous forme de règles sur les couples de lettres valides, ne sont utilisées que comme vérification. Par exemple, la connaissance sur les chaînes alphanumériques ne peut pas totalement s'exprimer sous forme simple et on vérifie, après la phase de « programmation dynamique syntaxique » que la chaîne est soit entièrement formée de lettres, soit entièrement formée de chiffres, soit un nom de route valide.

7. Résultats

L'ensemble de la méthode a alors été testé sur des coupures de cartes IGN au 1/25000 et 1/50000 (en se limitant aux chaînes horizontales pour le 1/50000).

Par ailleurs un critère de fiabilité a été défini et une chaîne est considérée comme douteuse si un des caractères est douteux du point de vue de la reconnaissance des formes, ou si une règle syntaxique n'est pas respectée, ou si cette chaîne est formée d'un seul mot composé d' une seule lettre et non répertorié.

Les résultats, avec une carte au 1/25000, sont les suivants, avec 1259 chaînes réelles présentes :

- 3 chaînes oubliées, correspondant toutes à des points de cotes entièrement connectés;
- 49 chaînes inventées, dont 41 correspondent à des mots de une lettre;
- 44 chaînes mal interprétées, dont 43 parce qu'un caractère connecté a été inventé;
- 8 chaînes mal découpées; ce sont toutes des chaînes « à disposition ».

Sur les 101 $(49 + 44 + 8)$ erreurs, 95 sont détectées comme douteuses, inversement 108 chaînes sont alertées à tort.

Les figures 11 à 14 montrent quelques exemples d'erreurs et d'alertes. Les chaînes entre parenthèses sont les chaînes alertées par le programme.

La figure 11 montre une chaîne oubliée car entièrement connectée. La figure 12 montre une chaîne alertée car le « *l* » ne respecte pas les règles d'élision. La figure 13 montre une chaîne alertée car le « *o* » (reconnu lors de la phase des caractères connectés) est douteux. Sur la figure 14, l'espace entre les caractères est supérieur au seuil toléré, le programme a donc reconnu des caractères isolés et il les alerte.

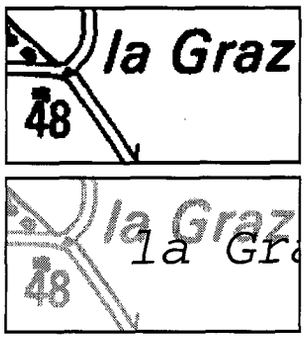


Figure 11. – Un «48» oublié.

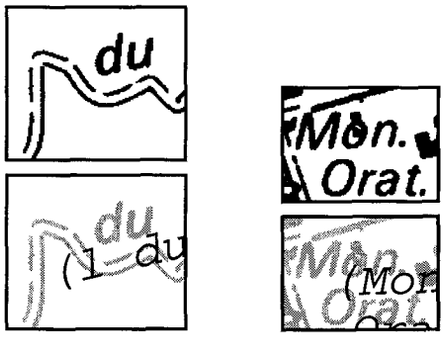


Figure 12 et 13. – Alertes sur des critères syntaxiques (12) et de formes (13).

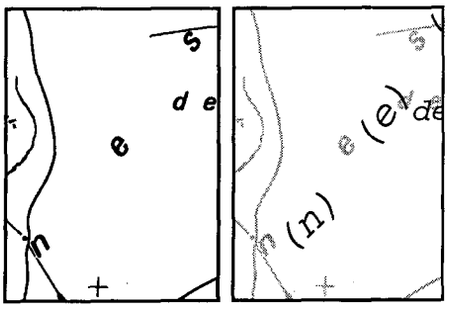


Figure 14. – Alertes sur des caractères isolés.

La figure 15.2 montre, sur la portion de carte correspondant à la figure 15.1, les résultats obtenus.

Globalement, nous estimons que la méthode donne des résultats de qualité suffisante pour les phases de bas niveau qui ont été décrites ici; les possibilités d'amélioration les plus prometteuses se situent au niveau syntaxique et sémantique.

8. Prolongement

Les chiffres qui ont été donnés ne sont pas encore tout à fait suffisants pour une utilisation industrielle. Le taux d'erreurs non alertées correspond à peu près au taux d'erreur des saisies manuelles. Nous espérons améliorer ces résultats :

- en utilisant des règles syntaxiques de plus haut niveau; par exemple base des mots communs les plus courants (il est impossible d'envisager un dictionnaire complet); on effectuera alors une correction automatique quand la suppression d'un caractère douteux permet de transformer un mot inconnu en un mot du dictionnaire partiel;
- en utilisant des règles sémantiques afin d'associer les chaînes aux objets et d'alerter certaines chaînes lorsque l'association est impossible;
- en analysant a posteriori la cohérence géométrique globale des chaînes créées; en effet, les règles utilisées lors du chaînage ne sont appliquées que sur des couples de lettres; ceci a pour conséquence que, par exemple, le seuil sur les différences d'orientation ne permet pas d'exprimer que les chaînes doivent être « régulièrement courbes » (au moins à l'intérieur d'un même mot).

Enfin nous comptons aborder le problème du croisement entre les chaînes extraites et le résultat de l'extraction des autres couches, ceci dans un but de validation et d'enrichissement mutuel. On peut citer à titre d'exemple : les chaînes de caractères expliquant une interruption du réseau routier et validant une hypothèse de reconnexion, les mots-clés (« borne », « pylône », « château »...) déclenchant la recherche du symbole associé ou encore l'attribution automatique des noms de routes à des tronçons du réseau.

Cette création de liens « topologiques » est l'étape indispensable vers l'alimentation automatique d'un SIG.

BIBLIOGRAPHIE

[Belaïd] A. Belaïd, Y. Belaïd, « Reconnaissance des formes, méthodes et applications », InterEdition, Paris, 1992.

[Carrosio] A. Carosio, R. Stengele, « Automatic pattern recognition for economic map revision », *16th International Cartographic Conference*, Cologne , Mai 1993, pp. 518-526.

[Cun] Y. Le Cun, « Handwritten Digit Recognition : Application of Neural Network Chips and Automatic Learning », *IEEE Communication Magazine*, Novembre 1989, pp. 41-46.

[Duda] R.O. Duda, P.E. Hart, « Pattern Classification and Scene Analysis », Willey Inter Science, New York, 1973.

[Fletcher] L.A. Fletcher, R. Katsuri, « A Robust Algorithm for Text String Separation from Mixed Text/Graphic Images », *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 10, n°6, 1988, pp. 910-918.

[Fu] K. S. Fu, « Syntactic Pattern Recognition and Application », Prentice Hall, Englewood Cliffs, New Jersey, 1982.

[Gondran] M. Gondran, M. Minoux, « Graphes et Algorithmes », *Collection de Direction des Etudes et Recherches d'Electricite de France*, Eyrolles, Paris, 1985.

[Han] W.J. Han, « Optical Character Recognition », *The RCA Multifont Reading Machine, G.L. FISHER et al.*, Ed. Spartan book, 1962.

[Lee] S.W. Lee, J.S Park, Y.Y. Tuang, « Performance evaluation of non linear shape normalization method for the recognition of large set handwritten characters », *2th Intenational Conference on Document Analysis and Recognition (ICDAR'93)*, Tsukuba, Octobre 1993, pp. 402-407.

[Lefrère] L. Lefrere, « Contribution au Développement d'Outils pour l'Analyse Automatique de Documents Cartographiques », *Thèse de Doctorat de l'Université de Rouen*, Rouen, Octobre 1993.

[Leplumey] I. Leplumey, J. Camillerap, G. Lorette, « A Robust Detecor for Music Staves », *2th Intenational Conference on Document Analysis and Recognition (ICDAR'93)*, Tsukuba, Octobre 1993, pp. 902-905.

[Likforman] L. Likforman-Sulem, C. Faure, « Une Méthode de Résolution des Conflits d'Alignements pour la Segmentation des Documents Manuscrits », *3ème Colloque National sur l'Ecrit et le Document (CNED 94)*, Rouen, Juillet 1994, pp. 265-272.

[Miclet] L. Miclet, « Méthodes Structurelles pour la Reconnaissance des Formes », *Collection technique et scientifique des Télécommunication*, Eyrolles, Paris 1984.

[Nakamura] A. Nakamura, O. Shiku, M. Anegawa, C. Nakamura, H. Kuroda, « A Method for Recognizing Characters Strings from Maps Using Linguistic Knowledge », *2th Intenational Conference on Document Analysis and Recognition (ICDAR'93)*, Tsukuba, Octobre 1993, pp. 561-564.

[Nieman] H. Nieman, Pattern Analysis and Understanding, *Springer Series in Information Sciences*, Springer–Verlag, Berlin 1989.

[Ogier] J.M. Ogier, R. Mullot, J. Labiche, Y. Lecourtier, « Intégration d'outils bas niveau dans une stratégie d'analyse de document de haut niveau », *9ème congrès Reconnaissance des Formes et Intelligence Artificielle (RFIA'94)*, Paris, Janvier 1994, pp. 533-544.

[Pierrot] M. Pierrot Deseilligny, « Lecture Automatique de Cartes », *Thèse de Doctorat de l'Université de Paris V*, Paris, Octobre 1994.

[Press] W.H. Press, B.P. Flannery, S.A. Teuklosky, W.T. Weterling, « Numerical Recipes in C », Cambridge University Press, Cambridge 1988, pp. 290-332.

[Rocha] J. Rocha, T. Pavlidis, « A Solution to the Problem of Touching and Broken Character », *2th Intenational Conference on Document Analysis and Recognition (ICDAR'93)*, Tsukuba, Octobre 1993, pp. 602-605.

[Simon] J.C. Simon, « La reconnaissance des formes par algorithmes », Masson, Paris, 1985.

[Vaxivière] P. Vaxivière, K. Tombre, « CAD Conversion of Mechanical Drawings », *IEEE Computer Magazine*, Vol. 25, n°7, pp. 46-54.

[Yan] H. Yan, « Color Map Image Segmentation using Optimized Nearest Neigbor Classifier », *2th Intenational Conference on Document Analysis and Recognition (ICDAR'93)*, Tsukuba, Octobre 1993, pp. 111-114.

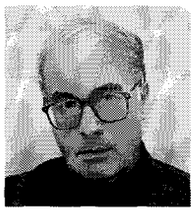
Manuscrit reçu le 1er février 1995.

LES AUTEURS

Pierrot DESEILLIGNY

Marc Pierrot Deseilligny est né le 21/09/62. Il est ancien élève de l'École Polytechnique et de l'École Nationale Supérieure des Techniques Avancées, docteur en informatique de l'université de Paris V. Il a travaillé pendant 6 ans à l'État–majour de l'Armée de Terre sur les problèmes de géographie numérique et la lecture automatique de cartes. Il est responsable depuis un an du projet « cartes scanées » à l'Institut Géographique National.

G. STAMON



G. Stamon a obtenu son diplôme de thèse d'Etat à l'université de Paris VI en 1974. Après avoir été Professeur à l'université de Besançon, il est nommé, en 1986, à l'université René Descartes à Paris. Actuellement, il est Président du Conseil Scientifique de l'UFR Mathématiques et Informatique et dirige un laboratoire de recherche. Auteur d'une centaine de papiers scientifiques, il travaille depuis 25 ans dans le domaine de la Reconnaissanc des Formes et la Vision par ordinateur.

H. LE MEN

H. Le Men est né le 21/09/55. Ancien élève de l'École Polytechnique et de l'École Nationale des Sciences Géographiques, il travaille depuis 1980 à l'Institut Géographique National. Après avoir dirigé le département de télédétection, puis le service de la recherche il est actuellement directeur technique adjoint, chargé des programmes de recherche et de développement.

Figure 15.a.

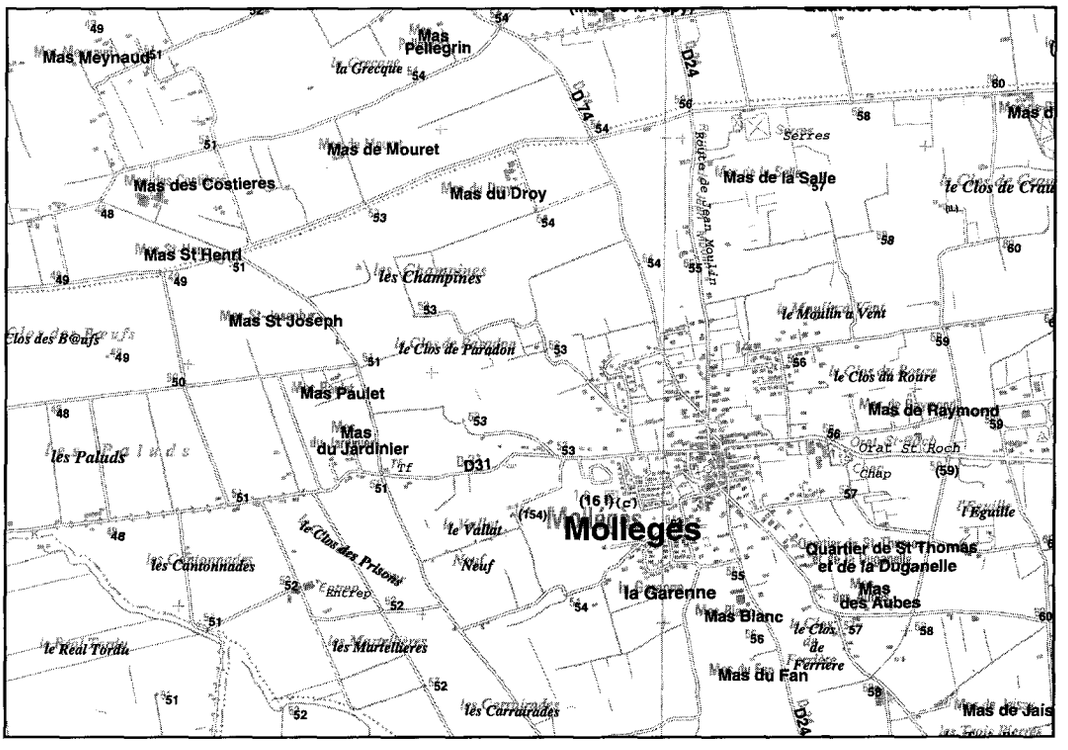


Figure 15.b.