

# Interprétation de cartes géographiques : Algorithmes de reconstruction des réseaux hydrographiques et routiers

## Geographic Map Understanding: Algorithms for Hydrographic and Road Networks Reconstruction

par Robert MARIANI\*, Fabrice LECOURT\*, Marc PIERROT DESEILLIGNY\*, Jacques LABICHE\*\*, Yves LECOURTIER\*\*\*

\* IGN/DT/SR, Laboratoire MATIS 2 av. Pasteur, 94160 Saint-Mandé  
Tel : (1) 43 98 81 28, Fax : (1) 43 98 85 81, Email : mariani@valandre.ign.fr.

\*\* LACP, Université de Caen Faculté des sciences. 14000 Caen

\*\*\* La3i-LACIS, Université de Rouen UFR des sciences et techniques 76821 Mont-Saint-Aignan

### *résumé et mots clés*

L'Institut Géographique National (IGN) a pour objectif de développer sur la carte IGN au 1/25000 un système d'interprétation totalement automatique et complet de la carte. Le but est de convertir automatiquement le fond de cartes existant sous forme papier, en une base de données d'objets géographiques directement manipulables par un SIG. Cet article décrit une méthode générale de haut niveau pour la reconstruction automatique des graphes des réseaux représentés sur les cartes géographiques. Elle a été appliquée aux réseaux hydrographiques et routiers qui sont essentiellement composés de lignes tiretées, de traits pleins interrompus et d'objets surfaciques interrompus. Le formalisme utilisé est celui de la théorie des graphes, qui permet de modéliser naturellement ces réseaux et d'explicitier les contraintes liées à leur reconstruction. Les connaissances a priori sur les réseaux réels et cartographiques sont directement intégrées dans le processus de reconstruction, et traduites soit comme des invariants que doivent vérifier les réseaux en cours de reconstruction, soit comme des mesures de qualité sur les connexions vraisemblables envisagées.

Interprétation de cartes géographiques, Analyse d'image, Modélisation, Arbre recouvrant de poids minimum, Reconstruction de réseau, Découpage polygonal, Théorie des graphes.

### *abstract and key words*

The French Institut Géographique National (IGN) wants to develop an automated map understanding system, for the geographical maps at scale 1/25000. The aim is to automatically convert the 2000 cartographic paper-maps in a geographical objects database, directly usable by a GIS. This paper describes a high-level method for the automated reconstruction of the network graphs, represented on the French geographic maps. This method is applied to the hydrographic and road networks, symbolized with dashed lines, interrupted solid lines and fragmented textured areas. The graph theory paradigm is used, which allows to naturally model those networks by graphs, and to formalize the constraints for their reconstruction. A priori knowledges on natural and cartographic networks are directly used in the reconstruction process, and translated either as invariants the networks must verify during the reconstruction, either as quality criterion for the likely considered connexions.

Geographic map understanding, Image analysis, Model, Minimum weighting spanning subtree, Network reconstruction, Polygonal clipping, Graph theory.

## 1. introduction

Dans le cadre de ses activités de recherche, L'IGN a pour but de développer sur la carte IGN au 1/25000, un système d'interprétation totalement automatique et complet de la carte. L'IGN a entrepris depuis plusieurs années la constitution d'une base de données topographiques, d'un contenu sémantique proche de celui de la carte au 1/25000, mais ayant une précision de l'ordre du mètre. Cette base de données est, aujourd'hui, essentiellement constituée par saisie manuelle à partir d'images aériennes, et ne sera pas achevée avant une quinzaine d'années. L'objectif de ce projet de lecture automatique de cartes, débuté en 1991 [1] est double : 1) fournir un modèle global de la scène pour guider l'interprétation de photos aériennes; 2) fournir des bases de données d'attente, avant que la BDTopo© ne soit achevée. Le but général de cette étude est de convertir automatiquement le fond cartographique, existant sous forme papier, en une base de données d'objets géographiques directement utilisables dans un SIG.

Cet article décrit une méthode générale de haut niveau pour la reconstruction automatique des graphes de réseaux représentés sur les cartes géographiques. Nous l'avons appliquée aux réseaux hydrographiques et routiers. Les cartes analysées sont imprimées à partir de quatre planches correspondant, chacune à une couleur d'impression de la carte : la planche de rouge, de bleu, de vert et de noir. L'IGN étant un producteur de cartes, nous travaillons directement sur ces planches et évitons ainsi le problème de séparation des couleurs qui apparaît lors du scannage en RVB (rouge vert bleu) d'une carte papier [2, 3]. Ce travail, débuté en 1991 [1], a essentiellement porté sur l'extraction des chaînes de caractères, du bâti et du réseau routier, ces trois thèmes étant représentés sur la planche de noir. Le taux de reconnaissance de chaînes de caractères est de l'ordre de l'erreur de saisie manuelle sur une carte « propre », mais suffit largement à notre application car nous travaillons avec un taux d'erreurs maximal de 10 %. L'extraction du réseau routier et du bâti est satisfaisante en plaine et en milieu péri-urbain, mais pose des problèmes en milieu urbain. Une approche différente pour le milieu urbain est en cours d'étude. Finalement, en travaillant avec un taux d'erreurs de 10 %, les cas douteux rencontrés doivent être signalés par le système à l'utilisateur. Actuellement, nous travaillons sur les quatre planches, et les algorithmes exposés ici sont consacrés aux planches de bleu et de rouge.

De nombreux travaux ont été effectués sur l'analyse de documents graphiques, comme par exemple le cadastre [4, 5], les plans d'ingénieurs [6, 7], les partitions musicales [8, 9], les images pétrographiques [10], les diagrammes de structures chimiques [11], les plans de câblage téléphonique [12] et les cartes géographiques [3, 13, 14, 1, 15, 16]. De manière générale, ce qui différencie la carte des autres types de documents, c'est d'une part, une grande densité d'informations et une légende très riche, d'autre part une très forte normalisation des règles de rédaction.

La démarche générale d'interprétation qui est adoptée, consiste à exploiter au maximum les connaissances a priori propre à chaque thème de la carte. Ces connaissances peuvent être liées, soit aux règles cartographiques de représentation des objets sur la carte, soit aux propriétés que vérifient les objets réels représentés. Ainsi, pour le cas des réseaux hydrographiques et routiers, nous utiliserons le fait que les réseaux réels sont généralement connexes (au sens de la théorie des graphes [17]), et que les lignes tiretées, les traits pleins et les surfaces utilisés pour les représenter, sont soumis à des règles strictes de tracé.

La planche que nous traitons ici porte sur la région de Noves. Elle offre une bonne représentation des réseaux hydrographique et routier en plaine. En effet, au niveau du réseau hydrographique, c'est une planche assez complète qui présente bien les différents niveaux hiérarchiques du réseau, depuis les rivières au débit important jusqu'aux cours d'eau temporaires, et ceci dans des proportions équilibrées.

La section suivante présente la méthodologie générale qui a été retenue. L'étape de prétraitements est décrite section 3, les sections 4, 5 et 6 portent sur la reconstruction des réseaux. Les résultats et performances seront exposés section 7. Enfin, dans la dernière section nous conclurons et proposerons des perspectives opérationnelles à ces travaux.

## 2. méthodologie

Le réseau présent sur la surface de la Terre portera le nom de réseau naturel ou réel, alors que le réseau représenté sur la carte géographique s'appellera le réseau cartographique. Cette définition s'applique aux réseaux hydrographique et routier.

Dans cette section, nous décrivons les caractéristiques des réseaux naturels et cartographiques et les conséquences qui en découlent pour la reconstruction. Ensuite, nous situons notre approche par rapport à certains travaux de la littérature, et enfin nous présentons le processus général que nous avons adopté pour réaliser cette opération.

### 2.1. caractéristiques des thèmes hydrographiques et routiers

Pour chaque réseau, nous distinguons deux types de connaissances : celles relatives au réseau réel, et celles relatives à sa représentation graphique. Dans ce qui suit, nous énonçons les règles qui régissent les propriétés générales des réseaux, les exceptions à ces règles et les conséquences pour la reconstruction.

#### 2.1.1. propriétés des réseaux réels

Le graphe du réseau hydrographique réel contient peu de composantes connexes et très peu de cycles (autant que d'îles). En fait,

il peut être assimilé à une forêt, c'est-à-dire un ensemble d'arbres [17]. Il y a peu de composantes connexes par rapport à la taille du réseau et donc le nombre d'arbres est limité. Le graphe du réseau routier est un graphe connexe (il y a très peu d'exceptions) et il contient à grande échelle un nombre important de cycles.

### 2.1.2. exceptions

Il y a deux exceptions majeures à ces règles :

- La présence des canaux, dans le réseau hydrographique, entraîne une apparition fréquente de cycles.
- A cause du fenêtrage lié à la carte papier, nous n'étudions qu'une partie du réseau réel, celle limitée par le rectangle (ou fenêtre) de la carte. Le fenêtrage crée des composantes connexes (et ouvre des cycles).

Ces exceptions affaiblissent les règles topologiques fortes sur les graphes des réseaux réels. Les graphes des réseaux cartographiques contiennent donc peu de composantes connexes et peu de cycles.

### 2.1.3. caractéristiques graphiques des réseaux

La représentation graphique du réseau hydrographique est structurée en trois classes : les lignes tiretées, les traits pleins et les objets surfaciques texturés. Pour chaque classe, il existe des règles sémantiques et graphiques différentes. Ainsi, les lignes tiretées représentent des cours d'eau temporaires, des canaux souterrains, des aqueducs souterrains, des limites de zones etc... Les traits pleins représentent des cours d'eau bordés d'arbres, des aqueducs au sol, des canaux d'alimentation de faible largeur. Enfin, les objets surfaciques texturés symbolisent les fleuves, les rivières, les lacs. Pour le réseau routier, il existe deux textures : l'aplat pour les routes principales et le tramé pour les routes secondaires. Pour chacune d'elles, il existe deux modes de représentation : le trait plein et la ligne tiretée, qui caractérisent le degré de viabilité d'une route.

## 2.2. conséquences pour la reconstruction

Les objets cartographiques étant fréquemment interrompus (lignes tiretées, occlusions par d'autres objets ...), une des étapes importantes de la tâche de reconstruction sera, après vectorisation, de rajouter les arêtes vraisemblables, ceci afin d'obtenir des réseaux cohérents vis-à-vis des connaissances topologiques a priori.

Les règles cartographiques utilisées pour représenter les lignes tiretées, les traits pleins et les objets surfaciques sont différentes. Afin d'exploiter ces connaissances propres à chaque thème, nous avons décomposé le processus de reconstruction du graphe d'un réseau en quatre étapes : 1) transformer les lignes tiretées en traits

pleins; 2) reconnecter les traits pleins entre eux; 3) reconstruire les objets surfaciques fragmentés; 4) connecter les traits pleins aux objets surfaciques. La transformation des lignes tiretées en traits pleins est réalisée en utilisant des règles spécifiques à la planche (intra-planche). La connexion des traits pleins entre eux utilise des règles inter-planches : nous devons, par exemple, prendre en compte les règles de superposition d'objets appartenant à des planches différentes et qui créent des interruptions de tracé des traits pleins. Ainsi, comme le montre les trois images suivantes (cf. figure 1), lorsqu'une route, dessinée sur la planche des noirs, passe au-dessus d'une rivière, le trait plein représentant la rivière sur la planche de bleu est interrompu à leur intersection.

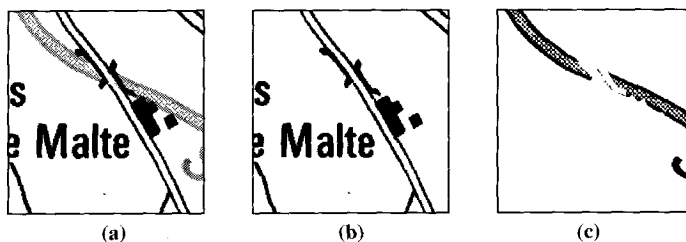


Figure 1. – Une rivière interrompue par une route.  
(a) Une route passant par dessus une rivière; (b) la planche de noirs; (c) la planche de bleu.

Cette analyse menée sur les réseaux réels et sur leur représentation graphique, nous a permis d'obtenir une décomposition du processus général de reconstruction, illustré plus loin.

## 2.3. positionnement par rapport aux autres travaux

Baucoup de travaux sur les documents graphiques ont porté sur la détection d'éléments linéaires, notamment sur les lignes tiretées. Certaines méthodes travaillent directement sur l'image, qu'elle soit en niveaux de gris ou binaire. Dans cette catégorie d'approche, nous pouvons citer la transformation de Hough [18, 19], l'appariement de lignes droites, d'arcs de cercle et de splines [18], et enfin celles qui utilisent des opérateurs directionnels de morphologie mathématique [20, 21].

Des méthodes concernant l'extraction de textures linéaires ont été proposées [22, 23, 24]. Elles offrent un cadre général de travail pour la détection de lignes présentes dans les documents, sans connaissances a priori. Par exemple, [22] analyse une image obtenue par amincissement, en extrayant les caractéristiques sur les courbes et les primitives de texture présentes dans l'image, puis calcule les distributions de ces éléments le long des courbes, et enfin, à l'aide de critères de similarité et de proximité, crée des classes de texture linéaires. Cette méthode est très intéressante du point de vue méthodologique, mais sans parallélisation des opérations, elle est très coûteuse.

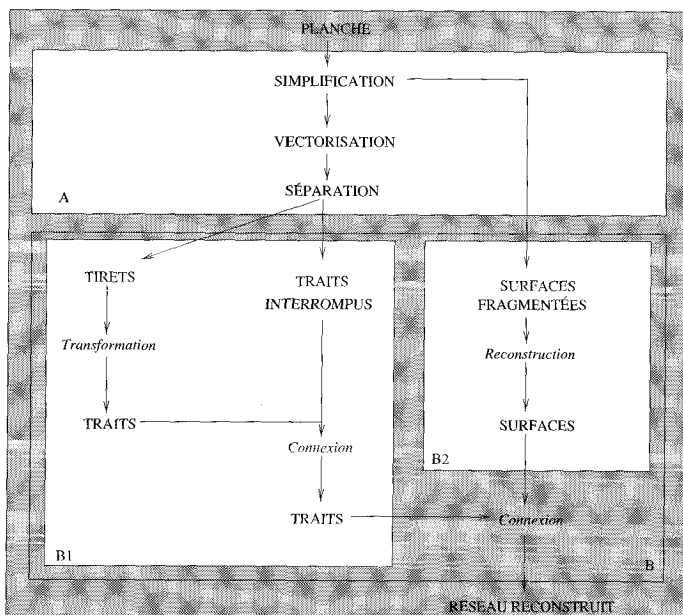
Notre démarche se rapproche de celles qui procèdent par simplification de la carte pour faciliter l'extraction de certains thèmes

[25], et qui utilisent des connaissances a priori sur les objets de la carte afin d'en améliorer la vectorisation [26, 27, 28].

Pour les réseaux, nous manipulerons un graphe, qui détient des propriétés topologiques, utilisées directement comme connaissances a priori dans le processus de reconstruction. Cette modélisation sous forme d'un graphe est naturelle pour ce type de réseau, et les outils de la théorie des graphes [17] offriront un moyen simple d'exprimer des requêtes sur cet objet.

## 2.4. description du processus

Le processus général (cf. figure 2) se décompose en deux grandes étapes : 1) prétraitements de l'image initiale; 2) reconstruction du réseau interrompu. La phase des prétraitements consiste principalement à reconnaître et à supprimer les objets qui ne font pas partie du réseau linéaire (réseau formé des tirets et des traits pleins), à vectoriser l'image restante et à séparer les traits pleins des tirets. Les étapes de la reconstruction ont été décrites au paragraphe (§ 2.2).



**Figure 2. – Processus général de reconstruction de réseau**  
 La partie A correspond à la phase des prétraitements de l'image initiale. La partie B correspond à la reconstruction du réseau. Celle-ci comporte la phase de reconstruction du réseau linéaire (B1) et celle de reconstruction des surfaces (B2)

Dans cet article, nous décrivons d'abord les phases de prétraitements de l'image initiale et de reconstruction du réseau linéaire, générale pour les deux réseaux. Ensuite, nous détaillons la reconstruction des objets fragmentés, opération spécifique au réseau hydrographique. La dernière étape de connexion des traits pleins aux surfaces est en cours de réalisation et sera brièvement abordée.

## 3. prétraitements de l'image initiale

Cette étape se décompose en trois parties : 1) simplification de l'image initiale; 2) vectorisation de l'image résultante par squelettisation et approximation polygonale [1]; 3) séparation des tirets et des traits pleins;

### 3.1. simplification de l'image

La simplification de l'image consiste, d'une part à reconnaître et à supprimer les objets qui n'intéressent pas la reconstruction du réseau, d'autre part à obtenir une représentation standardisée des objets intéressants (par exemple, la transformation des objets tramés du réseau routier en objets aplats pour la vectorisation). Les objets préalablement extraits lors de cette phase sont conservés et seront plus tard utilisés pour l'interprétation globale de la carte. Nous décrivons brièvement cette étape pour les deux planches considérées.

#### 3.1.1. simplification de la planche de bleu

Le but de cette opération est de supprimer tous les objets qui ne font pas partie du réseau linéaire. L'image résultante ne sera composée que de tirets et de traits pleins. Les objets qui perturbent la reconstruction du réseau hydrographique sont les suivants : les caractères (surtout ceux connectés au réseau), les disques connectés aux branches du réseau représentant les arbres qui bordent les ruisseaux, et les objets surfaciques texturés symbolisant les lacs, les rivières, les fleuves etc... Les autres objets qui représentent les marécages sont représentés par des particules connexes qui sont éliminées sur des critères de forme.

Dans les deux paragraphes suivants, nous présentons l'algorithme de marquage de la texture, ainsi que celui qui permet d'extraire les disques connectés. L'extraction des chaînes de caractères, connectées ou non, est décrite dans [29] et ne sera pas abordée ici. De même, la suppression des marécages n'est pas décrite ici.

##### 3.1.1.1. marquage de la texture

L'algorithme qui est présenté ici, est très spécifique à cette texture. Un travail est en cours de réalisation pour avoir une approche générique de traitement des textures utilisées sur les cartes. La bibliographie étant très importante sur le sujet, elle semble démesurée par rapport à l'usage que l'on en fait. [30] détaille une liste importante des méthodes existantes.

La texture des objets surfaciques est une texture fine (cf. figure 3 (a)). Les primitives qui la composent sont très petites et très proches. Ainsi, les méthodes statistiques peuvent donner de bons résultats. Par contre, ces méthodes échouent là où il y a très peu

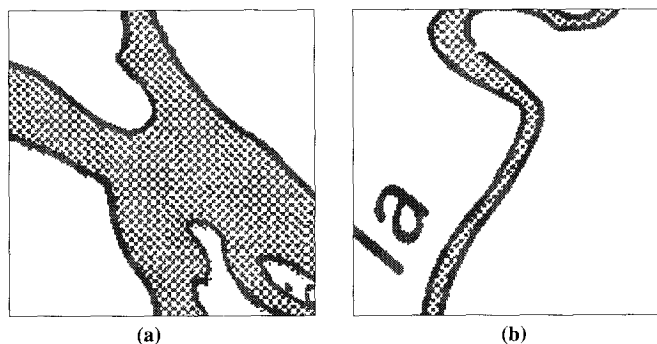


Figure 3. – Exemples d'objets texturés.  
(a) une rivière large; (b) une rivière très fine (zoom 3)

de texture représentée, comme dans les rivières de faible largeur (cf. figure 3 (b)). La méthode que nous avons retenue est une transformation de texture. Dans notre cas, c'est à la fois la plus simple et celle qui fournit les meilleurs résultats.

Après une analyse de l'histogramme des niveaux de gris, on distingue quatre plages de couleurs utilisées pour représenter la texture. L'algorithme commence par appliquer une table de conversion de couleurs sur l'image en niveaux de gris  $F_{i,j}$  en vue d'obtenir une image  $G_{i,j}$  d'étiquettes  $\{e_0, e_1, e_2, e_3\}$ , chacune correspondant à une plage de couleurs. Cette conversion des couleurs est définie ci-après :

$$G_{i,j} = \begin{cases} e_0 : F_{i,j} = 10 \\ e_1 : F_{i,j} \in [40..239] \\ e_2 : F_{i,j} = 240 \\ e_3 : F_{i,j} \in [0..9] \cup [11..39] \cup [241..255] \end{cases} \quad (1)$$

Cette conversion des couleurs n'est pas une méthode stable. En effet, d'une planche scannée à l'autre, les niveaux de gris qui représentent la texture, peuvent être légèrement différents. Comme nous l'avons dit précédemment, des travaux sont en cours pour développer un algorithme général de reconnaissance structurelle de texture fine, appliqué aux cartes géographiques. Cette méthode n'est donc qu'une solution d'attente, qui nous permet de traiter la reconstruction des réseaux, notre principal objectif actuellement.

L'étiquette  $e_0$  permet de coder le bord de la texture, les tirets, les traits pleins, ainsi que les petites primitives qui sont utilisées pour représenter la texture;  $e_1$  code la trame de la texture;  $e_2$  correspond au fond de l'image et aux petites particules qui composent la texture;  $e_3$  correspond au reste de l'image.

Nous ne considérons ensuite que les particules connexes d'étiquettes  $e_0, e_1$  ou  $e_2$ . Pour chaque type de particule  $e_i$ , nous avons défini un seuil  $S_i$  ( $i = 0..2$ ) sur la taille. L'algorithme opère en cinq étapes (cf. figure 4) :

- Attribution de l'étiquette *texture* aux particules connexes étiquetées  $e_i$  de taille  $T_i \leq S_i$ .

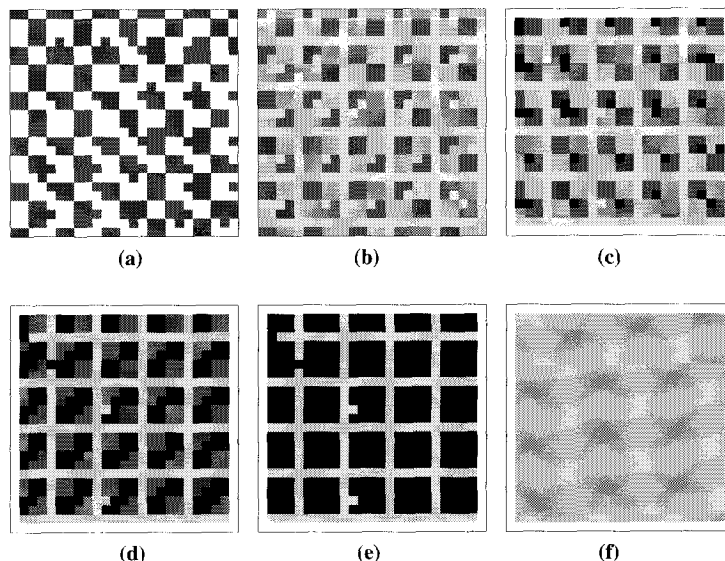


Figure 4. – Étiquetage.

(a) la texture initiale; (b) image des étiquettes  $e_0, e_1$  et  $e_2$ ; (c) attribution de texture aux petites particules  $e_0$  (en noir); (d) idem pour  $e_1$ ; (e) idem pour  $e_2$ ; à l'issue de cette étape, on a isolé la trame; celle-ci à une épaisseur de l'ordre du pixel; (f) disparition de la trame par une dilatation avec une boule de rayon 2. (grossissement de 20 : un carré représente un pixel)

- Étiquetage des pixels de la trame à texture par une dilatation morphologique [31] des pixels étiquetés *texture* sur les pixels étiquetés  $e_1$ , à l'aide d'une boule de rayon 2.

- Élimination des petites composantes connexes étiquetées *texture*.

Ce traitement bas niveau sur la texture du réseau hydrographique est suffisant pour l'application qui est décrite ici, et donne de bons résultats sur les planches traitées. Les images ci-dessous illustrent des résultats du marquage de la texture sur Noves.

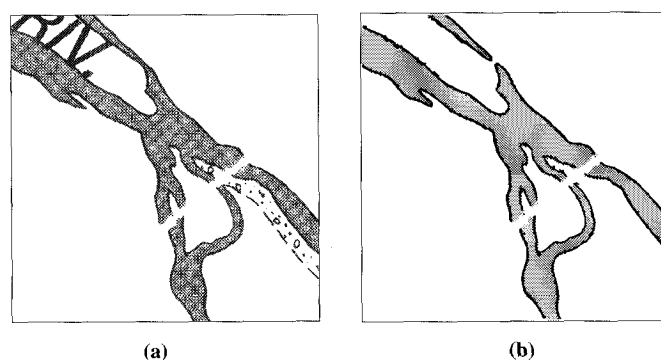


Figure 5. – Détection de texture sur Noves.

(a) image initiale texturée; (b) le résultat du marquage de la texture.

### 3.1.1.2. extraction des disques

L'objectif de cette étape est de reconnaître et d'extraire les disques qui représentent les arbres qui bordent les cours d'eau. Généralement, ces disques sont connectés au réseau hydrographique, et il

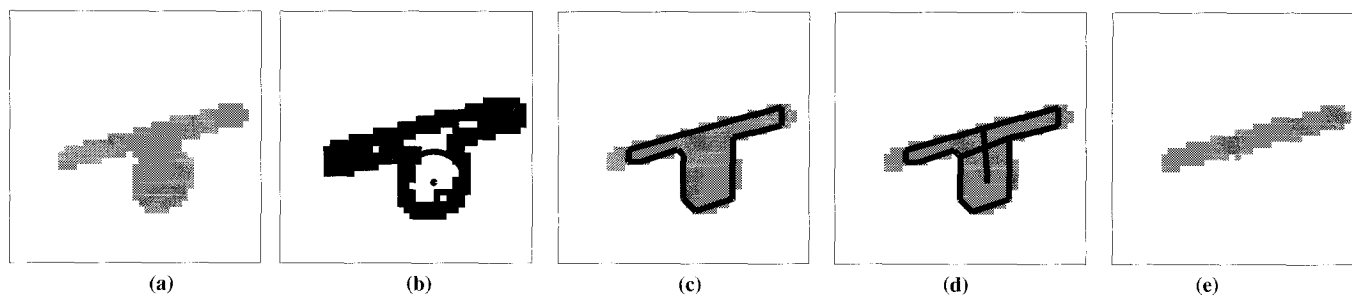


Figure 6. – Extraction des disques connectés. (a) image initiale; (b) image des gradients et localisation du centre; (c) suivi de contour et approximation polygonale; (d) droite passant par les points d'attache; (e) image après effacement (45×45 pixels).

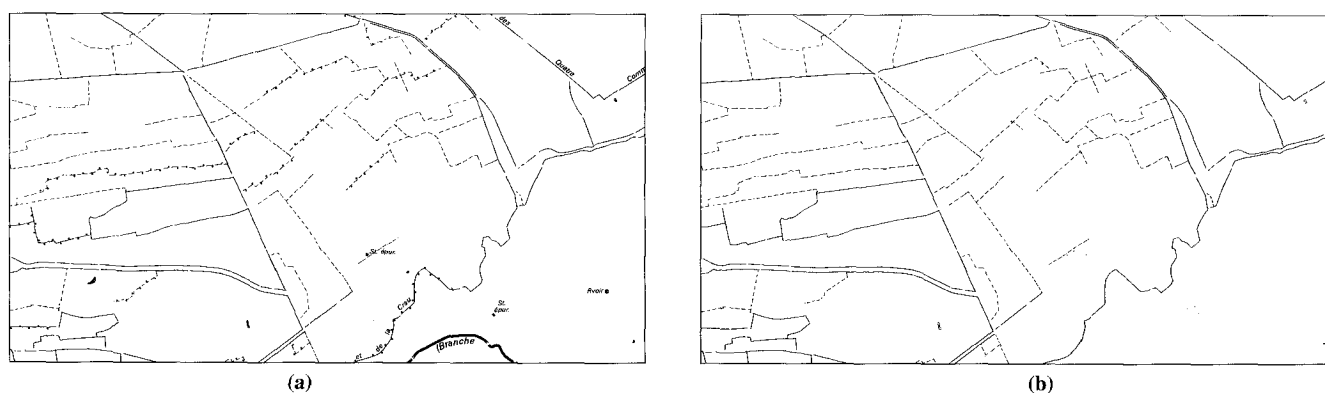


Figure 7. – Simplification de la planche de bleu. (a) image initiale et (b) image simplifiée (4048×2222 pixels).

est souhaitable, après la phase de reconnaissance, de les effacer sans créer d'interruptions supplémentaires au niveau du réseau.

Plusieurs méthodes décrites dans la littérature permettent de détecter fiablement des disques. Nous trouvons, par exemple, la transformée de Hough [19, 12, 32], les méthodes basées sur une estimation de paramètres [33, 34, 35], la morphologie mathématique [36], ainsi que les techniques de corrélation [30, 37].

D'autres méthodes, utilisant une approche descendante, considèrent directement que l'ensemble des disques constitue un attribut du réseau hydrographique, et mèneront simultanément la détection des lignes (ici les tirets, les traits pleins et les contours des surfaces) et des disques présents dans leur voisinage [13, 14]. Notre démarche est différente car elle consiste d'abord à extraire séparément tous les thèmes *élémentaires* de la carte, et ensuite à considérer le problème d'attribution, comme un problème de couplage de thèmes, à l'aide de connaissances a priori. De plus, ces disques sont aussi représentés sur la planche des noirs, d'où la nécessité de pouvoir réutiliser directement cet algorithme.

Pour chaque disque, nous estimons d'abord son centre  $c$  par optimisation d'une fonction de corrélation entre le gradient théorique d'un disque isolé et le gradient observé de l'élément traité (cf. figure 6 (b)). Ensuite, nous travaillons en mode vecteur pour extraire proprement le disque connecté. Pour cela, nous faisons un suivi de contour puis une approximation polygonale sur l'imagette

centrée en  $c$  (cf. figure 6 (c)), et nous obtenons ainsi un graphe, dont les noeuds et les arêtes sont respectivement, les sommet et les segments fournis par l'approximation polygonale. Après une extraction des faces de ce graphe [1], nous ne conservons que celle qui contient  $c$ , que nous appelons  $F$ .

Soit  $y$ , le segment qui joint  $c$  et le milieu de la plus grande arête de  $F$ . Nous recherchons le meilleur segment  $x$ , passant par au moins un point de concavité de  $F$ , qui coupe  $y$  et qui divise  $F$  en deux faces (cf. figure 6 (d)). Le coût associé à  $x$  dépend de l'angle que forment  $x$  et  $y$  et des distances des deux extrémités de  $y$  avec le point d'intersection de  $x$  et  $y$ . La droite portée par  $x$  délimitant deux demi-plans, nous éliminons tous les pixels noirs de la composante connexe qui sont situés dans le même demi-plan que  $c$  (cf. figure 6 (e)).

Sur la planche de noir, la confusion entre un disque connecté et un bâtiment connecté peut être importante. Nous avons alors recours à une phase de classification, qui est inutile sur la planche de bleu.

### 3.1.1.3. résultat de la simplification

Voici des exemples (cf. figure 7) de simplification de la planche des bleu. L'image résultante ne contient que des tirets et des traits pleins interrompus.

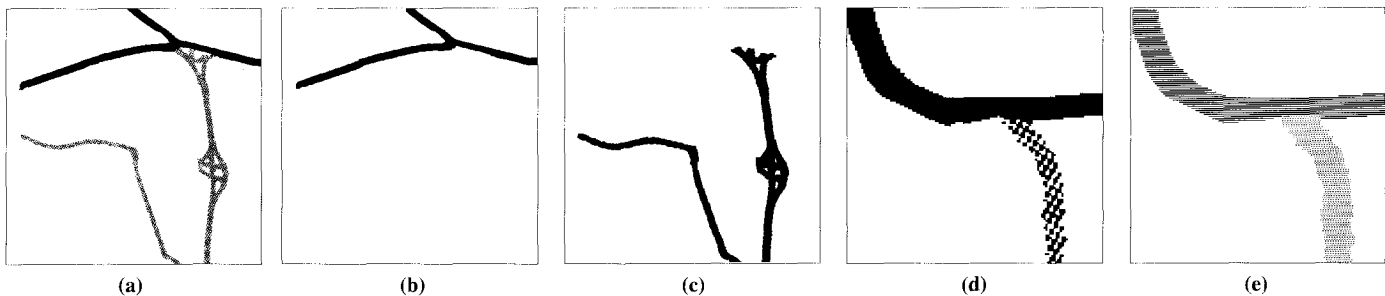


Figure 8. – Simplification de la planche des rouges.

(a) Image initiale; (b) Extraction de l'aplat; (c) Marquage de la trame; (500×500 pixels); (d) Image initiale; (e) Image finale avec marquage de la trame.

### 3.1.2. simplification de la planche des rouges

Afin de pouvoir squelettiser toute l'image, nous transformons la texture en aplat, à l'aide d'outils de la morphologie mathématique [31]. Dans un premier temps, nous extrayons les zones en aplat (cf. figure 8 (b)) par une fermeture avec une boule de rayon 2. Après soustraction de cette image à l'image initiale, il nous reste l'image de la trame. Celle-ci est transformée en aplat par une ouverture avec une boule de rayon 2 (cf. figure 8 (b)). L'image résultante est filtrée, en éliminant les petites composantes connexes noires. A la fin, elle ne contient plus que le réseau routier en aplat.

## 3.2. vectorisation

La vectorisation repose sur une méthode de squelettisation et d'approximation polygonale sur les branches du squelette. La méthode de squelettisation est décrite en détail dans [38]; elle fonctionne selon le principe suivant :

- Construction d'un graphe de lignes de plus grandes pentes et de lignes de crêtes. Il est homotope à la forme.
- Calcul des points d'ancrage (au sens défini dans [39]), selon différents critères.
- Ebarbulage ultime du graphe avec conservation des points d'ancrage.

La méthode d'approximation polygonale se fait par minimisation d'un critère tenant compte simultanément du nombre de points obtenus et de l'écart entre la courbe initiale et la courbe d'approximation. Le calcul de la courbe minimisant le critère s'effectue par plus court chemin [40] sur le graphe dont les noeuds sont les points de la courbe initiale. A l'issue de cette étape, nous obtenons un graphe non orienté.

Ce graphe initial est obtenu de la façon suivante : à chaque point  $i$  de l'approximation polygonale correspond un sommet  $i$  du graphe et à chaque segment  $[i, j]$  correspond un arête  $(i, j)$  du graphe.

Voici quelques définitions que nous utiliserons par la suite. Nous appelons **degré** d'un sommet  $i$ , le nombre de successeurs de  $i$ . Ainsi, un sommet extrémité aura un degré égal à 1, un carrefour

un degré supérieur à 2 et un sommet intermédiaire un degré égal à 2. Un **brin** est une chaîne [17] reliant deux sommets de degré différent de 2. Soit  $G = (X, U)$  un graphe où  $X$  est l'ensemble des sommets et  $U$  l'ensemble des arêtes.  $G' = (X', U')$  est un **sous graphe** de  $G$  si et seulement si  $X' \subseteq X$  et  $U' \subseteq U$ .

L'essentiel des étapes suivantes sera, à partir du graphe initial obtenu après la vectorisation, de reconstruire un graphe connexe et correct d'un point de vue géométrique et topologique.

## 3.3. séparation

Cette étape est justifiée par les raisons suivantes : 1) les tirets et les traits pleins constituent deux thèmes cartographiques différents et dans un contexte d'interprétation de cartes géographiques, il faut les distinguer; 2) cette séparation permet de simplifier le processus; 3) les règles cartographiques qui leur sont spécifiques, seront directement utilisées pour leur connexion. Nous considérons ici, l'image simplifiée (cf. figure 9 (a)). Elle est composée exclusivement de tirets et de traits pleins. Cette image a été vectorisée (cf. § 3.2) et nous avons obtenu un graphe non orienté.

Formellement, nous définissons un tiret comme un brin (cf. § 3.2) de longueur totale inférieure à un seuil donné  $S$ . En appliquant ce prédicat à chaque brin du graphe initial, nous obtenons le sous graphe des tirets et le sous graphe des traits pleins (cf. figure 9 (b) et (c)).

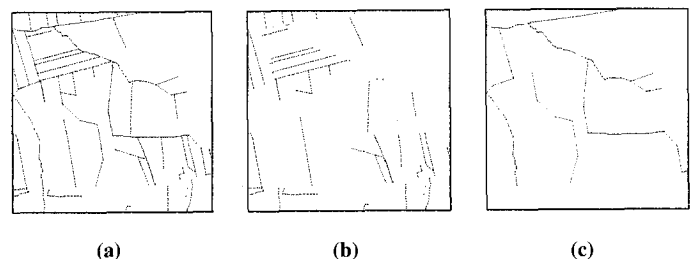


Figure 9. – Séparation des tirets et des traits pleins.

(a) graphe initial; (b) sous graphe des tirets; (c) sous graphe des traits pleins. (500×500 pixels).

Afin de n'oublier aucun tiret, le seuil  $S$  est assez grand, et nous sommes donc confrontés à une sur-détection. Les fausses détections, dues à des traits pleins de petite longueur, sont en nombre très limité (moins de 1%), et ne sont pas gênantes pour la raison suivante :

- Elles ne respecteront pas les règles cartographiques propres au tracé des tirets; or la transformation des tirets en traits pleins est, comme nous le verrons, très stricte quant à l'application de ces règles.

Dans les sections suivantes, nous décrivons les processus de transformation des lignes tiretées en traits pleins, et ensuite, de reconnexion des traits pleins entre eux.

## 4. transformation des lignes tiretées en traits pleins

Nous présentons les connaissances spécifiques au tracé des lignes tiretées ainsi que la fonction qui intègre les connaissances cartographiques. Cette dernière permet de donner une mesure de qualité aux connexions à rajouter. Ensuite, nous décrivons le formalisme utilisé et nous donnons une mesure de complexité de l'algorithme choisi. Pour finir, nous illustrerons cet algorithme sur un exemple.

### 4.1. connaissances spécifiques sur les lignes tiretées

De façon générale, nous distinguons deux règles cartographiques concernant les lignes tiretées :

- L'espace entre deux tirets ne varie pas beaucoup.
- Il n'y a pas de grande courbure entre deux tirets successifs.

Par défaut, le réseau est rectiligne, donc la courbure est une information importante. Aussi, pour éviter toute ambiguïté de lecture, le cartographe représentera très souvent de façon explicite, les courbures présentes dans le réseau réel. Ainsi, pour connecter deux tirets, deux paramètres sont importants : la distance séparant les deux tirets ainsi que leur tangence. Seuls sont spécifiques, pour chaque réseau, les intervalles de valeurs utilisés pour ces deux paramètres.

La fonction de poids définie ci-dessous représente le codage de cette connaissance a priori sur les tirets. Le but est de donner une mesure de qualité à une connexion envisagée.

### 4.2. qualité d'une connexion

Soient  $B$  et  $C$  deux sommets de degré 1 du graphe,  $A$  (resp.  $D$ ) l'unique voisin de  $B$  (resp.  $C$ ) et  $\alpha$  (resp.  $\beta$ ) l'angle entre le vecteur  $\overrightarrow{BC}$  et  $\overrightarrow{CD}$  (resp.  $\overrightarrow{BC}$  et  $\overrightarrow{AB}$ ).

La valeur de la fonction est petite si les deux critères suivants sont vérifiés : 1) la distance entre  $B$  et  $C$  est petite; 2)  $\alpha$  et  $\beta$  sont proches de 0. Nous écrivons  $w(B, C)$  la fonction de poids de  $BC$ . Soit  $d_{BC}$  la distance euclidienne entre  $B$  et  $C$ . Nous avons choisi :

$$w(B, C) = d_{BC} * (1 + \mu((1 - \cos \alpha) + (1 - \cos \beta))), \quad (2)$$

$$\mu > 0, \quad |\alpha| + |\beta| < \delta$$

où  $\delta$  est une constante permettant de rejeter les connexions dont les angles  $\alpha$  et  $\beta$  n'appartiennent pas à l'ensemble  $\{(\alpha, \beta) / |\alpha| + |\beta| < \delta\}$ , et  $\mu$  une constante utilisée pour pondérer le critère de tangence par rapport au critère de distance.

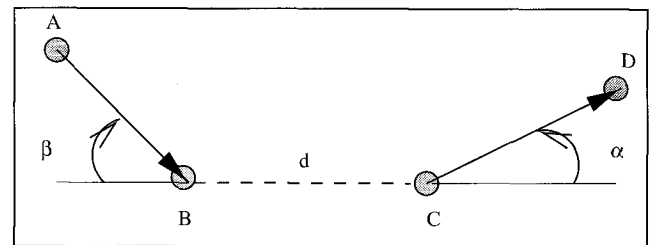


Figure 10. - Paramètres de la fonction de poids.

### 4.3. algorithme de reconstruction : formalisme et complexité

La transformation des lignes tiretées en traits pleins est modélisée comme un problème d'arbre recouvrant de poids minimum (ARPM) d'un graphe non orienté [41]. L'algorithme traite ce problème en tenant compte de la contrainte suivante, qui permet de conserver la planarité du graphe :

#### Contrainte de reconstruction

*Une arête n'est rajoutée dans le graphe que si elle ne coupe aucune arête de celui-ci*

Nous rappelons brièvement les fondements théoriques de l'ARPM. Soit  $G = (X, U)$  un graphe connexe (s'il ne l'est pas, cette définition s'applique à chaque composante connexe du graphe), où chaque arc  $u \in U$  est pondéré par  $w(u)$ . Etant donné  $G' = (X, U')$  un sous graphe de  $G$ . On appelle *poids de  $G'$*  le nombre  $w(G') = \sum_{u \in U'} w(u)$ . Le problème de l'ARPM est de

rechercher un arbre  $H$  tel que  $w(H) = \min_{H' \in \Omega} w(H')$  où  $\Omega$  est l'ensemble de tous les arbres de  $G$  possibles.



La solution du problème de l'ARPM d'un graphe ne dépend que du classement des arcs dans l'ordre des poids croissants. Il existe deux types d'algorithmes : celui de Kruskal et celui de Prim [41]. L'algorithme choisi ici est celui de Kruskal, dont voici un exposé rapide :

#### 4.3.1. algorithme de Kruskal

- Le graphe est donné par la liste de ses arcs  $\{u_1, \dots, u_n\}$ , triée dans l'ordre des poids croissants.
- Au début  $H = \{u_1\}$ . A l'étape  $k$ , si  $u_k$  ne forme pas de cycle, alors  $u_k$  est ajouté à  $H$ .
- On passe à l'arc suivant.

Dans notre application, nous partons d'un graphe initial existant  $G = (X, U)$ , et d'une liste d'arcs valides pondérés  $L$ , préalablement calculée et triée dans l'ordre des poids croissants. Un arc est valide s'il ne coupe pas  $G$  et si ses deux sommets sont de degré 1 (sommets extrémités). Il est pondéré par une fonction de poids (cf. § 4.2). Ainsi, un arc de  $L$  devra vérifier deux conditions pour être rajouté à  $G$  : 1) il ne crée pas de cycle dans  $G$ ; 2) il conserve la planarité de  $G$ .

#### 4.3.2. complexité

D'un point de vue strictement algorithmique, il y a deux structures de données associées aux graphes afin d'accélérer les algorithmes : 1) un tas (file de priorité ou heap en anglais) pour l'algorithme sur les graphes; 2) un quadtree pour les requêtes géométriques. Le quadtree permet un découpage récursif du plan sur lequel est superposé le graphe. Ce plan est représenté par la planche complète. Le découpage récursif d'une case d'un quadtree calqué sur un graphe, se fait jusqu'à ce que le nombre de sommets du graphe soit inférieur à un certain seuil  $Q$  et jusqu'à ce que la taille d'une case soit inférieure à un autre seuil  $T$ . Ainsi, le réglage des paramètres  $Q$  et  $T$  est important. Dans notre cas, nous choisissons en général  $Q = 10$  et  $T = 300$ .

La complexité de cet algorithme est la suivante : en reprenant les définitions de  $G$  et  $L$  on pose :  $N = |X|$  et  $M = |L|$ . Ainsi, pour chaque arc  $(i, j)$  de  $L$ ,  $i \in X$  et  $j \in X$ , on effectue :

- Un test de création de cycle; ceci est réalisé avec l'algorithme de plus court chemin entre les sommets  $i$  et  $j$ . La complexité (avec un tas) est connue et est de  $O(m \log N)$  [40],  $m$  étant le nombre de sommets développés dans l'arbre de plus court chemin entre  $i$  et  $j$ .
- Un test d'intersection de cet arc avec le graphe; cette opération est en  $O(\alpha \log N)$ , en supposant que le quadtree est bien équilibré; ici  $\alpha$  représente la longueur de l'arc courant, et signifie que le nombre de cases du quadtree à examiner est au plus de  $\alpha$ . Avec les paramètres  $T$  et  $Q$  fixés convenablement, on peut obtenir une complexité en  $O(\log N)$  (en général, le découpage est réalisé avec des cases de taille plus grandes que la longueur d'un arc, et un arc sera à cheval sur au plus 2 cases).

- Un ajout d'arc si les deux tests précédents sont vérifiés; ceci est en  $O(1)$ .

La complexité totale est donc majorée par  $O(MN \log N) + O(\lambda M \log N) + O(M)$ ,  $\lambda$  étant la longueur maximale d'un arc envisagé.

## 4.4. résultat

A l'issue de cette étape, les lignes tiretées ont été transformées en traits pleins. Ces derniers sont rajoutés au graphe des traits pleins initiaux (cf. figure 11 à droite). Nous allons maintenant connecter les traits pleins entre eux.

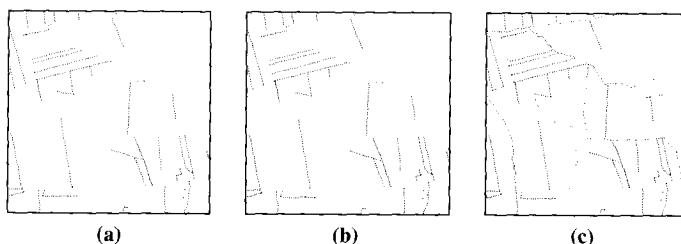


Figure 11. – Transformation des lignes tiretées en traits pleins. (a) le sous graphe des tirets; (b) sous graphe des tirets reconnecteds; (c) graphe des traits pleins. (500×500 pixels)

## 5. connexion des traits pleins entre eux

Nous présentons tout d'abord les connaissances spécifiques au tracé des traits pleins, puis la fonction qui traduit ces connaissances spécifiques sous forme d'une mesure de qualité d'une connexion. Ensuite, nous exposons un algorithme de visibilité d'une arête qui permettra de maintenir, de façon incrémentale, la planarité du graphe. Enfin, nous décrivons les formalismes et complexités des algorithmes de connexion des traits pleins entre eux.

### 5.1. connaissances spécifiques sur les traits pleins

Les interruptions des traits pleins sont dues à la présence, au même endroit, d'autres objets de la carte. Ces objets peuvent appartenir à la même planche ou bien à d'autres planches. Par exemple, une route peut partiellement recouvrir une rivière (cf. figure 1). Les interruptions ont une taille déterminée par les objets qui créent les occlusions. Contrairement aux tirets, où la courbure est une information très importante pour la connexion, ici elle n'a pas beaucoup d'importance pour la raison suivante :

- Deux traits pleins qui doivent être connectés, peuvent ne pas être alignés. Il est même fréquent qu'ils soient presque perpendiculaires.

Ainsi, une connexion entre deux traits pleins constitue souvent le prolongement d'un de ces deux traits.

Les carrefours sont toujours représentés sur le réseau routier, par contre ce n'est pas toujours le cas pour le réseau hydrographique. La reconstruction devra donc prendre en compte la reconstitution des carrefours.

### 5.2. qualité d'une connexion

Nous voulons une fonction qui privilégie les connexions par prolongement de traits pleins. Les traits pleins étant souvent rectilignes, il est en effet plus naturel d'utiliser le prolongement plutôt que la distance la plus courte (cf. figure 12). Soit  $a$ , un point situé sur une arête candidate à la connexion,  $d_{Ba}$  la distance euclidienne entre  $B$  et  $a$ . Soit  $\alpha$  l'angle que fait le vecteur  $\vec{Ba}$  avec la droite  $(AB)$ .

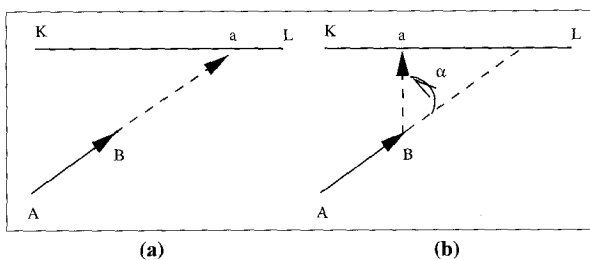


Figure 12. – Prolongement d'un trait plein. (a) préférence pour le prolongement plutôt que pour la distance la plus courte (b).

Nous avons  $W_a$  la fonction de poids associée au point  $a$  :

$$W_a = d_{Ba}^2 \cdot (1 + \tau \cdot \tan |\alpha|), \alpha \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right), \tau > 0 \quad (3)$$

La constante  $\tau$  permet de pondérer le critère de tangence par rapport au critère de distance. Soit  $[KL]$  l'arête candidate à la connexion. L'algorithme calcule le point  $a = a_{(\lambda_0)}$  de  $[KL]$  tel que :

$$W_{a_{(\lambda_0)}} = \min_{\lambda=0..1} \{W_{a_{(\lambda)}} / a_{(\lambda)} = \lambda \cdot K + (1 - \lambda) \cdot L\} \quad (4)$$

avec un pas de variation de  $\lambda$  égal à  $1/KL$ .

### 5.3. algorithme de visibilité pour une préservation incrémentale de la topologie

Le problème lié à la topologie, c'est-à-dire celui de la conservation de la planarité du graphe, est traité au fur et à mesure de la

reconstruction grâce à un algorithme de découpage polygonal [42], dont la tâche est d'en préserver la cohérence.

Nous appelons une **extrémité** un sommet de degré 1 du graphe et une **arête source** une arête ayant au moins une extrémité. Une connexion possible est calculée à partir d'une arête source; elle vérifie la contrainte de reconstruction (cf. § 4.3) et lie un sommet à une arête du graphe.

#### 5.3.1. algorithme

L'algorithme utilisé pour sélectionner les connexions possibles partant d'une arête, appelée arête source, procède en deux étapes (cf. figure 13) : 1) présélection rapide des arêtes géométriquement voisines de l'arête source; 2) calcul précis des connexions envisagées tout en respectant la topologie. Ceci est réalisé à l'aide d'un algorithme de visibilité (découpage polygonal) : nous ne conservons que les morceaux d'arêtes visibles depuis B en regardant dans la direction du vecteur  $\vec{AB}$ . Le domaine de visibilité de B est caractérisé par une demi-ellipse, orientée selon la direction de  $\vec{AB}$  et centrée en B. Les longueurs des deux rayons principaux de l'ellipse sont les longueurs maximales de la connexion autorisée dans la direction de l'arête source et dans une direction perpendiculaire à celle-ci.

Pour chaque arête source  $[AB]$ , nous calculons le découpage polygonal : pour chaque arête  $[JK]$  présente dans la demi-ellipse, nous considérons la surface qu'elle dissimule (cf. figure 13 (IJKL)), et nous découpons les autres arêtes avec cette surface. Tout ce qui est présent dans cette zone (IJKL) est invisible pour l'extrémité B de l'arête source. A la fin, nous ne conservons que les morceaux d'arêtes visibles. Enfin, l'ordre dans lequel les arêtes présentes dans la demi-ellipse sont considérées, n'a pas d'importance. Cependant, après avoir considéré une arête, la liste des arêtes restantes à explorer est mise à jour, en remplaçant chaque arête par ses portions visibles.

Le découpage d'une arête par une demi-ellipse se fait, d'abord en calculant la portion d'arête situé dans le « bon » demi-plan

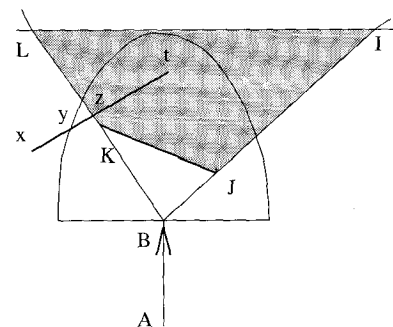


Figure 13. – Découpage elliptique et polygonal.

Le vecteur  $\vec{AB}$  représente une arête source dont l'extrémité est le point B.  $[xt]$  est d'abord découpé par l'ellipse. Seul  $[yt]$  est conservé,  $[xy]$  étant à l'extérieur de l'ellipse. Ensuite,  $[yt]$  est découpé par le polygone (IJKL), (représentant la zone cachée par  $[JK]$ , invisible pour B) et nous ne conservons que  $[yz]$ . En effet, il est impossible de connecter B à  $[zt]$ , car il faudrait passer au-dessus de  $[JK]$  (violation de topologie). Or, ce graphe du réseau linéaire est planaire.

(celui délimité par le petit axe de l'ellipse et qui ne contient pas A), puis en découpant cette portion d'arête avec l'ellipse. Ces calculs sont réalisés de façon analytique. Lorsque l'ensemble des portions d'arêtes présentes dans la demi-ellipse est établi, nous calculons alors l'ensemble des arêtes visibles. Le découpage d'un segment par un polygone est un algorithme classique utilisé en synthèse d'images [42], notamment lorsque le polygone en question représente la fenêtre d'affichage. Voici comment il fonctionne.

### 5.3.2. découpage d'un segment par un polygone convexe

Soit (IJKL) un polygone. La notion de segment visible dans l'algorithme décrit ci-après, correspond au cas classique où le polygone (IJKL) est vu comme une fenêtre d'affichage graphique. Il faut noter que l'algorithme décrit ci-après, ne fonctionne que si (IJKL) est convexe, ce qui sera toujours vrai ici.

Soit  $\gamma(p, D)$ , le signe de la valeur que l'on obtient en injectant le point  $p$  dans l'équation de la droite  $D$ . Une droite  $D$  partage le plan en deux demi-plans : un demi-plan positif et un demi-plan négatif. Une première opération sera de déterminer pour chaque droite  $D$  de (IJKL), le signe du demi-plan qui indique que l'on se trouve du « bon » côté de  $D$ . Nous notons le signe du « bon » demi-plan de  $D$   $\gamma_D^+$  et le signe du « mauvais » demi-plan de  $D$   $\gamma_D^-$ . Nous appelons  $E = \{(IJ)(JK)(KL)(LI)\}$ , l'ensemble des droites portées par les arêtes du polygone (IJKL).

Un point  $p$  se situe du « bon » côté de  $D$  si  $\gamma(p, D) = \gamma_D^+$ .  $p$  se trouve dans le quadrilatère défini par  $E$  si  $\forall D \in E : \gamma(p, D) = \gamma_D^+$ . Le segment  $[a,b]$  est totalement invisible dans  $E$  si  $\exists D \in E : \gamma(a, D) = \gamma_D^- \wedge \gamma(b, D) = \gamma_D^-$ , et  $[a,b]$  est totalement visible dans  $E$  si  $\forall D \in E : \gamma(a, D) = \gamma_D^+ \wedge \gamma(b, D) = \gamma_D^+$ .

Avec ces définitions, les lignes suivantes présentent l'algorithme en question. A la fin, la portion de segment visible (éventuellement vide) sera renvoyée.

- (1) si  $E$  est vide retourner ()
- (2) si  $[a,b]$  est totalement visible dans  $E$  retourner  $[a,b]$
- (3) si  $[a,b]$  est totalement invisible dans  $E$  retourner ()
- (4) si  $\gamma(a, D) \neq \gamma(b, D)$ , avec  $D$  la première droite de  $E$ , alors
  - Calcul du point d'intersection  $x$ , de  $[a,b]$  avec la droite  $D$  courante
  - si  $\gamma(a, D) = \gamma_D^+$  alors  $b \leftarrow x$  sinon  $a \leftarrow x$
  - $E \leftarrow E - D$
  - aller en (1)

Les signes des « bons » demi-plans des droites de (IJKL) sont calculés ainsi (sans traiter ici les cas dégénérés) :

$$\begin{aligned} \gamma_{(IJ)}^+ &= \gamma(K, (IJ)), & \gamma_{(JK)}^+ &= \gamma(L, (JK)), \\ \gamma_{(KL)}^+ &= \gamma(I, (KL)), & \gamma_{(LI)}^+ &= \gamma(J, (LI)). \end{aligned}$$

Le schéma ci-dessous illustre les trois cas de figure qui peuvent se rencontrer :

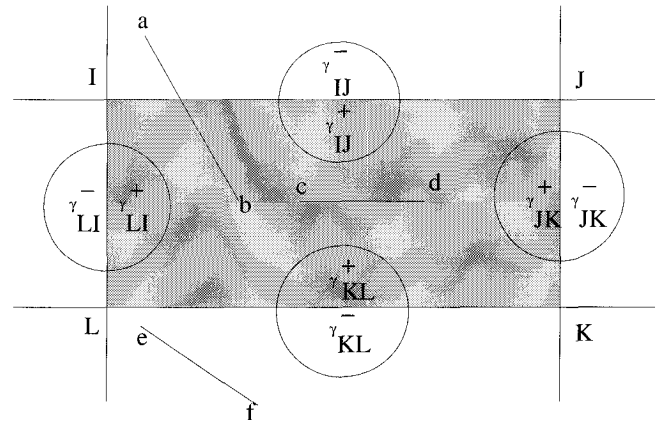


Figure 14. – Découpage de segments par (IJKL).  
Le segment  $[a,b]$  est partiellement dans (IJKL), et on a  $\gamma(a, (IJ)) \neq \gamma(b, (IJ))$  et  $\gamma(b, (IJ)) = \gamma_{(IJ)}^+$ , on découpe donc  $[a,b]$  par (IJ) et on obtient  $[x,b]$ , qui maintenant devient totalement visible dans  $\{(JK)(KL)(LI)\}$ . Le segment  $[c,d]$  est totalement visible dans  $E$ , alors que  $[e,f]$  est totalement invisible.

**Remarque :** Dans notre problème de visibilité, les portions d'arêtes visibles par un point, sont celles situées dans la demi-ellipse et qui ne sont pas dans (IJKL).

### 5.3.3. complexité

La complexité de cet algorithme se calcule de façon analogue au paragraphe (§ 4.3). Elle se résume en 4 calculs d'intersection pour une arête potentiellement visible par l'arête source. Avec un quadtree, nous sommes donc en  $O(4\alpha \log N)$ ,  $N$  étant le nombre d'arêtes présentes dans le quadtree, et  $\alpha$  la longueur de l'arête à découper.

## 5.4. algorithmes de reconstruction : formalismes et complexités

Ici encore, nous utilisons une variante de l'ARPM dont les noeuds de l'arbre sont les composantes connexes du graphe des traits. Nous lui associons une fonction de poids adaptée à chaque planche et élaborée grâce aux règles cartographiques spécifiques aux traits pleins. La connexion des traits pleins se termine par une phase de fermeture des cycles.

Suite aux exceptions précisées au paragraphe 6 2.1.2, nous ne pouvons pas utiliser directement la propriété de connexité globale des deux réseaux. Aussi, nous lui substituons une propriété, dite de connexité locale, définie de la façon suivante : « Si la distance géométrique entre deux points du réseau est petite, la longueur du plus court chemin entre ces deux points ne doit pas être grande ». Cette propriété est utilisée lors de la fermeture des cycles, car concrètement elle signifie que si un cycle est presque totalement représenté, il doit être fermé.

Au fur et à mesure de la reconstruction, l'algorithme considère qu'une connexion à rajouter est valide si elle respecte la contrainte

topologique de reconstruction définie précédemment (cf. § 4.3). Enfin, les connexions qui fermeront les cycles ne devront pas vérifier la propriété de connexité locale.

Nous présentons dans la suite, les formalismes et complexités des algorithmes de l'ARPM et de la connexité locale. L'algorithme de découpage polygonal, ainsi que la fonction de poids sont des paramètres de ces algorithmes et ont été décrits plus haut.

### 5.4.1. arbre recouvrant de poids minimum

Alors que pour les tirets, nous avons utilisé l'algorithme de Kruskal (cf. 6.4.3), ici nous utilisons l'algorithme de Prim [41]. Le choix s'est porté sur ce deuxième algorithme, car le graphe obtenu à l'étape précédente, contient peu de composantes connexes, mais par contre contient beaucoup d'arcs et de sommets. En effet, les interruptions des traits pleins sont rares par rapport à la taille du réseau.

Il est adapté ici, car il utilise une numérotation des composantes connexes, qu'il remet à jour à chaque ajout d'arc. Deux composantes connexes reliées par un arc n'en formeront plus qu'une qui aura un nouveau numéro. Savoir si un arc créera un cycle, est une opération très rapide car elle se résume à tester si les deux composantes connexes ont le même numéro.

Voici comment il s'énonce brièvement dans le cas général :

#### 5.4.1.1. algorithme de Prim

Soit  $G = (X, U)$ , le graphe en cours de construction. Il est supposé être de poids minimum, et ses composantes connexes sont numérotées, chaque sommet d'une composante portant le même numéro. Soit  $L = \{u_1, \dots, u_n\}$  la liste triée des arcs hypothétiques à rajouter, et  $\pi(i)$ , le numéro courant du sommet  $i$ .

On itère sur  $L$ , jusqu'à l'obtention de l'arbre de poids minimum.

- L'arc  $u_i = (a, b)$  est ajouté à  $G$  si  $\pi(a) \neq \pi(b)$ .
- Mise à jour de  $\pi$  : nouvelle numérotation de la nouvelle composante connexe obtenue.
- On passe à l'arc suivant dans  $L$ .

#### 5.4.1.2. adaptation à notre cas

Le graphe non orienté est modélisé par une liste de composantes connexes numérotées. Une composante connexe est constituée d'une liste d'arêtes; chaque arête, obtenue par vectorisation, est caractérisée par deux sommets. Nous appelons une **connexion valide**, une arête qui ne crée pas de cycles, qui respecte la contrainte topologique, et dont au moins un de ses sommets est de degré 1. Dans notre application, nous partons d'un graphe initial existant  $G = (X, U)$ , et d'une liste de connexions valides pondérées (§ 5.2)  $L$  préalablement calculée et triée dans l'ordre

des poids croissants. Ainsi, une arête de  $L$  devra vérifier deux conditions pour être rajoutée à  $G$  : 1) elle ne crée pas de cycle dans  $G$ ; 2) elle conserve la planarité de  $G$ .

Contrairement au réseau routier, les carrefours du réseau hydrographique ne sont pas toujours représentés. Une opération importante lors de la reconstruction sera de créer des carrefours et donc des sommets intermédiaires. Nous avons donc modifié l'algorithme précédent afin de considérer ce cas : après l'ajout d'une connexion valide dans le graphe, il faut envisager des connexions possibles entre les autres arêtes sources et cette dernière arête rajoutée, et remettre à jour la liste des connexions possibles.

#### 5.4.1.3. complexité

De façon analogue au paragraphe (§ 4.3) nous donnons une mesure de la complexité de cet algorithme. Etant donné  $G = (X, U)$  le graphe initial et  $G' = (X', U')$  le graphe final. Soient  $N = |X|$ ,  $M = |U|$ ,  $N' = |X'|$  et  $M' = |U'|$ , et  $R$  le nombre de composantes connexes de  $G$ . La complexité est majorée par la somme des complexités élémentaires suivantes :

- La complexité de l'algorithme de Prim est majorée par  $O((M' - M) \log N')$  (utilisation d'un « tas »)
- la visibilité est majorée par  $O(M' \log N')$  (cf. 6.5.3.3).
- Dans le cas du réseau hydrographique, un appel à la visibilité est effectué pour la création de carrefours :  $O((M' - M) \log N')$ .
- Pour les renumérotation :  $O(R)$  (gestion d'un dictionnaire).
- Pour l'ajout des arêtes et la création des sommets intermédiaires :  $O(4(M' - M)) + O(N' - N)$  : 1 suppression de l'arête, 1 création du sommet intermédiaire et création des 3 arcs qui se rattachent à ce dernier (création d'un carrefour en T).

La complexité totale est donc majorée par  $aO(M' \log N') + bO(N')$ , où  $a$  et  $b$  sont des constantes petites.

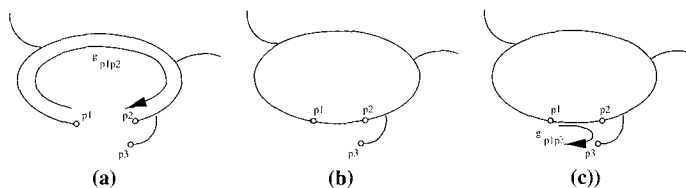
En général, le graphe de réseau manipulé est un graphe très peu dense ( $M \ll \frac{N(N-1)}{2}$ ), le nombre de connexions des traits pleins est très faible par rapport à la taille du réseau, ainsi que le nombre de composantes connexes initiales du réseau. Les temps de calcul seront donc raisonnables.

A la fin de cette étape, il ne reste dans la liste des connexions possibles que celles qui créent des cycles. L'algorithme décrit plus loin exploitera cette liste de connexions pour terminer la reconstruction, en utilisant une propriété dite de connexité locale.

### 5.4.2. propriété de connexité locale

#### 5.4.2.1. algorithme

Nous rappelons la propriété de connexité locale : « Si la distance géométrique entre deux points est faible, la longueur du plus court chemin entre ces deux points à travers le réseau ne doit pas être très grande ». La figure 15 donne trois exemples pour illustrer cette propriété.



**Figure 15. – La connectivité locale et son application.**  
 Dans la figure (a), les arêtes  $[p_1p_2]$  et  $[p_1p_3]$  ne vérifient pas la connectivité locale. Ces deux connexions sont donc permises. Nous avons choisi de connecter  $p_1$  et  $p_2$  car c'est la meilleure des deux connexions au sens de la fonction de poids (figure (b)). Maintenant,  $[p_1p_3]$  respecte la propriété de connectivité locale, et donc cette connexion est rejetée (figure (c)). En pratique, nous ajoutons au graphe, les arêtes qui ne vérifient pas cette propriété. De cette façon, nous évitons de créer de petits cycles.

Formellement, à l'aide d'un algorithme de plus court chemin [40], nous avons : Soit  $d_{AB}$  la distance géométrique entre A et B,  $g_{AB}$  la longueur du plus court chemin entre A et B dans le graphe. Nous avons la relation suivante :

$$\text{Connecter}(A, B) = (g_{AB}/d_{AB} > K_{min}) + \wedge (g_{AB}d_{AB} > L_{min}) \quad (5)$$

$K_{min}$  est la valeur minimale que peut prendre le rapport entre  $g_{AB}$  et  $d_{AB}$ .  $L_{min}$  est la longueur minimale autorisée d'un cycle créé par la connexion  $[AB]$ . Si la condition sur  $K_{min}$  ou sur  $L_{min}$  n'est pas vérifiée, la connexion est rejetée.

**Remarque :** cette propriété est utilisée pour la fermeture des cycles, mais elle est, en fait, implicite pour la reconstruction par l'ARPM si nous considérons que la distance  $g_{AB}$  entre deux points n'appartenant pas aux mêmes composantes connexes dans le graphe est infinie.

#### 5.4.2.2. complexité

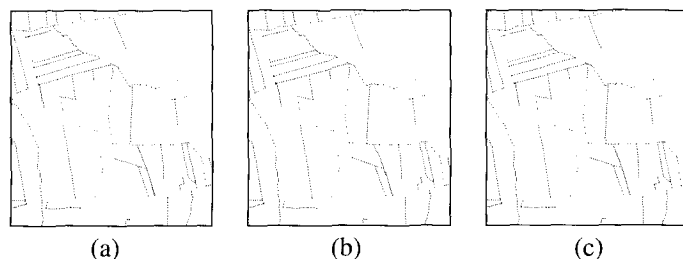
La complexité de cet algorithme est plus simple à calculer : soient  $L$  la liste d'arcs envisagés et  $R = |L|$ ; soient  $G = (X, U)$  le graphe initial et  $G' = (X', U')$  le graphe final,  $N = |X|$ ,  $M = |U|$ ,  $N' = |X'|$  et  $M' = |U'|$ . Pour chaque arc nous avons les opérations élémentaires suivantes :

- Un test d'intersection avec le graphe (topologie) :  $O(\alpha \log N')$  (cf. § 4.3).
- Un calcul de plus court chemin, majoré par :  $O(m \log N')$  [40] (cf. § 4.3).
- Une somme, une division et deux test et un ajout d'arc (formation d'un T) :  $O(8)$ .

Un majorant de cette complexité est donc :  $O(RM' \log N') + O(8R)$ . En général,  $R$  est petit, le graphe est creux et les tests sur  $K_{min}$  et  $L_{min}$  sont intégrés dans le calcul du plus court chemin et permettent de stopper la recherche rapidement.

#### 5.4.3. résultats

Lors de la phase de transformation des tirets en traits pleins (cf. § 4) nous avons obtenu un graphe des traits pleins interrompus (cf. figure 16 à gauche). Après la première étape de reconstruction par l'ARPM (cf. figure 16 au milieu), les connexions possibles restantes sont celles qui ferment les cycles. Nous sélectionnons de façon itérative, la meilleure connexion possible et l'ajoutons au graphe (cf. figure 16 à droite). Celle-ci vérifie trois critères : elle a le poids le plus faible, elle respecte la topologie et elle ne vérifie pas la propriété de connectivité locale. Ces deux derniers critères sont calculés à partir du graphe obtenu après chaque itération.



**Figure 16. – Connexion des traits pleins entre eux.**  
 (a) graphe des traits pleins initial, (b) après l'ARPM, et (c) après fermeture des cycles. (500×500 pixels).

### 5.5. conclusion

À l'issue de cette étape, nous avons transformé les lignes tiretées en traits pleins, puis connecté les traits pleins entre eux. Afin de préserver la topologie, ces opérations sont réalisées en tenant compte d'une contrainte de planarité (cf. 6.4.3). Pour la connexion des traits pleins, nous avons utilisé un découpage polygonal, précédé pour des raisons de performance, d'un découpage elliptique où l'ellipse définit un domaine de visibilité pour une arête source donnée.

## 6. extensions pour le réseau hydrographique

Le réseau hydrographique a la particularité, par rapport au réseau routier, de contenir des objets surfaciques. Ces objets sont fragmentés. Il faut les reconstruire puis les connecter aux traits pleins.

### 6.1. reconstruction de surfaces

Ce traitement concerne la planche de bleu. Ici, la fragmentation des objets surfaciques est due à la présence, au même endroit, de

routes de la planche des noirs (cf. figure 1 et cf. figure 9). Par opposition aux routes, qui ont une épaisseur constante, les objets surfaciques du réseau hydrographique ont une épaisseur variable et ne sont pas forcément linéaires. Ainsi, nous avons développé une méthode appropriée de connexion de surfaces, différente de celle utilisée pour la connexion des routes.

Nous travaillons à partir d'une image ternaire  $I$  qui ne contient que les objets surfaciques : la texture est codée 1, et les bords 2. Le principe (cf. figure 17) est de calculer le squelette de l'image  $J$ , où le fond de  $I$  est codé 1 et le reste 0, de l'ébarbuler et d'en calculer l'approximation polygonale. L'ébarbulation se fait selon des critères d'angle, d'aire et de distance, décrits dans [1]. Nous obtenons alors un graphe, dont chaque arête passe par une zone où l'objet est supposé être fragmenté.

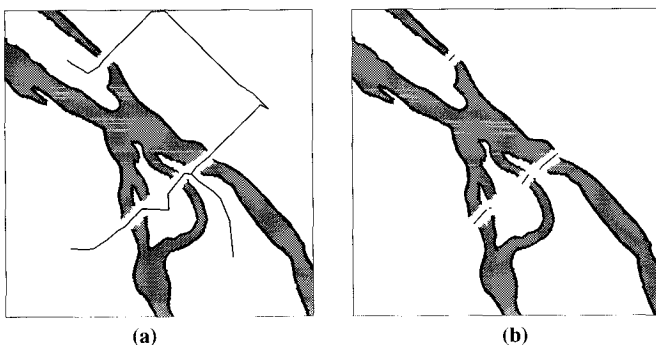


Figure 17. – Localisation des zones de fragmentation. (a) image de l'objet texturé et du squelette du complémentaire; (b) squelette ébarbulé

Le critère d'aire est un seuil sur l'aire des zones d'influences et nous permet d'éliminer les zones étroites, alors que le critère d'angle est un seuil sur les zones d'influences étalées, qui s'exprime pour un point du squelette, comme le rapport entre l'aire et la plus courte distance de la zone d'influence au carré. Le seuillage sur la distance se fait en sélectionnant les points du squelette qui sont proches de la matière.

Ensuite, nous associons à chaque arête  $[p_1, p_2]$ , deux rectangles  $r_1$  et  $r_2$ , ayant pour côté adjacent l'arête  $[p_1, p_2]$ , et pour hauteur une valeur fixée.

D'abord, on détermine si l'arête  $[p_1, p_2]$  est susceptible de générer une connexion. Pour chaque point du segment de Bresenham [42]  $[p_1, p_2]$ , une sonde est lancée perpendiculairement à  $[p_1, p_2]$  (cf. figure 18 (a)). Celle-ci s'arrête au premier pixel de code positif, et on incrémente alors un compteur de pixels codés 2 (bords de texture). A la fin, si le rapport entre le compteur et le nombre de points de  $[p_1, p_2]$  excède un certain seuil, l'arête  $[p_1, p_2]$  n'est pas retenue.

Soit  $F$ , une fonction calculant le nombre de pixels de l'image binaire  $I$  de valeur 1 (objets) dans la zone définie par le rectangle  $r$ . On a :

$$F(r) = \text{Card}\{p : I(p) = 1 \wedge p \in (\text{Dom}(I) \cap r)\}$$

avec  $I(p)$ , valeur au point  $p$  de  $I$ , et  $\text{Dom}(I)$  le domaine de définition de l'image. Pour les deux rectangles  $r_1$  et  $r_2$ , nous définissons  $G$  telle que :

$$G(r_1, r_2) = \min(F(r_1), F(r_2))$$

Puis, en faisant tourner légèrement et simultanément  $r_1$  et  $r_2$  autour du centre de l'arête, nous calculons le couple  $(a, b)$  qui réalise le maximum de  $G$ . A la fin, nous remplissons un polygone dont les quatre sommets sont calculés comme le montre la figure suivante : (cf. figure 18)

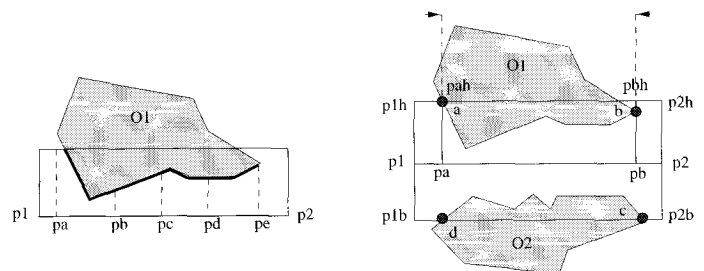


Figure 18. – Création de connexion. (a) une arête invalide (présence de bord de texture); (b) Synthèse de la connexion.

L'exemple ci-dessus (b) montre comment sont calculés les points  $a$  et  $b$ , à partir du rectangle  $r_1 = (p_1, p_{1h}, p_{2h}, p_2)$  porté par l'arête  $[p_1, p_2]$  du squelette de l'image  $J$ . Ici, on veut reconnecter l'objet fragmenté composé de  $O_1$  et  $O_2$ . Pour calculer le point  $a$ , nous déplaçons un point  $p_a$  sur le segment  $[p_1, p_2]$  en partant de  $p_1$ , et considérons le segment  $[p_a, p_{ah}]$  perpendiculaire à  $[p_1, p_2]$ . Ceci est répété jusqu'à ce que  $[p_a, p_{ah}]$  coupe l'objet  $O_1$ . Nous appelons  $a$ , le point de plus proche de  $p_a$  sur le segment  $[p_a, p_{ah}]$ . Le calcul du point  $b$  est analogue, avec  $p_b$  qui varie de  $p_2$  vers  $p_a$ . Le calcul des points  $c$  et  $d$ , pour l'objet  $O_2$  s'effectuent de la même manière à partir du rectangle  $r_2 = (p_1, p_2, p_{2b}, p_{1b})$ . A la fin, le polygone  $(abcd)$  est rempli.

Les images qui suivent (cf. figure 19) montrent un exemple de reconstruction.

## 6.2. connexions des traits pleins aux surfaces

Cette étape est la dernière du processus de reconstruction du réseau et est en cours de réalisation. Elle ne concerne que le réseau hydrographique. L'objectif de cette opération est de rajouter les connexions vraisemblables entre le réseau linéaire et les surfaces tout en considérant les contraintes prédéfinies (cf. § 4.3). Les surfaces seront vectorisées par suivi de contour et approximation polygonale, par exemple avec la convention « matière à droite », et ensuite les traits pleins seront prolongés, au niveau de leurs extrémités, sur ces contours.

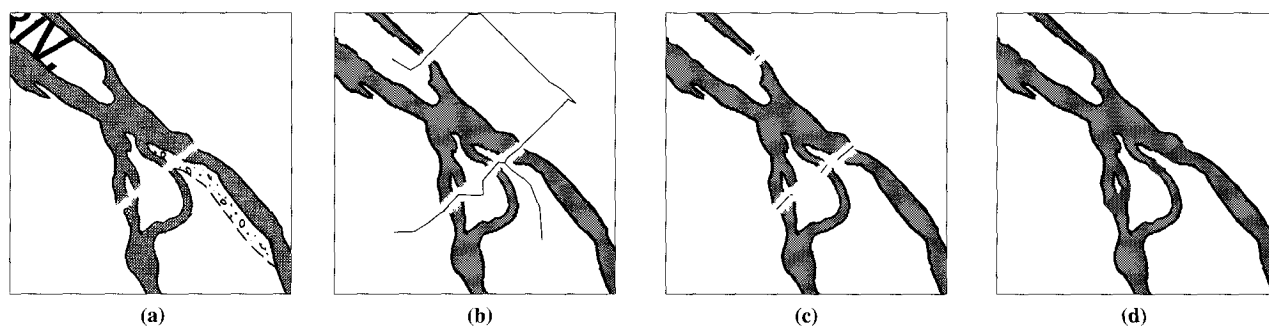


Figure 19. – Reconstruction de surface.

(a) image initiale; (b) image de la texture, du bord et du squelette ébarbuté; (c) seuillage sur la distance du squelette et approximation polygonale; (d) synthèse des connexions; (500×500 pixels).

## 7. résultats et performances

Les tests effectués ici, portent sur la planche de Noves. Cette dernière offre une bonne représentation du réseau hydrographique en plaine, notamment pour les différents niveaux hiérarchiques (tirets, traits, surfaces) représentés dans des proportions équilibrées. Elle offre enfin l'avantage d'être à la fois simple et complète.

Pour le réseau hydrographique, nous avons choisi comme paramètres de la fonction de poids des tirets (cf. § 4.2) :  $\delta = \pi/4$ ;  $\mu = 1.0$ , et 100 comme seuil de rejet d'une connexion (valeur de  $w()$ ) (cf. § 4.2)). Pour la connexion des traits pleins (cf. § 5.2), les seuls paramètres à régler sont les longueurs des demi-axes principaux de l'ellipse de visibilité. Pour le prolongement, nous avons choisi 100 pixels, et pour la perpendiculaire 40 pixels. Ici, l'image de gauche (cf. figure 20) représente l'image initiale et celle de droite l'image du réseau reconstruit. La reconstruction en mode vecteur sur la planche complète s'est effectuée en 1 minute et 20 secondes.

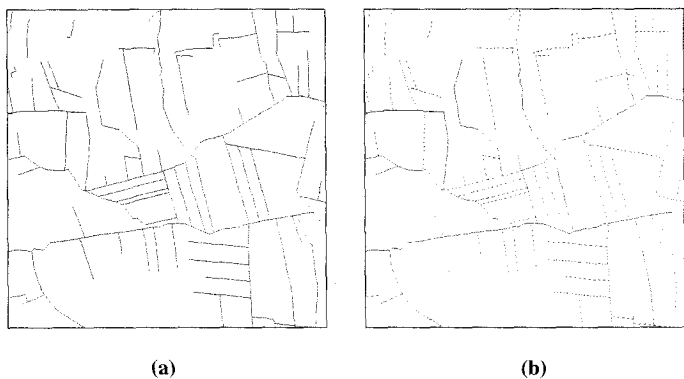


Figure 20. – Reconstruction du réseau hydrographique.  
(a) image initiale; (b) réseau reconstruit (2000×2000 pixels).

Pour le réseau routier (cf. figure 21), nous avons choisi  $\delta = \pi/3$ ;  $\mu = 1.0$  et comme seuil de rejet d'une connexion 200. Les longueurs des demi-axes principaux de l'ellipse ont pour valeur : prolongement 120 pixels et perpendiculaire 40 pixels. Les performances sont encore plus importantes car le graphe est beaucoup plus creux et il n'y a pas de génération de carrefour à traiter.

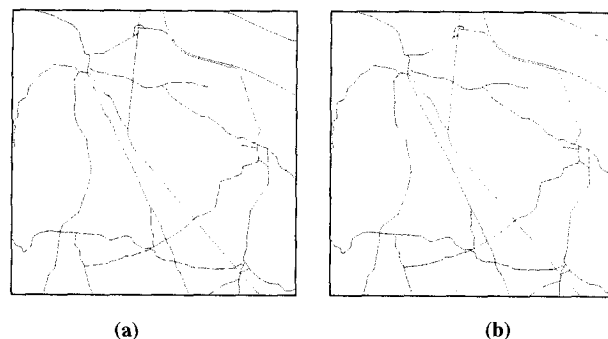


Figure 21. – Reconstruction de réseau routier.  
(a) image initiale; (b) réseau reconstruit (Image 5000×5000).

Il y a deux sources possibles de mauvaise reconstruction que le système doit être en mesure de signaler : celles qui sont dues à une mauvaise simplification de la carte, et celles dues à des situations ambiguës. Une mauvaise simplification de la carte est obtenue, par exemple, en oubliant un disque ou un caractère trop bruité (cas rare) dans l'image initiale. Une situation ambiguë est celle pour laquelle il n'existe pas de solution triviale, voire pas de meilleure solution. Un tel cas, pour lequel il existe plusieurs reconstructions possibles a priori, est présenté dans les images suivantes (cf. figure 22) :

Ce cas ne s'est présenté qu'une seule fois pour la planche de Noves, et concerne uniquement les carrefours du réseau hydrographique, les carrefours du réseau routier étant toujours représentés. Pour corriger ou signaler de telles reconstructions, il faudra réaliser un outil spécialisé sur les carrefours du réseau hydrographique.

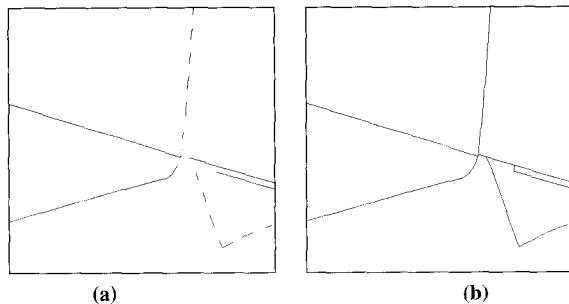


Figure 22. – Un cas ambigu de reconstruction.  
(a) image initiale; (b) réseau reconstruit (Image 500×500).

D'autres tests ont été effectués en région montagneuse, qui soulèvent des problèmes liés au relief. Le réseau routier ne posant pas de problèmes ici, nous ne présentons que les résultats obtenus sur le réseau hydrographique. Le paramétrage est le même que celui utilisé pour Noves.

Les principaux problèmes de validité topologique viennent du fait de la création de connexions impossibles. Ces connexions invalides ne peuvent pas être détectées dans le seul cadre bidimensionnel de l'image. Ici, il faut avoir une connaissance a priori sur le relief, afin d'éviter, par exemple, de connecter deux ruisseaux séparés par un col. L'utilisation de la BDZ©, la base de données altimétriques de l'IGN, comme contrainte supplémentaire dans le processus de reconstruction de réseau, s'inscrit dans la continuité logique de ces travaux.

## 8. conclusion et perspectives

L'IGN a pour objectif de développer sur la carte IGN au 1/25000 un système d'interprétation totalement automatique et complet de la carte. Le but est de convertir automatiquement le fond de cartes existant sous forme papier, en une base de données d'objets géographiques directement manipulables par un SIG. L'IGN étant un producteur de cartes, nous travaillons directement sur les planches mères. Les travaux exposés ici portent sur la planche de bleu, qui contient le thème hydrographique, et sur la planche de rouge qui contient l'intérieur de certaines routes de la planche de noir.

L'objectif de ce travail est d'automatiser l'extraction des réseaux hydrographique et routier. Dans les deux cas, il y a une phase de prétraitements de la planche, qui vise à reconnaître et extraire tous les thèmes qui ne concernent pas les réseaux, puis à vectoriser l'image restante qui ne contient que des tirets et des traits pleins, et enfin à séparer les tirets des traits pleins. La reconstruction est quant à elle divisée en quatre étapes : 1) la transformation des tirets en traits pleins; 2) la connexion des traits pleins entre eux; 3) la reconstruction des surfaces fragmentées; 4) la connexion des traits pleins aux surfaces.

Nous insistons ici sur l'extraction et la reconstruction des graphes des réseaux hydrographique et routier. L'algorithme est générique et il s'appuie sur les connaissances a priori définies sur les réseaux réels et cartographiques. Le formalisme utilisé est celui de la théorie des graphes. Cette représentation des réseaux à l'aide de graphes est naturelle et offre un moyen d'explicitier les contraintes sur ces objets, ainsi que les mécanismes de reconstruction. La reconstruction des réseaux linéaires est réalisée à l'aide de l'ARPM et de la connexité locale. La contrainte de planarité est maintenue au fur et à mesure du déroulement de la reconstruction avec un algorithme de découpage polygonal. Les vraisemblances des connexions, spécifiques pour chaque réseau et pour chaque type de tracé, ont été élaborées en utilisant les connaissances a priori sur les deux réseaux.

Les tests effectués portent sur Noves. Elle offre une bonne représentation des réseaux hydrographique et routier en plaine. En effet, au niveau du réseau hydrographique, c'est une planche assez complète qui présente bien les différents niveaux hiérarchiques du réseau, des rivières au débit important aux cours d'eau temporaires, et ceci dans des proportions équilibrées. Enfin, l'obtention du réseau routier de la planche de rouge, permet de valider partiellement celui extrait de la planche de noir, et ainsi de tester des mécanismes de validation inter-thèmes et inter-planches.

L'algorithme de reconstruction de surface ne convient pas, car il est insuffisant pour détecter une incohérence de connexion. Par contre, il nous a permis de fournir une ébauche d'une solution. La localisation de zones où la reconstruction est probable, restreint déjà l'espace de recherche, mais la façon dont est calculée une connexion peut amener à des incohérences. Pour cela, nous travaillons sur une approche totalement vecteur, avec les contours de la texture orientés avec la convention « matière à droite ». Ainsi, une connexion sera composée d'un couple de vecteurs anti-parallèles et mutuellement visibles avec la convention précédente (l'un se trouvant à la droite de l'autre et vice-versa), et les tests de validité topologique seront alors possibles.

Enfin, les connexions ajoutées aux réseaux sont hypothétiques et d'origines différentes. Il est nécessaire de mettre en place un système de validation. Ce dernier s'inspirera des travaux de [14, 13]. Pour les traits pleins et les objets surfaciques, les connexions sont calculées en supposant que les interruptions sont dues à d'autres objets. Ici, un processus de validation inter-planches doit être développé. Pour les lignes tiretées, les connexions sont principalement justifiées par les règles cartographiques utilisées pour les représenter. Un processus de validation intra-planche doit aussi être mis en oeuvre. De plus, l'intégration de données altimétriques issues de la BDZ© se fait ressentir pour la reconstruction des réseaux en zones montagneuses.

Finalement, des algorithmes de couplage de thèmes doivent être développés. Ils concernent notamment l'association des toponymes avec le graphe du réseau hydrographique, l'attribution « cours d'eau bordés d'arbres », aux éléments effectivement bordés d'arbres (disques).



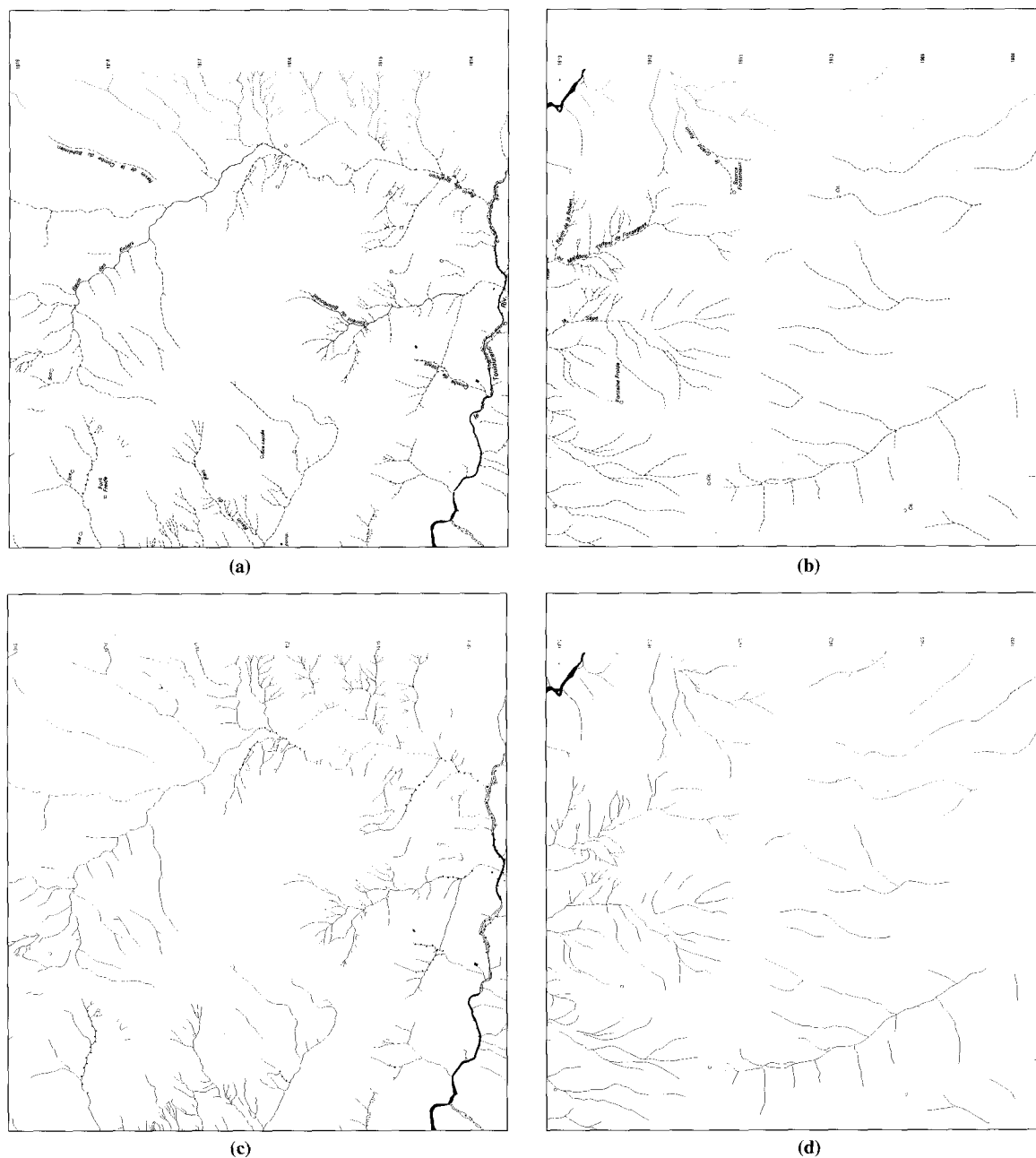


Figure 23. – Réseau hydrographique sur le Mont-Ventoux.  
Les images (a) et (c) sont les images initiales; l'image (b) (resp. (d)) représente le réseau hydrographique extrait. (Image 4000×4000).

## BIBLIOGRAPHIE

- [1] M. Pierrot Deseilligny, *Lecture Automatique de Cartes*. Thèse, Université René Descartes, Centre Universitaire des Saints-Pères, UFR de Mathématiques et Informatique, Paris V, Oct. 1994.
- [2] A. Khotanzad and E. Zink, "Color Paper Map Segmentation Using Eigenvector Line Fitting," in *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation, San Antonio, TEXAS (USA)*, pp. 190–194, Apr. 1996.
- [3] L. Lefrère, *Contribution au Développement d'Outils pour l'Analyse Automatique de Documents Cartographiques*. Thèse, Université de Rouen, Rouen, Oct. 1993.
- [4] J. Ogier, *Contribution à l'Analyse Automatique de Documents Cartographiques : Interprétation de Données Cadastrales*. Thèse, Université de Rouen, Rouen, Jan. 1994.
- [5] L. Boatto and al., "An Interpretation System for Land Register Map," *IEEE Computer Society Press*, vol. 25, pp. 25–33, July 1992.

## Interprétation de cartes géographiques

- [6] P. Vaxivière and K. Tombre, "Ceslstin : CAD Conversion of Mechanicals Drawings," *IEEE Computer Society Press*, vol. 25, pp. 46–54, July 1992.
- [7] S. Joseph and T. Pridmore, "Knowledge-Directed Interpretation of Mechanical Engineering Drawings," *IEEE Transactions on PAMI*, vol. 14, pp. 928–940, Sept. 1992.
- [8] B. Couasnon, « Formalisation de la Connaissance Grammaticale *a priori* pour l'Analyse de Documents : Application aux Partitions d'Orchestre », in *Actes 10<sup>ème</sup> Congrès AFCET Reconnaissance des Formes et Intelligence Artificielle*, Rennes, pp. 465–474, Jan. 1996.
- [9] I. Leplumey, J. Camillerapp, and G. Lorette, "A Robust Detector for Music Staves," in *Proceedings of 2nd International Conference on Document Analysis and Recognition, Tsukuba (Japan)*, pp. 902–905, Oct. 1993.
- [10] R. Thomson, "Micrograph Analysis and Image Understanding," in *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation, San Antonio, TEXAS (USA)*, pp. 48–53, Apr. 1996.
- [11] J. McDaniel and J. Balmuth, "Automatic Interpretation of Chemical Structure Diagrams," in *Lecture Notes in Computer Sciences*, vol. 1072, pp. 148–158, 1996.
- [12] R. Mullot, F. Brisepierre, Y. Lecourtier, M. Collinas, and J. Ogier, "Méthode Robuste de Localisation de Vignettes Circulaires sur des Plans Techniques," in *Actes du Quatrième Colloque National sur l'Écrit et le Document*, Nantes, pp. 77–84, July 1996.
- [13] G. Myers, P. Mulgaonkar, C. Chen, J. DeCurtins, and E. Chen, "Verification-Based Approach for Automated Text and Feature Extraction From Raster-Scanned Maps," in *Proceedings of IAPR International Workshop on Graphics Recognition, Penn State Scanticon (USA)*, pp. 90–99, Aug. 1995.
- [14] J. Den Hartog, T. Ten Kate, and J. Gerbrands, "Knowledge-Based Segmentation for Automatic Map Interpretation," in *Proceedings of IAPR International Workshop on Graphics Recognition, Penn State Scanticon (USA)*, pp. 71–80, Aug. 1995.
- [15] R. Mariani, M. Pierrot Deseilligny, J. Labiche, Y. Lecourtier, and R. Mullot, "Geographic Map Understanding. Algorithms for Hydrographic Network Reconstruction," in *Lecture Notes in Computer Sciences*, pp. 514–515, Dec. 1995.
- [16] M. Pierrot Deseilligny, H. Le Men, and G. Stamon, "Map Understanding for GIS Data Capture. Algorithms for Road Network Graph Reconstruction," in *Proceedings of 2nd International Conference on Document Analysis and Recognition, Tsukuba (Japan)*, pp. 676–679, Oct. 1993.
- [17] M. Gondran and M. Minoux, *Graphes et Algorithmes*, pp. 1–33. Vol. 1 of de la Direction des Etudes et Recherches d'Electricité de France [43], 1985.
- [18] L. O'Gorman and R. Kasturi, *Document Image Analysis*, pp. 101–105. Los Alamitos, California : IEEE Computer Society Press, 1995.
- [19] R. Duda and P. Hart, "Use of the Hough Transform to Detect Lines and Curves in Pictures," *Communication of the Association of Computer Machinery*, vol. 15, no. 1, pp. 11–15, 1972.
- [20] H. Yamade, K. Yamamoto, and K. Hosokawa, "Directional Mathematical Morphology and Reformalized Hough Transformation for the Analysis of Topographic Maps," *IEEE Transactions on PAMI*, vol. 15, no. 4, pp. 380–387, 1993.
- [21] G. Agam, H. Luo, and I. Dinstein, "Morphological Approach for Dashed Lines Detection," in *Proceedings of IAPR International Workshop on Graphics Recognition, Penn State Scanticon (USA)*, pp. 23–32, Aug. 1995.
- [22] T. Kasvand, "Linear Texture in Line Drawing," in *Proceedings of 8th International Conference on Pattern Recognition, Paris (France)*, pp. 398–401, Oct. 1986.
- [23] C. Lai and R. Kasturi, "Detection of Dashed lines in Engineering Drawings and Maps," in *Proceedings of First International Conference on Document Analysis, Saint-Malo, France*, pp. 507–514, 1991.
- [24] D. Dori, L. Wenyin, and M. Peleg, "How to Win a Dashed Line Detection Contest," in *Lecture Notes in Computer Sciences*, vol. 1072, pp. 286–300, 1996.
- [25] T. Nagao, T. Agui, and M. Nakajima, "An Automatic Road Vector Extraction Method from Maps," in *Proceedings of 9th International Conference on Pattern Recognition, Rome (Italy)*, pp. 585–587, 1988.
- [26] D. Antoine, "A Technical Document Understanding System Based on a priori Knowledge," in *Proceedings of 6th Scandinavian Conference on Image Analysis, Oulu (Finland)*, pp. 843–846, Dec. 1989.
- [27] O. Hori and A. Okasaki, "High Quality Vectorization Based on a Generic Object Model," in *Pre-proceedings of IAPR Workshop on Syntactic and Structural Pattern Recognition, Murray Hill, NJ (USA)*, pp. 137–153, Aug. 1990.
- [28] J. Ogier, J. Labiche, R. Mullot, and Y. Lecourtier, "Attributes Extraction for French Map Interpretation," in *Proceedings of 2nd International Conference on Document Analysis and Recognition, Tsukuba (Japan)*, pp. 672–675, Oct. 1993.
- [29] M. Pierrot Deseilligny, H. Le Men, and G. Stamon, « Lecture automatique des écritures sur cartes scannées », in *Actes du Troisième Colloque National sur l'Écrit et le Document, Rouen*, pp. 195–202, July 1994.
- [30] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*, pp. 477–506. London Glasgow Weinheim New-York Tokyo Melbourne Madras : Chapman and Hall Computing, 1994.
- [31] M. 31 and J. Chermant, *Précis d'Analyse d'Images*, pp. 82–86. Paris : Presses du CNRS, 1989.
- [32] E. Davies, "A Modified Hough Scheme for General Circle Location," *Pattern Recognition Letters*, vol. 7, pp. 37–43, 1988.
- [33] S. Thomas and Y. Chan, "A Simple Approach for the Estimation of Circular Arc Center and Its Radius," in *Computer Vision, Graphics and Image Processing*, vol. 45, pp. 362–370, 1989.
- [34] R. Takiyama and N. Ono, "A Least Square Error Estimation of the Center and Radii of Concentric Arcs," in *Pattern Recognition Letters*, vol. 10, pp. 237–242, 1989.
- [35] D. De Bruccq, M. Amara, and V. Ruiz, « Segmentation de Tracés Manuscrits par Arcs de Cercle », in *Actes du Quatrième Colloque National sur l'Écrit et le Document, Nantes*, pp. 155–161, July 1996.
- [36] S. Beucher and L. Vincent, "Introduction aux Outils Morphologiques de Segmentation," tech. rep., Centre de Morphologie Mathématique. Ecole des Mines, 1989.
- [37] C. Maquair, « Reconnaissance des Plans de Réseau EDF : Etude de Faisabilité », tech. rep., ENSEA, Université de Cergy Pontoise, Sept. 1996.
- [38] M. Pierrot Deseilligny, *Annexes Lecture Automatique de Cartes*. Thèse, Université René Descartes, Centre Universitaire des Saints-Pères, UFR de Mathématiques et Informatique, Paris V, Oct. 1994.
- [39] L. Vincent, *Algorithmes Morphologiques à Base de Files d'Attentes et de Lacets. Extension aux Graphes*. Thèse, Ecole Nationale Supérieure des Mines de Paris, Paris, 1990.
- [40] M. Gondran and M. Minoux, *Graphes et Algorithmes*, pp. 35–68. Vol. 1 of de la Direction des Etudes et Recherches d'Electricité de France [43], 1985.

[41] M. Gondran and M. Minoux, *Graphes et Algorithmes*, pp. 112–115. Vol. 1 of de la Direction des Etudes et Recherches d'Electricité de France [43], 1985.

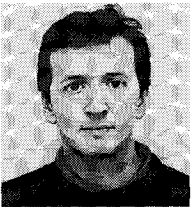
[42] Y. 42, *Elements de CAO. Matériels et Logiciels de Base*, vol. 1, pp. 179–197. Paris, Londres, Lausanne : Hermès, 1988.

[43] M. Gondran and M. Minoux, *Graphes et Algorithmes*. Collection de la Direction des Etudes et Recherches d'Electricité de France, 1985.

*Manuscrit reçu le 12 juin 1996*

### LES AUTEURS

#### Robert MARIANI



Robert Mariani est né le 27 novembre 1967. Après un DEA informatique de l'Université de Nancy 1, il termine une thèse menée à l'Institut Géographique National, dans l'équipe « cartes scannées », portant sur la lecture automatique de cartes au 1/25000.

#### Yves LECOURTIER



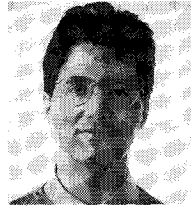
Yves Lecourtier a soutenu un doctorat du troisième cycle en traitement du signal en 1978 puis un doctorat d'état en automatique théorique en 1985 à l'Université Paris-Sud. Son domaine de recherche portait alors principalement sur l'étude des propriétés structurelles des modèles dynamiques décrits par équations d'état. Professeur à l'Université de Rouen depuis 1987, il a réorienté ses activités vers la reconnaissance de formes et les réseaux neuro-mimétiques avec applications principales à l'analyse de documents et la reconnaissance de l'écrit. Il préside actuellement le GRCE, groupe de recherche en communication écrite, qui rassemble les chercheurs français du domaine.

#### Marc 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 mois à l'État-major 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 scannées » à l'Institut Géographique National.

#### Fabrice LECOURT



Fabrice Lecourt est né le 23 août 1970. Il est ingénieur des travaux géographiques et cartographiques de l'État à l'Institut Géographique National, et titulaire d'un DEA en sciences d'informations géographiques. Il travaille depuis 1994 dans l'équipe « cartes scannées ».

#### Jacques LABICHE



Jacques Labiche est né le 19 Janvier 1945. Après un doctorat de troisième cycle de Physique du solide, soutenu à l'Université de Rouen en 1975, un thèse d'état de Physique soutenu à Bordeaux en 1984, il devient en 1986 maître de Conférences à l'IUT de Rouen. Parallèlement, il réoriente son domaine de recherche vers la reconnaissance de caractères et de documents au sein du LA31-LACIS. Depuis 1994, Professeur à l'Université de Caen, il co-dirige un laboratoire d'automatique (LACP) où il développe une stratégie « vision active » pour la saisie de documents.