

## **NOTE TO USERS**

**This reproduction is the best copy available**

**UMI**



DAVID THIBAUT

**LA TRANSFORMÉE DE L'AXE MÉDIAN: APPLICATIONS  
À LA GÉNÉRALISATION CARTOGRAPHIQUE ET  
À LA MODÉLISATION DU TERRAIN**

Mémoire présenté  
à la Faculté des études supérieures  
de l'Université Laval  
pour l'obtention  
du grade de maître ès sciences (M.Sc.)

Département des sciences géomatiques  
FACULTÉ DE FORESTERIE ET DE GÉOMATIQUE  
UNIVERSITÉ LAVAL

Décembre 1999



**National Library  
of Canada**

**Acquisitions and  
Bibliographic Services**

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

**Bibliothèque nationale  
du Canada**

**Acquisitions et  
services bibliographiques**

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file Votre référence*

*Our file Notre référence*

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-48960-4

## RÉSUMÉ

Par ses propriétés géométriques uniques, dont celle d'être un descripteur de forme, et par ses relations étroites avec la triangulation Delaunay, l'axe médian ou squelette peut être utilisé pour solutionner divers problèmes fréquemment rencontrés en géomatique. Gold (1999) a développé, à partir des travaux d'Amenta et al. (1998), un algorithme permettant d'extraire la croûte et l'axe médian d'un ensemble non structuré de points. L'algorithme de Gold (1999) a donné naissance à deux applications.

La première est un algorithme de généralisation cartographique. La rétraction des branches de l'axe médian permet la généralisation des lignes et des polygones, tout en rendant impossible le chevauchement des lignes et en assurant le maintien des relations topologiques entre les entités cartographiques. La deuxième application est une méthode permettant d'enrichir les modèles numériques d'élévation (TIN) et, par la même occasion, d'y intégrer de l'information topographique additionnelle pour accroître leur fiabilité.

Cette recherche vise à l'atteinte des objectifs suivants. En premier lieu, elle souhaite apporter, par le développement des applications ci-haut mentionnées, des solutions à deux problèmes soulevés dans la littérature scientifique: 1) les algorithmes de généralisation cartographique entraînent parfois des chevauchements d'objets lorsque ceux-ci sont près les uns des autres (Weibel, 1990); 2) la génération d'une triangulation à partir de courbes de niveau produit des triangles plats aux zones d'inflexion prononcée (Van Kreveld, 1997). En second lieu, ce mémoire veut démontrer le potentiel d'applicabilité de l'axe médian pour le développement d'algorithmes simples et puissants destinés au traitement des données spatiales.

## **AVANT-PROPOS**

Je tiens tout d'abord à remercier très sincèrement mon directeur de maîtrise, monsieur Christopher Gold, pour l'excellence de son enseignement, pour ses précieux conseils et pour le grand dévouement dont il a fait part. Également, j'aimerais remercier pour leurs commentaires messieurs Alfonso R. Condal et Brian Morse, tous deux examinateurs de mon mémoire. Enfin, je tiens à remercier tous mes collègues et ami(e)s, ainsi que le personnel du Centre de Recherche en Géomatique de l'Université Laval, pour leur aide et leur appui.

Finalement, j'aimerais remercier la Chaire Industrielle de Géomatique appliquée à la foresterie de l'Université Laval pour le financement qui a rendu possible cette recherche.

# TABLE DES MATIÈRES

**RÉSUMÉ**

**AVANT-PROPOS**

**TABLE DES MATIÈRES**

**TABLE DES ILLUSTRATIONS**

## 1. INTRODUCTION

## 2. NOTIONS THÉORIQUES

- 2.1 Structure de données Quad-Edge p. 5
- 2.2 La triangulation Delaunay et le diagramme de Voronoï
  - 2.2.1 Définition et propriétés p. 8
  - 2.2.2 Algorithmes p. 11
- 2.3 L'axe médian et la croûte
  - 2.3.1 L'axe médian p. 14
  - 2.3.2 La croûte p. 18
  - 2.3.3 Algorithmes d'extraction de l'axe médian et/ou de la croûte
    - 2.3.3.1 L'approche d'Amenta et al. (1998) p. 20
    - 2.3.3.2 L'approche de Gold et al. (1999) p. 20

## 3. GÉNÉRALISATION PAR RÉTRACTION DE L'AXE MÉDIAN

- 3.1 Revue des travaux sur la généralisation p. 23
- 3.2 Algorithme de généralisation par rétraction du squelette
  - 3.2.1 Présentation p. 24
  - 3.2.2 Caractéristiques de l'algorithme p. 26
  - 3.2.3 Description du code de programmation p. 27

## 4. APPLICATIONS

- 4.1 Applications utilisant les courbes
  - 4.1.1 Construction de cartes polygonales p. 31
  - 4.1.2 Construction de réseaux p. 34
  - 4.1.3 Construction de courbes de niveau p. 35

<b>4.2 Applications utilisant l'axe médian</b>	<b>p. 35</b>
<b>5. MODÉLISATION DE TERRAIN</b>	
<b>5.1 Revue des travaux sur la modélisation du terrain</b>	<b>p. 38</b>
<b>5.2 Le modèle grille versus le modèle à base de triangles (TIN)</b>	<b>p. 38</b>
<b>5.3 Les diverses sources de données des MNT</b>	<b>p. 40</b>
<b>5.4 L'interpolation</b>	
5.4.1 Introduction	p. 42
5.4.2 Interpolation à partir de points arbitraires	p. 43
5.4.3 Interpolation par voisin naturel : la méthode "aire volée"	p. 44
<b>5.5 TIN produit à partir d'un semis de points obtenu par photogrammétrie</b>	<b>p. 46</b>
<b>5.6 TIN produit à partir de courbes d'élévation</b>	<b>p. 47</b>
<b>6. ENRICHISSEMENT DE MODÈLES NUMÉRIQUES D'ÉLEVATION (TIN) PAR INTERPOLATION LE LONG DE L'AXE MÉDIAN</b>	
<b>6.1 Pourquoi enrichir un modèle numérique d'élévation?</b>	<b>p. 49</b>
<b>6.2 Méthode d'enrichissement par interpolation le long de l'axe médian</b>	
6.2.1 Présentation	p. 49
6.2.2 Description du code de programmation	p. 55
<b>7. ÉVALUATION DES ALGORITHMES DÉVELOPPÉS</b>	
<b>7.1 Généralisation par rétraction de l'axe médian</b>	<b>p. 62</b>
<b>7.2 Enrichissement de modèles numériques d'élévation (TIN) par interpolation le long de l'axe médian.</b>	<b>p. 67</b>
<b>8. LE MODULE DE VISUALISATION 3D</b>	
<b>8.1 Objectifs</b>	<b>p. 74</b>
<b>8.2 Développement</b>	
8.2.1 Présentation	p. 74
8.2.2 Densification et interpolation du TIN	p. 75
8.2.3 Projection du modèle sur le plan	p. 75
8.2.4 Déplacement de l'observateur autour du modèle	p. 76
8.2.5 Séquence d'envoi des triangles parents et ordre d'affichage des triangles enfants	p. 77

8.2.6 Illumination du modèle.	p. 79
<b>8.3 Flux des opérations à l'intérieur du module de visualisation 3D</b>	<b>p. 80</b>
<b>8.4 Évaluation du module de visualisation 3D</b>	<b>p. 82</b>
<b>9. TRAVAUX FUTURS</b>	
9.1 Généralisation des branches d'ordre 2	p. 86
9.2 Problèmes d'échantillonnage sur les courbes	p. 87
9.3 Modélisation du terrain	
9.3.1 Calcul de pente en travers des courbes de niveau	p. 89
9.3.2 Modèle d'écoulement	p. 89
<b>10. CONCLUSION</b>	<b>p. 91</b>
<b><i>RÉFÉRENCES BIBLIOGRAPHIQUES</i></b>	<b>p. 93</b>

## TABLE DES ILLUSTRATIONS

<b>Figure 2.1</b> Le Quad-Edge permet de connecter, par une série de pointeurs, la triangulation Delaunay (primal) et le diagramme de Voronoï (dual) au sein d'une même unité.	p. 5
<b>Figure 2.2</b> Structure Quad-Edge pour deux triangles.	p. 5
<b>Figure 2.3</b> Création d'un edge avec MakeEdge.	p. 6
<b>Figure 2.4</b> Connexion et déconnexion de deux Quad-Edges avec Splice.	p. 8
<b>Figure 2.5</b> Un triangle Delaunay a comme caractéristique de posséder un cercle circonscrit vide de tout autre site.	p. 8
<b>Figure 2.6</b> La triangulation de Delaunay (DT) est en relation voisine avec le diagramme de Voronoï (V) puisqu'elle constitue son graphe dual.	p. 9
<b>Figure 2.7</b> Diagramme de Voronoï pour un ensemble de onze (11) générateurs sur le plan.	p. 10
<b>Figure 2.8</b> Procédure d'insertion d'un point dans la triangulation Delaunay.	p. 11
<b>Figure 2.9</b> Procédure d'inversion de la diagonale d'un triangle suite à l'insertion d'un point.	p. 13
<b>Figure 2.10</b> L'approche "Diviser pour régner" pour la construction d'un diagramme de Voronoï.	p. 13
<b>Figure 2.11</b> Exemples d'axe médian pour des formes simples.	p. 15
<b>Figure 2.12</b> Axe médian d'une feuille d'érable.	p. 15
<b>Figure 2.13</b> L'axe médian est constitué des segments du diagramme de Voronoï qui ne croisent pas l'enveloppe de l'objet ou croûte.	p. 16

<b>Figure 2.14</b> Procédure permettant de reconstruire l'enveloppe de l'objet ou croûte à partir de l'axe médian	p. 16
<b>Figure 2.15</b> La propagation uniforme de l'excitation sur le plan se fait sous forme de vagues qui ne se recoupent pas.	p. 17
<b>Figure 2.16</b> Analogie du "feu de brousse" pour conceptualiser l'axe médian qui est ici représenté par la ligne rouge.	p. 18
<b>Figure 2.17</b> Un ensemble non structuré de points à gauche, versus la croûte constituée par la connexion de chaque paire de points adjacents à droite.	p. 19
<b>Figure 2.18</b> Définition de la croûte d'après Amenta et al. (1998) basée sur le diagramme Voronoï/Delaunay.	p. 19
<b>Figure 2.19</b> Extraction de la croûte avec l'approche d'Amenta et al. (1998) pour une feuille d'érable.	p. 20
<b>Figure 2.20</b> L'approche de Gold (1999) pour l'extraction de la croûte et de l'axe médian.	p. 21
<b>Figure 3.1</b> Ensemble du processus de généralisation par rétraction de l'axe médian.	p. 25
<b>Figure 3.2</b> L'algorithme de généralisation par rétraction de l'axe médian élimine les perturbations mineures le long des courbes.	p. 26
<b>Figure 3.3</b> L'axe médian externe (exo-squelette) joue le rôle de frontière en isolant chacun des objets cartographiques.	p. 27
<b>Figure 3.4</b> Un tronçon de croûte est constitué de deux edges adjacents appartenant à la croûte et dont leurs trois vertex sont cocirculaires indiquant ainsi l'emplacement d'une feuille de l'axe médian.	p. 27
<b>Figure 3.5</b> Variables utilisées pour le calcul de la nouvelle position d'un vertex sur l'arc de cercle.	p. 28
<b>Figure 4.1</b> Exemples de courbes fermées et ouvertes.	p. 30
<b>Figure 4.2</b> La modèle spaghetti ou le vectoriel non structuré topologiquement.	p. 30

<b>Figure 4.3</b> Transformation Quad-Edge à Quad-Arc.	p. 33
<b>Figure 4.4</b> Axe médian d'un réseau hydrographique.	p. 35
<b>Figure 4.5</b> Axe médian pour une carte urbaine.	p. 36
<b>Figure 4.6</b> Axe médian de trois caractères de l'alphabet.	p. 36
<b>Figure 4.7</b> Axe médian d'un ensemble de courbes de niveau.	p. 37
<b>Figure 5.1</b> Modélisation du terrain à l'aide d'une grille.	p. 39
<b>Figure 5.2</b> Modélisation du terrain par un réseau de triangles irréguliers.	p. 40
<b>Figure 5.3</b> Exemples de captage photogrammétrique par échantillonnage régulier pour deux types de densité.	p. 40
<b>Figure 5.4</b> Exemples d'échantillonnage progressif permettant d'adapter la densité de l'échantillonnage en fonction de la complexité du terrain.	p. 41
<b>Figure 5.5</b> Les isohypses ou courbes de niveau sont les données les plus fréquemment utilisées pour la conception de MNT.	p. 41
<b>Figure 5.6</b> Critères de recherche des points candidats. Au haut, le critère de distance. Au milieu, le critère de nombre. Au bas, le critère basé sur des secteurs.	p. 43
<b>Figure 5.7</b> Le critère Voronoï pour la sélection des points candidats.	p. 44
<b>Figure 5.8</b> Interpolation "aire volée".	p. 44
<b>Figure 5.9</b> Valeurs obtenues par interpolation "aire volée".	p. 45
<b>Figure 5.10</b> Exemple d'un lissage hermitien. Le modèle au bas est la version lissée du modèle présenté au haut.	p. 45
<b>Figure 5.11</b> Exemple du problème des "triangles plats".	p. 48

<b>Figure 6.1</b> Les branches de l'axe médian indiquent l'emplacement de talus ou de petites dépressions suivant l'élévation des courbes de niveaux supérieur et inférieur.	p. 50
<b>Figure 6.2</b> Interpolation de valeurs sur le tronc de l'axe médian représenté ici en rouge.	p. 51
<b>Figure 6.3</b> Première approche pour l'interpolation de valeurs sur les branches de l'axe médian.	p. 51
<b>Figure 6.4</b> Les cercles de la première approche d'interpolation sur les branches de l'axe médian.	p. 52
<b>Figure 6.5</b> Deuxième approche pour l'interpolation de valeurs sur les branches de l'axe médian.	p. 53
<b>Figure 6.6</b> Les cercles de la deuxième approche d'interpolation sur les branches de l'axe médian.	p. 53
<b>Figure 6.7</b> Comparaison des valeurs obtenues à l'aide des deux approches d'interpolation pour une branche de l'axe médian représentant un talweg (ligne qui relie les points les plus bas d'une vallée).	p. 54
<b>Figure 6.8</b> En l'absence d'un point coté, tout le sommet du modèle constitue en fait une zone critique où les triangles sont plats.	p. 55
<b>Figure 6.9</b> Calcul de la valeur du vertex de référence pour l'interpolation au sommet du modèle.	p. 55
<b>Figure 6.10</b> Repérage d'un segment du tronc de l'axe médian pour interpolation.	p. 56
<b>Figure 6.11</b> Repérage d'un vertex de jonction tronc-branche en vue de l'interpolation sur la branche.	p. 57
<b>Figure 6.12</b> Variables utilisées pour l'interpolation sur les branches de l'axe médian.	p. 59
<b>Figure 6.13</b> À peu de chose près, les valeurs obtenues par interpolation au sommet du modèle demeurent sensiblement les mêmes, quel que soit le vertex de référence utilisé.	p. 60

<b>Figure 7.1</b> Étapes successives de la généralisation de la feuille d'érable. Les résultats obtenus après chacune des cinq itérations sont présentés.	p. 62
<b>Figure 7.2</b> Étapes successives pour la généralisation d'une carte polygonale. Les résultats obtenus après chacune des quatre itérations sont présentés.	p. 64
<b>Figure 7.3</b> Étapes successives pour la généralisation d'un fichier de courbes de niveau. Les résultats obtenus après chacune des six itérations sont présentés.	p. 66
<b>Figure 7.4</b> Extrait d'une carte topographique de la région du mont Albert (Québec, Canada)	p. 67
<b>Figure 7.5</b> Vue d'ensemble du modèle non enrichi et non interpolé.	p. 68
<b>Figure 7.6</b> Vue rapprochée de la figure 7.5 avec identification des zones de triangles plats et courbes de niveau intermédiaire à équidistance de 10 mètres.	p. 68
<b>Figure 7.7</b> Vue d'ensemble du modèle enrichi mais non interpolé.	p. 69
<b>Figure 7.8</b> Vue rapprochée de la figure 7.7 avec superposition de l'axe médian et courbes de niveau intermédiaire à équidistance de 10 mètres.	p. 69
<b>Figure 7.9</b> Distribution des courbes de niveau intermédiaire avant et après l'enrichissement.	p. 70
<b>Figure 7.10</b> Modèles interpolés présentés avec effet d'ombrage avant et après l'enrichissement.	p. 71
<b>Figure 7.11</b> Cas extrême de triangles plats.	p. 71
<b>Figure 7.12</b> Modèle produit dans <i>Surfer</i> avec interpolation par krigeage à partir d'un fichier non enrichi. [...].	p. 72
<b>Figure 7.13</b> Modèle produit dans <i>Surfer</i> avec interpolation par distance inverse à la puissance 2 à partir d'un fichier non enrichi. [...].	p. 72
<b>Figure 7.14</b> Modèle produit après enrichissement et avec interpolation "aire volée". [...].	p. 73

<b>Figure 8.1</b> Densification du TIN par subdivision des triangles. Est présentée ici, une subdivision de facteur 3 qui engendre 9 triangles enfants ( $3^2$ ).	p. 75
<b>Figure 8.2</b> Matrice de translation.	p. 76
<b>Figure 8.3</b> Matrice de perspective.	p. 76
<b>Figure 8.4</b> Axes de rotation du modèle numérique d'élévation.	p. 76
<b>Figure 8.5</b> Les matrices de rotation X, Y et Z.	p. 77
<b>Figure 8.6</b> Les six séquences de parcours du maillage de triangles en fonction de la position de l'observateur.	p. 78
<b>Figure 8.7</b> Ordre d'affichage des triangles enfants en fonction du type de triangle parent soit "2U" et "2D".	p. 79
<b>Figure 8.8</b> L'illumination d'un triangle se fait en comparant le vecteur normal du triangle nommé N au vecteur de la source lumineuse nommé L dont la position est définie par l'utilisateur.	p. 79
<b>Figure 8.9</b> Ordre séquentiel des opérations réalisées pour le traitement d'un triangle donné à l'intérieur du module de visualisation 3D.	p. 81
<b>Figure 8.10</b> Visualisation d'un TIN représentant un fleuve encastré.	p. 82
<b>Figure 8.11</b> Visualisation d'un TIN représentant une montagne de la région du Lac Killarney au Missouri.	p. 83
<b>Figure 9.1</b> Les divers ordres des branches de l'axe médian sont représentés par une couleur distinctive.	p. 86
<b>Figure 9.2</b> Exemple d'une courbe sur-échantillonnée pour laquelle on devrait réduire le nombre de vertex qui la constituent.	p. 87
<b>Figure 9.3</b> Exemple d'une méthode pouvant être utilisée pour réduire le nombre de vertex sur une courbe.	p. 88

**Figure 9.4**

p. 89

Les segments “anti-croûte” indiquent l’orientation de la pente du terrain et peuvent, par conséquent, servir à la confection d’un modèle d’écoulement.

**Graphique 7.1**

p. 63

Variation du temps de traitement et du nombre de vertex déplacés pour chacune des cinq itérations nécessaires à la généralisation complète de la feuille d’érable.

**Graphique 7.2**

p. 63

Variation du temps de traitement et du nombre de vertex déplacés pour chacune des quatre itérations nécessaires à la généralisation complète d’une carte polygonale.

**Graphique 7.3**

p. 65

Variation du temps de traitement et du nombre de vertex déplacés pour chacune des six itérations nécessaires à la généralisation complète d’un fichier de courbes de niveau.

**Tableau 2.1**

p. 7

Implantation de la structure Quad-Edge en Pascal Orienté-Objet (Delphi).

**Tableau 2.2**

p. 12

Description de l’algorithme *Walk* permettant, à partir de n’importe quel edge, de trouver le triangle contenant un point P. [...].

**Tableau 4.1**

p. 42

La structure topologique *Arc/Node*.

## 1. INTRODUCTION

Dans le domaine des sciences géomatiques et plus spécifiquement celui de la cartographie informatisée, les champs d'application sont nombreux, notamment en raison de la diversité croissante des données disponibles. À titre d'exemple, citons les données forestières, les données socio-démographiques, les levés géodésiques, les semis de points par captage photographique, etc. Tout ces types de données sont traités afin d'en extraire de l'information pertinente pour son utilisateur. Leur traitement et leur intégration se feront à l'intérieur d'un système d'information géographique communément désigné par l'acronyme SIG.

À l'intérieur d'un tel système, la réalité est modélisée sous forme d'entité géométrique. En effet, l'espace ne peut être représenté que s'il est fragmenté en morceaux individuellement manipulables. Par exemple, les éléments surfaciques de la réalité, tels les lacs, les lots et les peuplements forestiers, seront représentés dans le système par un polygone, soit une entité géométrique à deux dimensions. Par ailleurs, les éléments linéaires, comme les routes, les rivières et les frontières, seront représentés par des lignes, soit une entité à une dimension. Quant aux éléments ponctuels, telle une bouche d'égout, ils sont représentés par un point. Toutefois, rappelons que la géométrie d'un objet est également assujettie à son échelle de représentation. Ainsi, un bâtiment sera modélisé, suivant sa taille, sous forme d'un polygone à l'échelle du 1:5000, mais sous la forme d'un point à l'échelle du 1:50 000.

L'intégration de données topographiques à l'intérieur d'un SIG requiert généralement la conception d'un modèle numérique d'élévation. Le modèle numérique, qu'il soit sous forme d'une grille ou d'un réseau de triangles, permet, comme son nom l'indique, de modéliser la composante Z associée à toute coordonnée X,Y. Ce type de modèle fournit une représentation de la réalité terrain, "utilisable" à l'intérieur d'un tel système, contrairement à celle offerte par les courbes de niveau. Nous présentons les structures et les types de données utilisés pour la modélisation du terrain à la section 5. Également, un module de visualisation 3D a été développé dans le cadre de cette recherche et il est présenté à la section 8.

Les données initiales qui alimentent habituellement les SIG vectoriels sont de deux types: les courbes et les points. Les polygones n'en faisant pas partie car ils sont eux-mêmes constitués

par des courbes. En fait, si l'on souhaite intégrer dans le système de l'information relative aux peuplements forestiers, on devra numériser ou vectoriser chacune des limites des peuplements, soit chacune des courbes définissant la frontière des polygones forestiers. L'intégration de l'information spatiale se fait par un processus de numérisation lorsque la donnée est sous forme analogique (carte papier) ou par vectorisation lorsque la donnée est sous forme matricielle (carte balayée) et que l'on souhaite la convertir sous forme vectorielle. Le format vectoriel est le plus utilisé car il permet la représentation d'objet par une série de coordonnées reliées les unes aux autres, conférant ainsi une précision géométrique très grande aux entités cartographiques. Contrairement au format vectoriel, le format matriciel est plus gourmand en espace mémoire car il représente le terrain de façon continue. En réalité, ce ne sont pas seulement les objets d'intérêt qui sont modélisés, mais également l'espace entre chacun d'eux.

Lorsque l'on désire utiliser un document cartographique numérisé à une grande échelle pour une application où il sera manipulé à une échelle plus réduite, il peut s'avérer nécessaire, si la différence d'échelle est grande, de généraliser le document. La généralisation permettant ainsi d'éliminer l'information devenue superflue à cette nouvelle échelle. Divers algorithmes de généralisation existent, mais tous n'obtiennent pas de bons résultats lorsqu'il s'agit de préserver l'intégrité topologique des objets. L'algorithme de Douglas-Peucker (1973) est couramment employé pour la généralisation des lignes, mais il produit parfois le chevauchement de lignes lorsque celles-ci sont trop proches les unes des autres. Nous présentons un nouvel algorithme de généralisation à la section 3 dont les caractéristiques sont, entre autres, d'assurer le maintien de la topologie et de ne produire aucun chevauchement de lignes.

La reconstruction de courbes connectées à partir de données non structurées est un problème qui a monopolisé plusieurs chercheurs au cours des dernières années. En d'autres mots, comment procéder pour reconstruire une courbe à partir d'un ensemble de points non structuré mais suffisamment échantillonné. Amenta et al. (1998) ont montré que cela est possible et ils ont défini le seuil d'échantillonnage requis entre les points pour permettre la construction de la courbe qu'ils nomment croûte. Pour établir ce seuil, l'axe médian à l'ensemble des points fut utilisé. Cet axe est un descripteur de forme et a été étudié pour la première fois par Blum (1967). La section 2 présente quelques-unes des notions théoriques nécessaires à la bonne compréhension de ce mémoire de recherche dont le diagramme de Voronoï, la triangulation

Delaunay, la croûte et l'axe médian.

Deux ans plus tôt, Gold et al. (1996) avaient également travaillé sur la reconstruction des courbes à partir de points étiquetés mais non structurés. Dans leur cas, il s'agissait de reconstruire les frontières des polygones forestiers à partir d'un ensemble de points qui avaient été saisis de part et d'autre des frontières. Les points étaient par la suite étiquetés et on ne retenait que la frontière (segments des cellules Voronoï) passant entre les points dont la valeur de l'étiquette était différente. Par après, Gold (1999) a repris et perfectionné la méthode d'Amenta et al. (1998). Il a développé un algorithme qui permet, au cours d'une seule opération, de construire les courbes d'un ensemble de points, mais aussi d'extraire l'axe médian à ces points. Cet algorithme, qui porte le nom de *One-Step Boundary and Skeleton Extraction Algorithm* a ensuite été implanté informatiquement par l'auteur. La section 4 présente quelques applications utilisant les courbes et l'axe médian.

L'extraction des courbes, à partir d'un ensemble non structuré de points, est le problème de base qui a fait naître l'idée de cette recherche. À partir de l'algorithme de Gold (1999), deux applications ont été développées par l'auteur afin d'apporter une solution à certains problèmes fréquemment rencontrés lors du traitement, de la modélisation et de la manipulation des données spatiales. Le développement des applications fut basé sur les hypothèses de travail suivantes:

1. La généralisation de l'axe médian permet de réduire la complexité graphique de la croûte, soit la frontière de l'objet. Ceci est basé sur le constat que les objets simples possèdent un axe médian simple.
2. L'insertion des vertex de l'axe médian permet de solutionner le problème des triangles plats, provoqué par l'emploi de courbes de niveau lors de la confection d'un modèle numérique d'élévation (TIN).

En regard de la première hypothèse, nous avons développé un algorithme de généralisation des courbes qui fonctionne par rétraction des branches de l'axe médian. L'algorithme peut traiter aussi bien les courbes ouvertes que fermées. Il n'est aucunement affecté par la densité des

objets sur le plan et, contrairement à la plupart des algorithmes traditionnels actuellement utilisés, il préserve les relations topologiques des objets et n'utilise aucune tolérance. La section 3 est consacrée entièrement à la présentation et à la description de ce nouvel algorithme de généralisation.

En s'appuyant sur la seconde hypothèse, nous avons mis au point une technique d'enrichissement pour modèle numérique de terrain en réponse au problème suivant: la création d'un TIN à partir de courbes de niveau entraîne l'apparition de triangles plats aux endroits de plus forte inflexion sur les courbes. L'interpolation de valeurs d'élévation sur l'axe médian, suivi ensuite de l'insertion de ces valeurs dans le TIN, permet d'éliminer ces triangles indésirables. La section 6 présente cette méthode en détail.

Les résultats obtenus par l'algorithme de généralisation, suite au traitement de divers fichiers sont présentés à la section 7. De même, les modèles numériques d'élévation obtenus après application de la méthode d'enrichissement y sont également présentés. Enfin, nous apportons une évaluation sur l'efficacité de la méthode d'enrichissement en comparant nos résultats à ceux produits par un logiciel commercial d'interpolation.

La dernière section de ce mémoire est consacrée à la présentation de nouvelles avenues de recherche qui mériteront d'être explorées dans l'avenir. Nous y présentons un algorithme permettant de réduire le nombre de points sur une courbe sur-échantillonnée. Également, nous montrons comment il s'avère possible d'utiliser certains segments du diagramme de Voronoï pour concevoir un modèle d'écoulement. Enfin, la conclusion suivra cette section.

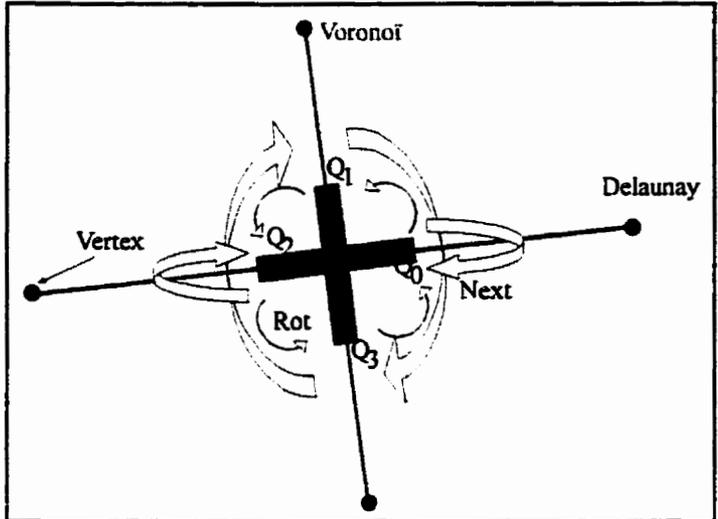
## 2. NOTIONS THÉORIQUES

### 2.1 Structure de données Quad-Edge

Le Quad-Edge est une structure de données spatiales qui a été développée par Guibas et Stolfi (1985). C'est une structure symétrique qui permet de connecter, par une série de pointeurs, la triangulation Delaunay et le diagramme de Voronoï au sein d'une même unité (figure 2.1).

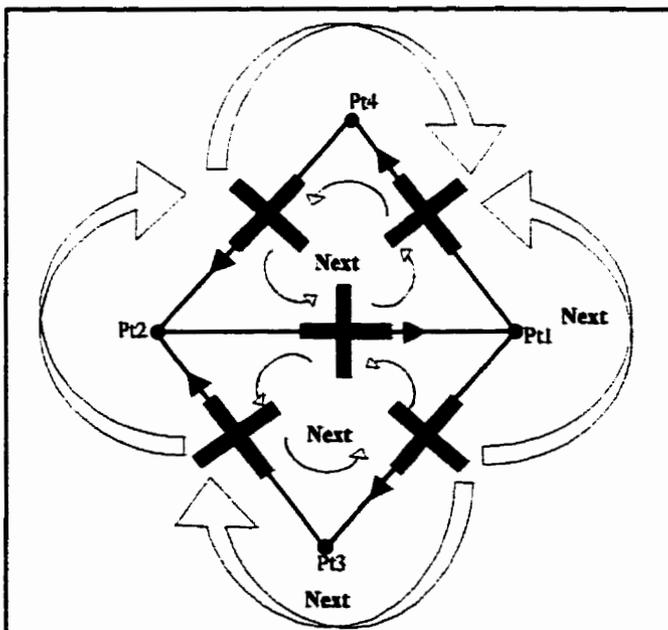
Tel que présenté sur la figure 2.1, le Quad-Edge possède quatre branches ou edges. À l'extrémité de chacune d'elle, existe trois (3) pointeurs permettant soit d'obtenir la coordonnée d'un vertex, soit d'effectuer un

Figure 2.1



Le Quad-Edge permet de connecter, par une série de pointeurs, la triangulation Delaunay (primal) et le diagramme de Voronoï (dual) au sein d'une même unité.

Figure 2.2



Structure Quad-Edge pour deux triangles.

déplacement quelconque dans un Quad-Edge (Rot) ou d'un Quad-Edge vers un autre (Next). La figure 2.2 présente la structure Quad-Edge pour 2 triangles Delaunay, mais sans le Voronoï.

Le Quad-Edge possède de nombreux avantages qui en font un candidat d'excellence pour créer la topologie. En outre, il permet les déplacements entre la triangulation Delaunay et le diagramme de Voronoï de façon transparente. Ainsi, il devient possible d'obtenir réponse aux questions suivantes: trouver le ou les

voisin(s) d'une entité sélectionnée, trouver le point le plus proche d'un point spécifié, naviguer dans le graphe selon une direction particulière, etc. De plus, cette structure se montre très adaptée à la programmation Orientée Objet.

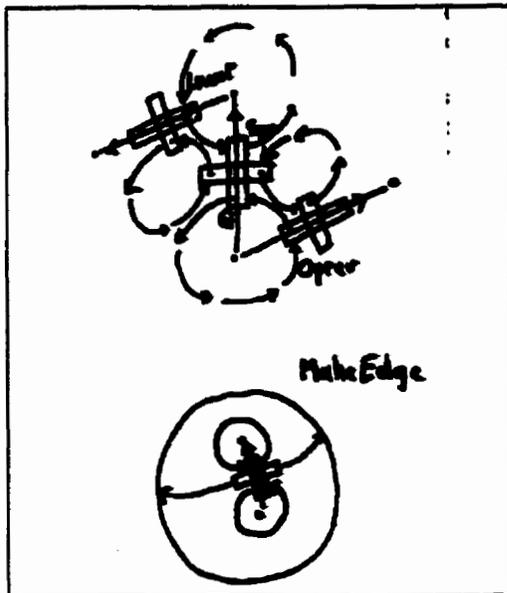
La structure du Quad-Edge en Pascal Orienté-Objet (Delphi) est la suivante.

```
TQuad = class
  N : TQuad; (pour Next)
  R : TQuad; (pour Rot)
  V : TPointLL; (pour Vertex)
End;
```

L'opérateur **N** permet de se déplacer autour d'un vertex ou d'une face en sens anti-horaire. **R** permet de passer d'une branche à l'autre d'un Quad-Edge en sens anti-horaire. Enfin, **V** correspond à la coordonnée vers laquelle pointe l'edge. Il existe aussi cinq autres opérateurs qui sont **Sym**, **Oprev**, **Lnext**, **Onext** et **Rprev**. Cependant, ils ne sont pas en soit des opérateurs primaires car ils sont constitués d'une succession d'opérateurs **N** et **R**. Par exemple, **Sym** permet d'obtenir le Quad opposé d'un Quad donné; il s'agit de deux commandes **R** successives

(**Sym** = **Quad.R.R**). **Oprev** permet de se déplacer au edge suivant en sens horaire autour du vertex associé avec la branche originale du Quad-Edge; il équivaut à **Quad.R.N.R**. **Lnext** permet de se déplacer au edge suivant en sens horaire autour du vertex associé avec la branche opposée du Quad-Edge; il équivaut à **Quad.R.R.R.N.R**, mais aussi à **Quad.Sym.Oprev**. **Onext** permet d'aller au edge situé en sens anti-horaire par rapport au vertex d'origine. **Rprev** permet de se déplacer au edge situé en sens anti-horaire, mais par rapport au vertex opposé d'**Onext** (**Rprev** = **Sym.Onext**).

Figure 2.3



Création d'un edge avec **MakeEdge**.  
Source: Gold, 1998a

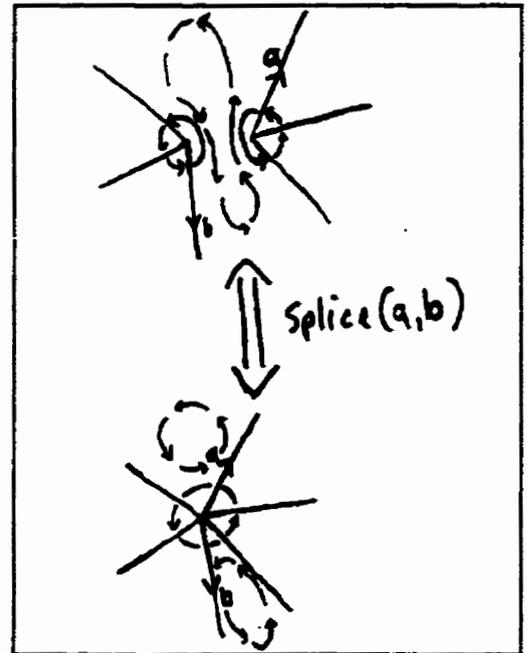
Nous avons présenté les opérateurs permettant de



rétablir la connexion à l'intérieur des triangles ainsi reliés. **Figure 2.4**

Il existe également deux autres méthodes utilisant Splice, ce sont **Delete** et **Swap**. La première permet de déconnecter et de détruire un edge. La seconde permet de changer la diagonale dans une triangulation et de réassigner les pointeurs modifiés afin de rétablir la connexion entre les Quad-Edges.

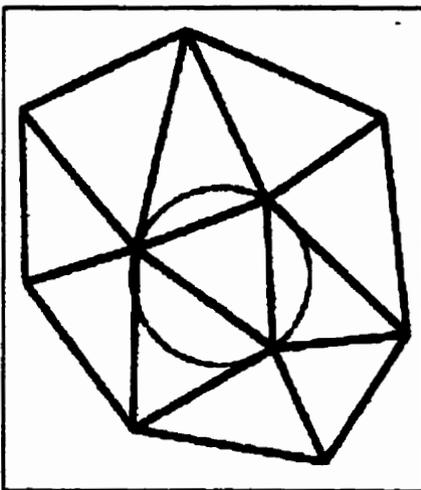
Le tableau 2.1 présente l'implantation en langage Delphi de la structure Quad-Edge. Bien que les avantages associés à l'utilisation de la structure Quad-Edge soient nombreux, celle-ci présente toutefois l'inconvénient d'être très gourmande en espace mémoire, en raison du nombre important de pointeurs utilisés.



Connexion et déconnexion de deux Quad-Edges avec Splice.  
Source: Gold, 1998a

## 2.2 La triangulation Delaunay et le diagramme de Voronoï

**Figure 2.5**



Un triangle Delaunay a comme caractéristique de posséder un cercle circonscrit vide de tout autre site.  
Source: Van Kreveld, 1997

### 2.2.1 Définition et propriétés

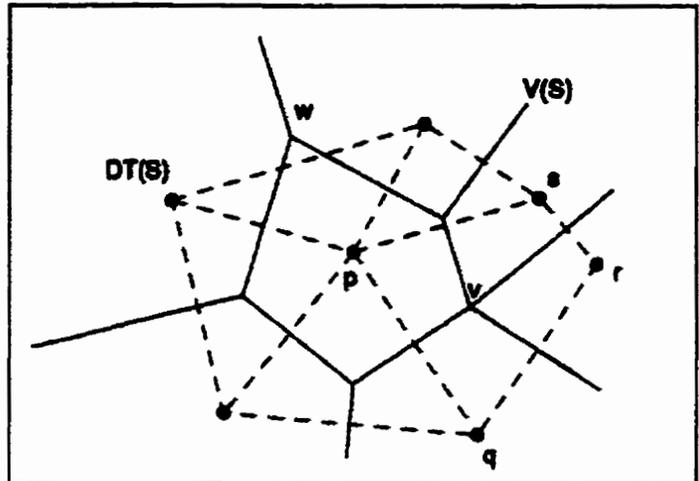
La triangulation de Delaunay est une partition de l'espace en triangles dont les sommets sont les sites. Elle doit son nom à B. Delaunay qui développa ce type de triangles vérifiant la propriété suivante et qui définit cette triangulation de façon unique (figure 2.5).

**Propriété :** Les cercles circonscrits aux triangles sont vides de tout autre site.

Ce type de triangle a l'avantage d'être visuellement esthétique.

tique, en ce sens que la longueur de ses arêtes est minimisée lors de sa construction. Par conséquent, son utilisation pour des applications de modélisation de terrain est particulièrement appréciée. Pour plus d'information sur la construction d'une triangulation Delaunay, le lecteur est invité à consulter l'ouvrage de Peucker et al. (1978).

**Figure 2.6**



La triangulation de Delaunay (DT) est en relation voisine avec le diagramme de Voronoï (V) puisqu'elle constitue son graphe dual.

Source: Aurenhammer et al., 1996

La triangulation de Delaunay est en relation voisine avec le diagramme de Voronoï puisqu'elle constitue son

graphe dual. En effet, pour chaque vertex Voronoï correspond un triangle Delaunay. De même, les générateurs des cellules Voronoï correspondent aux sommets des triangles Delaunay et pour chaque segment Voronoï existe un segment Delaunay orthogonale associé. Ainsi, si nous utilisons le diagramme de Voronoï comme base, nous pouvons construire la triangulation de Delaunay en reliant les sites, soit les générateurs des cellules Voronoï (figure 2.6).

En 1850, un mathématicien nommé S.G. Dirichlet a commencé à s'intéresser à ce type de construction, d'où l'appellation de "mosaïque de Dirichlet", souvent attribuée au diagramme de Voronoï. Mais c'est le nom de M.G. Voronoï, célèbre mathématicien du 20<sup>ème</sup> siècle, qui fut donné à ce diagramme en raison du fait qu'il a défini cette construction et l'a exploitée en profondeur.

Dû aux larges champs d'applications qui l'utilisent, le diagramme de Voronoï porte plusieurs noms dans la littérature, souvent associés à la personne qui s'en est servi pour la première fois. Ainsi, dans le domaine de la géographie, on le nomme "polygones de Thiessen", en l'honneur du climatologue A.H. Thiessen qui, en 1911, délimita chaque station météorologique par un polygone (cellule Voronoï) afin d'améliorer l'estimation des moyennes des précipitations sur de larges surfaces (Aurenhammer, 1991). En sciences naturelles, Blum (1967) utilisa cette structure sous le nom de "transformation de Blum" pour la conception d'un descripteur de

forme. Comme nous le verrons à la section 2.3, Blum (1967) utilisa un sous-ensemble du diagramme de Voronoï (axe médian) pour construire un modèle permettant la définition simplifiée des formes.

Le diagramme de Voronoï est une structure géométrique permettant de résoudre des questions à caractère spatial, telles la proximité et la contiguïté. À propos des propriétés mathématiques particulières et uniques des diagrammes de Voronoï, Aurenhammer (1991) mentionne :

*“This has led several authors to believe that the Voronoi diagram is one of the most fundamental constructs defined by a discrete set of points. Finally, Voronoi diagrams have proved to be a powerful tool in solving seemingly unrelated computational problems and therefore have increasingly attracted the attention of computer scientists in the last few years.”*

Le diagramme de Voronoï se définit comme suit.  $E^d$  désigne l'espace euclidien de dimension  $d$ , et  $\delta$  la distance euclidienne. Soit  $S$  l'ensemble des  $n$  points (appelé sites) de  $E^2$ . Pour chaque site  $p$  la cellule Voronoï  $Vor(p)$  de  $p$  est l'ensemble des points de  $E^2$  qui sont plus proches de  $p$  que de tous les autres sites de  $S$ .

$$Vor(p) = \{x \in R^2, \forall q \in S - \{p\} / \delta(x,p) < \delta(x,q)\}.$$

La droite, qui découpe le plan, sépare tous les points les plus proches du générateur  $p$  à ceux plus proches de  $q$ . Elle est également définie comme la bissectrice du segment Delaunay reliant  $p$  à  $q$  en raison de la dualité Voronoï/Delaunay. Ainsi, les polygones créés par cette partition de l'espace forment le diagramme de Voronoï, noté  $V(S)$ , d'un ensemble  $S$  fini de points générateurs. La figure 2.7 donne un exemple du diagramme de Voronoï pour un ensemble de onze (11) générateurs sur le plan.

**Figure 2.7**

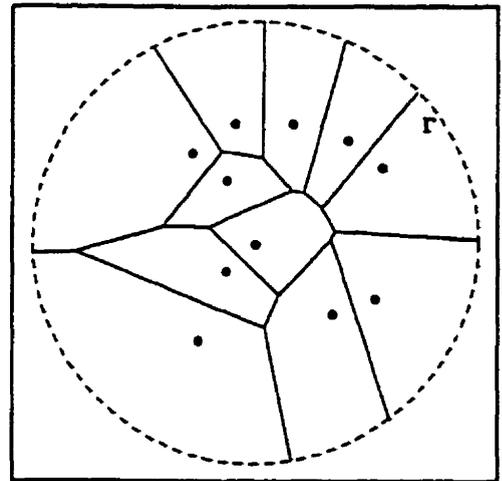


Diagramme de Voronoï pour un ensemble de onze (11) générateurs sur le plan.  
Source: Aurenhammer et al., 1996

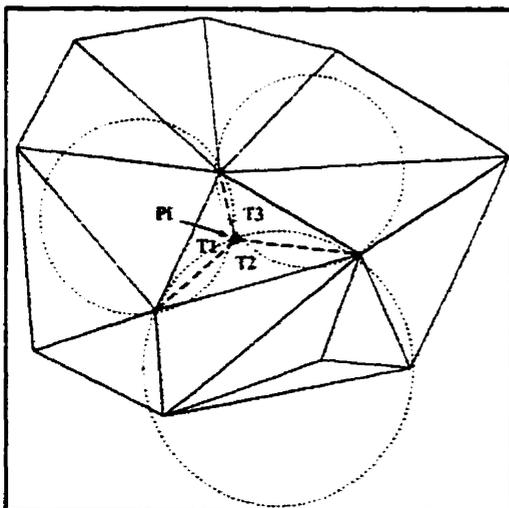
Chaque nœud du diagramme de Voronoï est défini comme étant le centre du cercle dont trois (3) points générateurs  $y$  sont cocirculaires. Ces cercles sont les cercles circonscrits de la triangulation Delaunay des  $n$  points générateurs et c'est pour cette raison que le diagramme de Voronoï est dit le dual de la triangulation Delaunay. Une droite séparant  $p$  et  $q$  sur le plan ne peut être créée que si les cercles centrés à  $p$  et  $q$  possèdent chacun trois points cocirculaires.

Pour plus d'information sur le diagramme de Voronoï, le lecteur est invité à consulter l'ouvrage d'Aurenhammer (1991) de même que celui d'Aurenhammer et al. (1996).

### 2.2.2 Algorithmes

Il existe dans la littérature scientifique plusieurs algorithmes pour la construction du diagramme Voronoï et de son dual, mais nous en aborderons seulement deux dans le cadre de cette recherche. Le premier algorithme utilise une approche de type incrémental alors que le second emploie l'approche "diviser pour régner". Le lecteur désirant connaître d'autres algorithmes pour la construction du diagramme de Voronoï est invité à consulter le rapport technique d'Aurenhammer et al. (1996). Les algorithmes seront présentés en considérant que la structure de base pour la construction est la triangulation Delaunay; le diagramme de Voronoï pourra ensuite être produit très facilement par après.

**Figure 2.8**



Procédure d'insertion d'un point dans la triangulation Delaunay.

La méthode incrémentale a l'avantage d'être simple et efficace. Elle présente une complexité moyenne linéaire de  $O(n)$  pour l'insertion de chaque point. Elle fut présentée pour la première fois par Green et Sibson (1978). Débutons par une triangulation Delaunay de  $n$  points initiaux dans lequel nous désirons ajouter un nouveau point nommé  $P_i$ . La première étape consiste à trouver le triangle  $T$  contenant  $P_i$ . Pour ce faire, nous utilisons un algorithme nommé *Walk* qui permet de rechercher dans la triangulation le triangle contenant  $P_i$  et ce, à partir de

n'importe quel emplacement dans le maillage. Cet algorithme de recherche utilise les propriétés des déterminants. On débute avec un segment donné de la triangulation dont les coordonnées des vertex ( $X1, Y1$  et  $X2, Y2$ ) sont placées dans une matrice  $3 \times 3$  suivant un ordre de lecture qui est anti-horaire à l'intérieur du triangle. La coordonnée de  $P_i$  soit  $X3, Y3$  est également placée dans cette matrice. On obtient alors une matrice de la forme suivante:

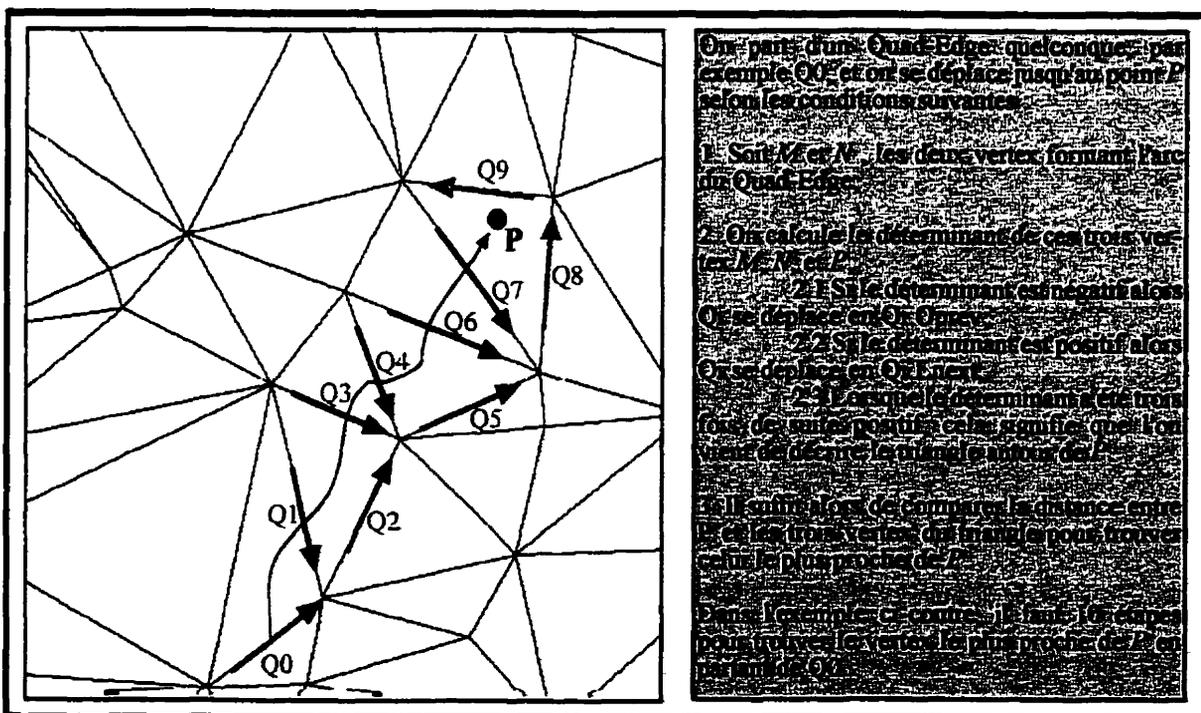
$$\begin{vmatrix} X1, Y1, 1 \\ X2, Y2, 1 \\ X3, Y3, 1 \end{vmatrix}$$

Il suffit ensuite de calculer le déterminant de cette matrice. S'il est positif, alors on poursuit avec le traitement du prochain segment du présent triangle selon un sens anti-horaire. Par contre, si le déterminant

obtenu est négatif, on se déplace alors à l'intérieur du triangle adjacent au segment traité. Ce n'est que lorsque le déterminant aura été positif trois fois de suite que le triangle contenant  $P_i$  aura été trouvé. Cet algorithme présente une complexité de  $O(\sqrt{n})$  par point et de  $n\sqrt{n}$  au total. Le tableau 2.2 décrit le fonctionnement de l'algorithme *Walk* selon la terminologie Quad-Edge

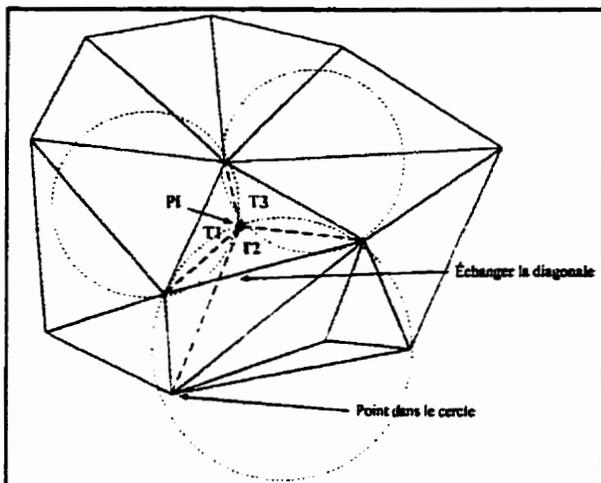
Une fois le triangle trouvé, on scinde celui-ci en trois nouveaux triangles nommés T1, T2 et T3 dont  $P_i$  est le sommet commun au trois (figure 2.8). L'opération suivante consiste à tester si T1,

Tableau 2.2



Description de l'algorithme *Walk* permettant, à partir de n'importe quel edge, de trouver le triangle contenant un point  $P$ . Les flèches rouges indiquent les edges pour lequel le déterminant était négatif, alors que les flèches bleues indiquent ceux pour lequel le déterminant calculé était positif.

Figure 2.9



Procédure d'inversion de la diagonale d'un triangle suite à l'insertion d'un point.

$$\begin{vmatrix} X1, Y1, (X1^2+Y1^2), 1 \\ X2, Y2, (X2^2+Y2^2), 1 \\ X3, Y3, (X3^2+Y3^2), 1 \\ XP, YP, (XP^2+YP^2), 1 \end{vmatrix}$$

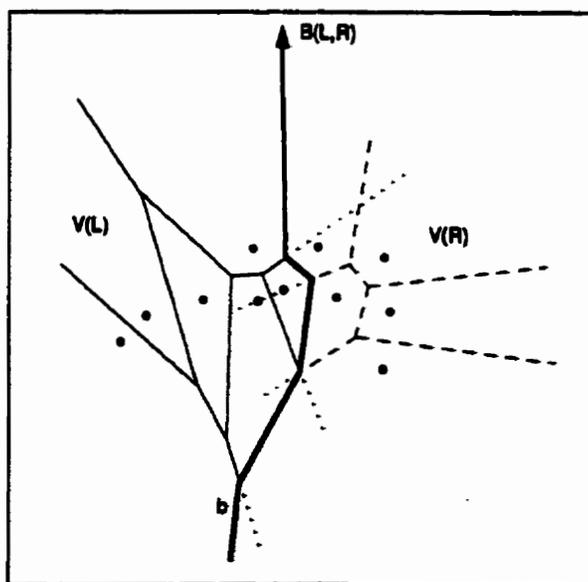
Si le déterminant de cette matrice est supérieur à zéro, alors le cercle est vide et on ne change rien. Par contre, si le déterminant est inférieur à zéro, le point  $P_i$  est dans le cercle et, par conséquent, on inverse la diagonale du triangle concerné avec celui de son voisin (figure 2.9). L'algorithme

utilisé pour inverser la diagonale se nomme *Swap*. On répète l'opération jusqu'à ce que tous les triangles voisins aient leur cercle vide.

Cette méthode est facile à implanter et est relativement efficace tant et aussi longtemps que le nombre de point dans le maillage n'est pas trop élevé. En effet, plus le maillage est dense, plus le temps mis par *Walk* pour atteindre le triangle en question est long. Mentionnons également que cette méthode procède par mise à jour locale de la triangulation, ce qui signifie que l'ajout ou l'effacement d'un point ne requiert pas une reconstruction complète de la triangulation. Elle est donc particulièrement adaptée pour une utilisation conjointe avec la

$T_2$ ,  $T_3$  et leurs voisins constituent des triangles de type Delaunay, c'est-à-dire que leurs cercles circonscrits sont vides. On définit donc un cercle pour chacun de ces triangles et on évalue si chacun des cercles est vide à l'aide d'une fonction qui utilise une matrice  $4 \times 4$ . À partir des 3 paires de coordonnées  $[X_1, Y_1; X_2, Y_2; X_3, Y_3]$  provenant du triangle obtenu suivant un sens de lecture anti-horaire et le point  $P_i [X_P, Y_P]$  qui a été inséré, le contenu de la matrice est le suivant:

Figure 2.10



L'approche "Diviser pour régner" pour la construction d'un diagramme de Voronoï.

Source: Aurenhammer et al., 1996

structure de données Quad-Edge.

L'approche "diviser pour régner" est bien différente. Elle fut proposée pour la première fois par Shamos et Hoey (1975). Elle vise à fragmenter un problème complexe en sous-problème jusqu'à obtenir des entités facilement solvables et de rassembler chacune des solutions obtenues pour constituer le résultat final. Cette approche permet de concevoir des algorithmes efficaces et rapides avec le niveau de complexité le plus bas,  $O(n \log n)$  dans le pire des cas. Aurenhammer (1991) souligne: "*Divide and conquer is the basis of a large class of algorithms for computing generalized Voronoi diagrams in the plane*". Concrètement, la procédure consiste d'abord à diviser la triangulation Delaunay en deux sous-ensembles nommés respectivement  $V(L)$  et  $V(R)$  et séparés par une ligne verticale nommée  $B(L,R)$  (figure 2.10). Par la suite, on trie les coordonnées des points générateurs en  $X$  croissant avec lesquelles on crée deux listes nommées  $LV(L)$  et  $LV(R)$ . Le tri peut être réalisé par un algorithme comme le *quick sort* ou le *heap sort*. La première liste contient les vertex triés du sous-ensemble  $V(L)$  et l'autre, les vertex de  $V(R)$ . On débute par  $LV(L)$ . Si elle contient un nombre inférieur ou égal à trois générateurs, on génère le diagramme Voronoï nommé  $V_i$  de ces trois vertex. Par contre, si elle contient plus de trois vertex, la liste est subdivisée à nouveau en deux sous-listes. On traite ensuite chacune de ces nouvelles listes de générateurs. Cet algorithme fonctionne donc de manière récursive. Une fois tous les sous-diagrammes Voronoï construits, ils sont assemblés pour former le diagramme résultant.

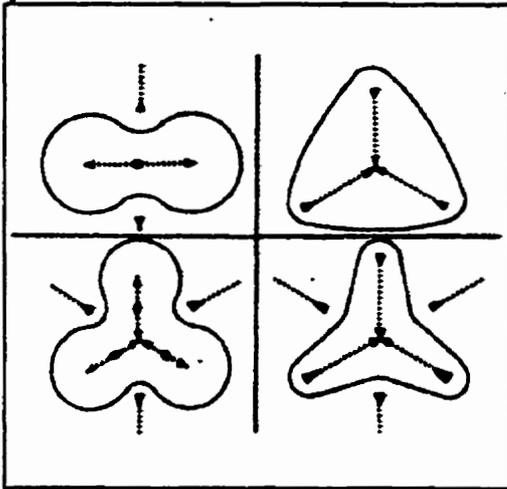
## 2.3 L'axe médian et la croûte

### 2.3.1 L'axe médian

L'axe médian, communément appelé squelette, est une structure intimement liée à la croûte qui est définie en détail à la section 2.3.2. Cette structure a fait l'objet de nombreuses recherches dans le domaine de la géométrie algorithmique, notamment pour la reconnaissance et la description de formes (Blum, 1967; Ogniewicz, 1994), la reconstruction de courbe (Amenta et al., 1998), la généralisation cartographique (Gold et al., 1999b), la modélisation du terrain (Thibault et al., 1999), etc. Débutons d'abord en présentant la définition et les propriétés de l'axe médian.

L'axe médian est une structure géométrique, qui fournit une description différente de la forme d'un objet, à la fois simplifiée mais sans perte réelle d'information morphologique. L'axe médian est une représentation de dimension D-1 d'un

**Figure 2.11**



Exemples d'axe médian pour des formes simples.

Source: Blum, 1967

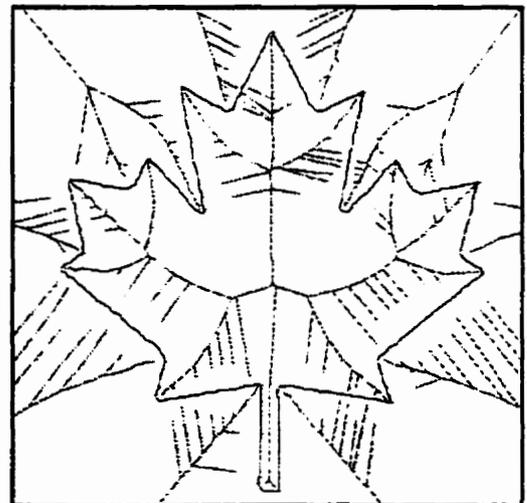
objet de deux dimensions. Un objet surfacique possède donc un axe médian linéaire à une dimension. Un cas fait cependant exception, celui du cercle, dont l'axe médian est un point.

En raison de ses caractéristiques uniques, l'axe médian est fréquemment utilisé en robotique pour la reconnaissance des formes, mais aussi en reconnaissance optique de caractères par les logiciels d'OCR (*Optical Characters Recognition*).

La figure 2.11 donne un exemple de l'axe médian pour des formes simples. Pour donner un exemple concret de ce descripteur de forme, l'axe médian peut être interprété comme les nervures d'une feuille végétale (figure 2.12).

Mais d'un point de vue mathématique, quelle définition donner à l'axe médian? En voici une. Les vertex qui constituent l'axe médian sont les centres des cercles circonscrits de la triangulation Delaunay. Ceci présuppose que les vertex constituant l'objet ont été triangulés. Chaque vertex de l'axe médian est donc situé à une position médiane par rapport à trois points (cocirculaires) composant l'enveloppe de l'objet.

**Figure 2.12**



Axe médian d'une feuille d'érable.

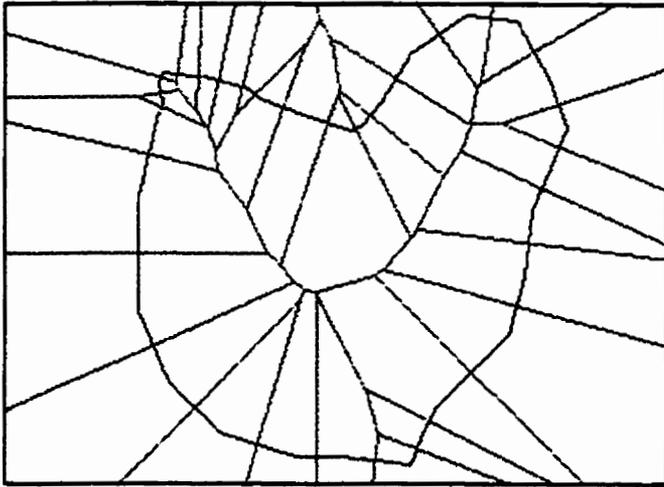
*"The MAF (medial axis function) can be considered a symmetrical central description of the space whose boundary is the stimulus contour (Blum, 1967)"*.

Cette implication des cercles circonscrits de la triangulation Delaunay dans la définition de l'axe médian nous permet de définir le théorème suivant :

**Théorème:** L'axe médian est un sous-ensemble du diagramme de Voronoï.

En fait, la relation entre l'axe médian et le diagramme de Voronoï est la suivante: pour isoler l'axe médian, on ne retient que les segments du diagramme de Voronoï qui ne croisent pas les

**Figure 2.13**



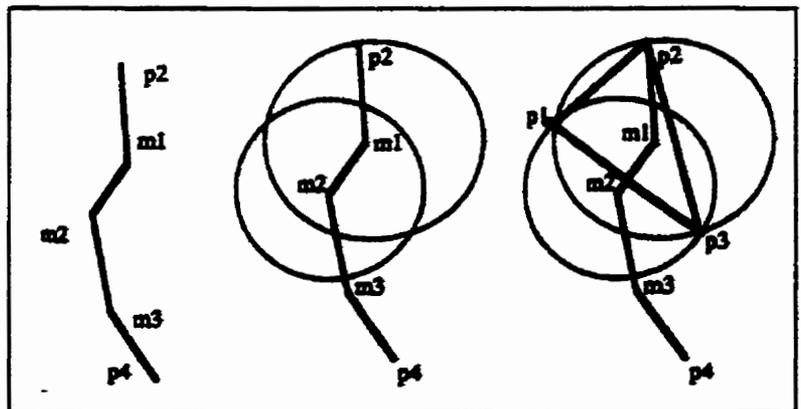
L'axe médian est constitué des segments du diagramme de Voronoï qui ne croisent pas l'enveloppe de l'objet ou croûte.

segments de la triangulation Delaunay constituant l'enveloppe de l'objet ou croûte. Sur la figure 2.13, l'enveloppe de l'objet est représentée en bleu et les segments du diagramme de Voronoï appartenant à l'axe médian en rouge. Dans le cas d'objets convexes, il y aura présence de triangles à l'extérieur de l'objet, d'où formation d'un axe médian externe ou exo-squelette. Les objets dont l'enveloppe est concave ne posséderont qu'un axe médian interne ou endo-squelette.

La construction de l'axe médian d'un objet est une opération réversible en soit, c'est-à-dire qu'en possédant uniquement l'axe médian comme source d'information, il est possible de reconstruire l'objet dans son

intégralité. Pour ce faire, il faut **Figure 2.14**

stocker, lors de la construction de l'axe médian, le rayon du cercle circonscrit à chacun des nœuds de l'axe médian. On pourra par après reconstruire l'objet en générant les cercles dont les intersections constitueront alors les vertex de l'objet (figure 2.14).

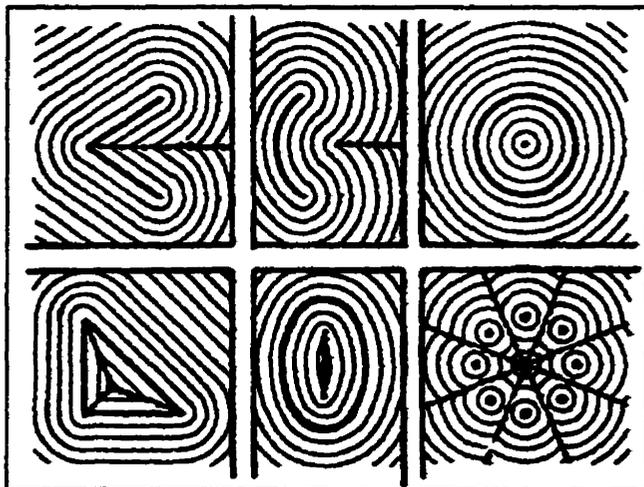


Procédure permettant de reconstruire l'enveloppe de l'objet ou croûte à partir de l'axe médian.

Source: Chithambaram et al., 1991

*“Since the contour and the MAF description are equivalent and invertible there is no information lost or gained in the transformation.”(Blum, 1967).*

**Figure 2.15**



La propagation uniforme de l'excitation sur le plan se fait sous forme de vagues qui ne se recoupent pas.  
Source: Blum, 1967

De nombreux chercheurs ont utilisé l'axe médian comme descripteur de forme, notamment Blum (1967) et Ogniewicz (1994) pour ne nommer que ceux-ci. Blum (1967) a été celui qui a défini l'axe médian, c'est lui qui en a jeté les bases. Il considérait que les modèles mathématiques disponibles à l'époque étaient incapables de définir correctement les formes des objets, en raison de leur nombre et de leur

variété dans la nature.

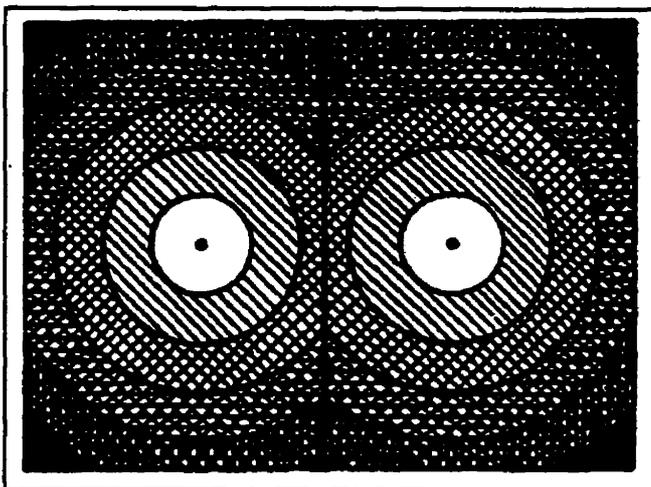
*“[...] , it is necessary to develop some way of obtaining shape properties without becoming enmeshed in the huge combinatories of simple congruence views”(Blum, 1967).*

Voici la méthode utilisée par Blum (1967) pour construire l'axe médian des objets. Partons du principe que le plan dans lequel se situe l'objet est isotrope et continu. À chacun des points placés sur le plan, on définit les trois propriétés suivantes : l'excitation, la propagation et le temps mort. D'abord, un point peut être excité ou non; à ceci est rattachée une valeur numérique de 0 ou 1. Deuxièmement, un point excité stimule un point adjacent selon un délai proportionnel à la distance, ceci est la propagation. Enfin, une fois un point excité, celui-ci n'est plus affecté, pour une période de temps déterminé, à toute autre excitation.

Un stimulus visuel est d'abord déclenché sur l'enveloppe de l'objet. Celui-ci atteint certains points et les excite, ce qui, par une série de réactions en chaîne, propage l'excitation uniformément sur le plan dans toutes les directions, sous la forme de vagues qui ne se recoupent pas. Chacune de ces vagues représente une figure parallèle à l'original. La figure 2.15 donne un

exemple du résultat obtenu. La propagation d'un "feu de brousse" est l'analogie qui nous permet de mieux conceptualiser la définition de l'axe médian (figure 2.16). Imaginons que le pourtour d'un objet est en feu et que ce feu progresse vers l'intérieur de l'objet à vitesse constante, là où les flammes se rencontreront marquent l'emplacement de l'axe médian. Notons que lorsque la forme de l'axe médian est celle d'une ligne (figure 2.16), l'objet en question est symétrique. Pour plus de détail sur cette méthode de construction de l'axe médian, le lecteur est invité à consulter l'excellent ouvrage de Blum (1967).

Figure 2.16



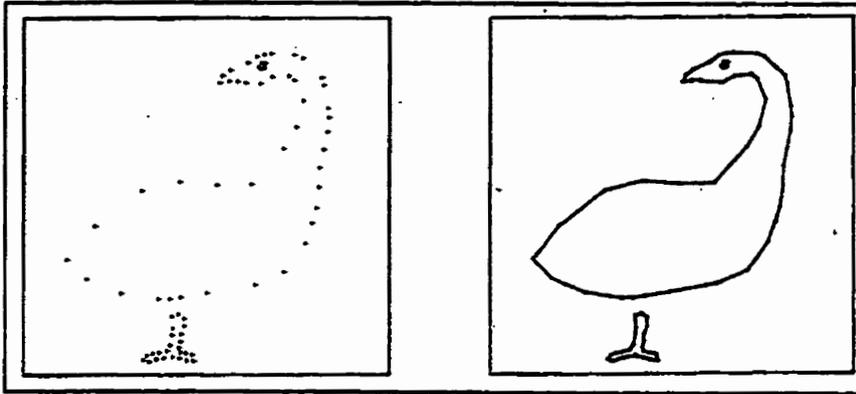
Analogie du "feu de brousse" pour conceptualiser l'axe médian qui est ici représenté par la ligne rouge.  
Source: Blum, 1967

Ogniewick (1994) a également travaillé avec l'axe médian. Brièvement, il a développé une extension multi-échelle de l'axe médian en combinant de l'information dérivée à partir de différentes représentations de l'objet (*scale-space hierarchy*) et de l'axe médian lui-même. Cette nouvelle version de l'axe médian a l'avantage de pouvoir décrire des objets complexes caractérisés par des frontières très irrégulières et dentelées.

### 2.3.2 La croûte

Le terme de croûte est apparu pour la première fois dans la littérature à l'intérieur d'un article écrit par Amenta et al. (1998). L'auteur y définit la croûte comme une construction polygonale constituée par la connexion de chaque paire de points adjacents à l'intérieur d'un ensemble non structuré de points (figure 2.17).

Pour que la croûte puisse être construite, certaines conditions géométriques doivent être satisfaites. La première condition est relative à la densité de l'échantillonnage. Amenta et al. (1998) ont défini l'indice  $R$  qui représente la distance entre un vertex de la croûte et son vertex voisin

**Figure 2.17**

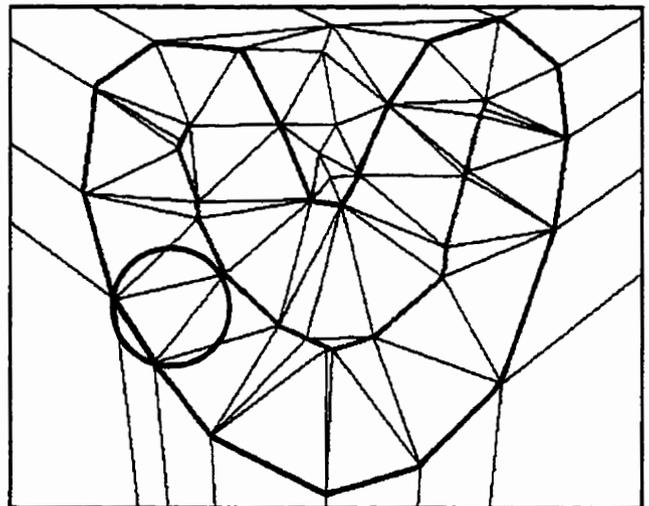
Un ensemble non structuré de points à gauche, versus la croûte constituée par la connexion de chaque paire de points adjacents à droite.  
Source: Amenta et al., 1998

le plus proche sur l'axe médian. Si cet indice est inférieur à 0.40, tous les vertex de la croûte sont connectés par un segment. Par contre, si  $R$  est inférieur à 0.252, la croûte ne contient aucun segment entre deux vertex non adjacents. En faisant varier  $R$ , il est possible

de contrôler la densité des vertex en fonction de la complexité de la courbe. Ainsi, les endroits de la courbe peu détaillés pourront être représentés avec peu de vertex.

En référant au diagramme **Figure 2.18**

Voronoi/Delaunay, Amenta et al. (1998) suggère une autre définition de la croûte tel que présentée sur la figure 2.18. La croûte (trait noir gras) est un ensemble de segments d'une triangulation Delaunay de  $N$  vertex échantillons dont les cercles circonscrits (en bleu) ne contiennent aucun autre vertex échantillon ni vertex de l'axe médian (en rouge). Cette définition est vraie à condition toutefois que la courbe soit suffisamment échantillonnée, c'est-à-dire qu'elle réponde aux critères d'échantillonnage précédemment mentionnés.

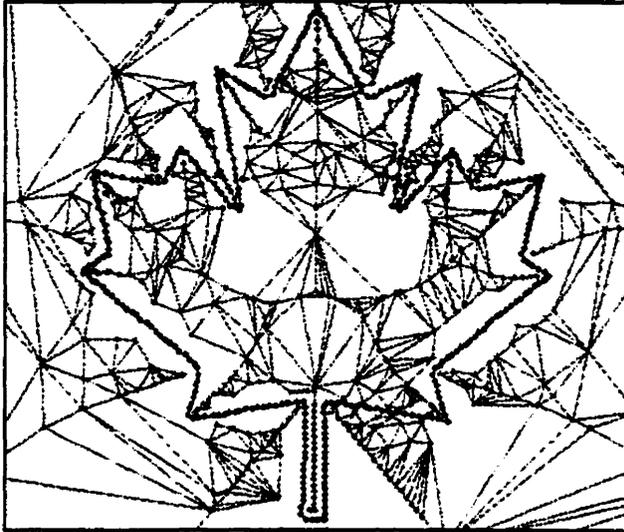


Définition de la croûte d'après Amenta et al. (1998) basée sur le diagramme Voronoi/Delaunay.

### 2.3.3 Algorithmes d'extraction de l'axe médian et/ou de la croûte

#### 2.3.3.1 L'approche d'Amenta et al. (1998)

**Figure 2.19**



Extraction de la croûte avec l'approche d'Amenta et al. (1998) pour une feuille d'érable.

Les travaux réalisés par Amenta et al. (1998) visaient à développer une technique afin d'extraire la croûte d'un maillage de triangles Delaunay. Leurs travaux ont été précurseurs au développement de l'algorithme de Gold (1999) dont il sera question un peu plus loin. La méthode qu'ils ont développée, simple en soit, s'avère très efficace pour isoler la croûte, bien qu'elle ne donne pas de bons résultats pour l'extraction de l'axe médian. La procédure est la suivante: pour un ensemble de points  $S$  dont on a construit la triangulation Delaunay et le diagramme de Voronoï asso-

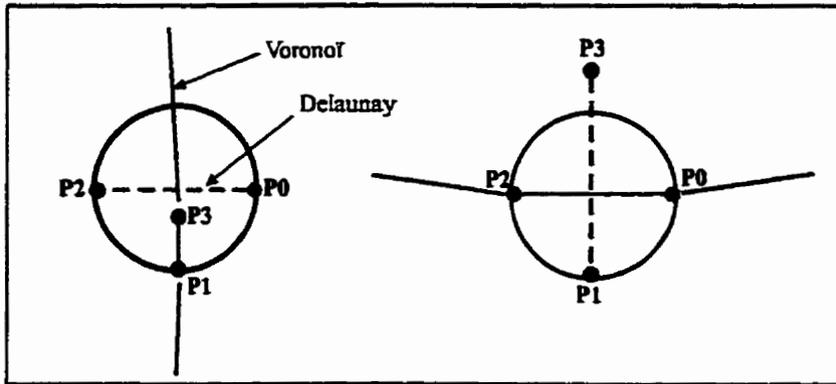
cié, on isole les vertex du diagramme de Voronoï que l'on introduit ensuite dans la triangulation. On obtient ainsi une triangulation enrichie par les vertex Voronoï. Pour extraire la croûte, il suffit, en parcourant la triangulation, de conserver uniquement les segments Delaunay dont les cercles circonscrits sont vides de tout vertex Voronoï et Delaunay. La figure 2.19 donne un exemple d'un fichier de point sur lequel l'algorithme d'Amenta et al. (1998) a été appliqué afin d'en extraire la croûte.

#### 2.3.3.2 L'approche de Gold (1999)

Suite aux travaux d'Amenta et al. (1998) sur l'extraction de la croûte, Gold (1999) a décidé d'adapter l'algorithme développé par Amenta et al. (1998) afin d'extraire, en une seule opération, la croûte mais également l'axe médian. Gold (1999) se sert de la structure de données Quad-Edge pour effectuer les déplacements dans la triangulation, mais une autre structure topologique aurait pu également être utilisée. L'algorithme développé par Gold (1999) se

nomme le *One-Step Crust and Skeleton Extraction* et il a été implanté par l'auteur à l'intérieur de notre programme maison (*Voronoi ToolKit*) comme une fonction appelée *Crust* (croûte). Son fonctionnement est d'autant plus facilité qu'il utilise la structure Quad-Edge qui, comme nous l'avons mentionné à la section 2, unifie le diagramme Voronoï et la triangulation Delaunay au

**Figure 2.20**



L'approche de Gold (1999) pour l'extraction de la croûte et de l'axe médian.

sein d'une même structure.

L'algorithme de Gold (1999) consiste à déterminer, pour un diagramme Voronoï/Delaunay, si un segment donné de la triangulation fait partie de la croûte ou non? Pour ce faire, on définit un cercle

passant par les deux vertex (P0 et P2) d'un segment Delaunay, de même qu'à un troisième vertex nommé P1, situé sur le segment Voronoï associé (dual). La figure 2.20 illustre le test. En employant la terminologie Quad-Edge, ces trois vertex sont nommés Edge.V (P0), Edge.Sym.V (P2) et Edge.Sym.R.V (P1). On teste si le vertex opposé à P1, soit P3 (Edge.R.V), se situe à l'intérieur du cercle. Si c'est le cas, alors le segment Delaunay  $P0 \Rightarrow P2$  ne fait pas partie de la croûte, mais le segment Voronoï  $P1 \Rightarrow P3$ , lui, fait partie de l'axe médian (le test de la croûte renvoie FAUX). À l'inverse, si P3 est situé hors du cercle, le segment Delaunay  $P0 \Rightarrow P2$  appartient à la croûte (le test de la croûte renvoie VRAI) alors que le segment Voronoï  $P1 \Rightarrow P3$  n'appartient pas à l'axe médian.

Pour déterminer si P3 est à l'intérieur du cercle ou non, on utilise une fonction appelée *Ccircle* qui calcule la coordonnée du centre et le rayon du cercle formé par les vertex P0, P1, P2. Si la distance euclidienne entre le centre du cercle et P3 est inférieure au rayon du cercle, cela indique que P3 est situé à l'intérieur du cercle. Dans le cas contraire, il est à l'extérieur. Ceci se traduit mathématiquement par les formules suivantes:

Le formule pour obtenir la coordonnée du centre du cercle ( $C_x$ ,  $C_y$ ) est la suivante (Gold, 1998b) :

$$C_x = (-1/2) * (P/R)$$

$$C_y = (-1/2) * (Q/R)$$

où P, Q et R sont les déterminants des matrices suivantes, produites à partir des vertex P0, P1 et P2 situés sur l'arc de cercle.

$$P = \begin{vmatrix} P_0y, (P_0x^2+P_0y^2), 1 \\ P_1y, (P_1x^2+P_1y^2), 1 \\ P_2y, (P_2x^2+P_2y^2), 1 \end{vmatrix} \quad Q = - \begin{vmatrix} P_0x, (P_0x^2+P_0y^2), 1 \\ P_1x, (P_1x^2+P_1y^2), 1 \\ P_2x, (P_2x^2+P_2y^2), 1 \end{vmatrix} \quad R = \begin{vmatrix} P_0x, P_0y, 1 \\ P_1x, P_1y, 1 \\ P_2x, P_2y, 1 \end{vmatrix}$$

Le rayon du cercle s'obtient en calculant la distance euclidienne entre le centre du cercle (Cx,Cy) et un des vertex situé sur l'arc de cercle, tel P1, par exemple.

$$\text{Rayon du cercle: } \sqrt{((C_x - P_{1x})^2 + (C_y - P_{1y})^2)}$$

Cependant, il existe un cas d'exception, soit celui où le cercle défini n'est pas valide. Ceci survient lorsque les vertex P0, P1 et P2 sont positionnés de façon linéaire. Dans ce cas, on retient le segment P0⇒P2 comme faisant partie de la croûte.

Le *One-Step Crust and Skeleton Extraction Algorithm* a donné lieu au développement de deux applications dérivées. Débutons tout d'abord par la première application, c'est-à-dire l'algorithme de généralisation par rétraction de l'axe médian.

### 3. GÉNÉRALISATION PAR RÉTRACTION DE L'AXE MÉDIAN

#### 3.1 Revue des travaux sur la généralisation

Les algorithmes de généralisation cartographique sont d'un très grand intérêt pour ceux et celles qui oeuvrent dans le domaine des SIG et de la cartographie, en raison des besoins fréquents de réduire la complexité graphique des entités cartographiques dans le but de les représenter à plus petite échelle. Weibel (1995) ajoute: *"Another reason to reduce the accuracy and resolution of data set is the homogenization of different sets in the process of data integration"*.

Les diverses techniques de généralisation cartographique ont été sujet à discussions depuis nombre d'années déjà, souvent sous les appellations de simplification, élimination et déplacement. Les efforts investis dans ce domaine ont été mis au développement d'algorithmes pour la généralisation d'objets isolés comme, par exemple, des maisons, des routes, etc. Les travaux de Douglas et Peucker (1973) sur la généralisation des lignes ont été couronnés de succès et l'algorithme qu'ils ont développé fait aujourd'hui figure de référence dans le domaine. Malheureusement, ces algorithmes ne prennent souvent pas en compte la topologie des objets cartographiques. Ceci peut causer de sérieux problèmes lorsque l'on tente de généraliser des courbes de niveau ou tout autre objet adjacent car il y aura risque de chevauchements de lignes. La préservation de la topologie doit donc être considérée comme une caractéristique essentielle d'un bon algorithme de généralisation.

*"The problem with many existing methods for the generalization of spatial data is that they have been developed with no sufficiently focused objective in mind. Although they achieve generalization-like behaviour, it is not always clear whether they can be used for model generalization because they may not fulfil the above requirements. Likewise, many of these algorithms are also not suited for cartographic generalization because they do not pay attention to cartographic principles"*(Weibel, 1995).

Weibel (1995) considère que l'algorithme de Douglas et Peucker (1973) ne peut être considéré comme un vrai modèle de généralisation. Selon lui, une méthode de généralisation, pour être valable, devrait produire des résultats prévisibles et reproductibles; devrait minimiser les dif-

férences entre le modèle résultant et l'original; devrait réduire au maximum le volume de données; devrait préserver la topologie; devrait utiliser le moins de paramètres de contrôle possible et devrait surtout être efficace. D'autres travaux pertinents sur ce domaine incluent ceux de Weibel (1992), McMaster (1989), Brassel et Weibel (1988) ainsi que Bundy et al. (1995). Les travaux de Bundy et al. (1995) sont d'un intérêt particulier pour cette recherche car il utilise une structure topologique pour gérer les divers aspects de la généralisation.

Les travaux précédents de Gold (1999) et Gold et al. (1999a) sur l'extraction de la croûte et de l'axe médian ont permis de fournir les outils (*One-Step Crust and Skeleton Extraction Algorithm*) requis à l'élaboration d'une nouvelle méthode de généralisation cartographique qui utilise le concept de rétraction de l'axe médian. L'idée maîtresse derrière la conception de cette technique est simple: les objets simples ont des axes médians simples (Blum, 1967). Cela signifie qu'en modifiant la forme de l'axe médian, en le simplifiant, on réduit du même coup la complexité graphique de l'objet.

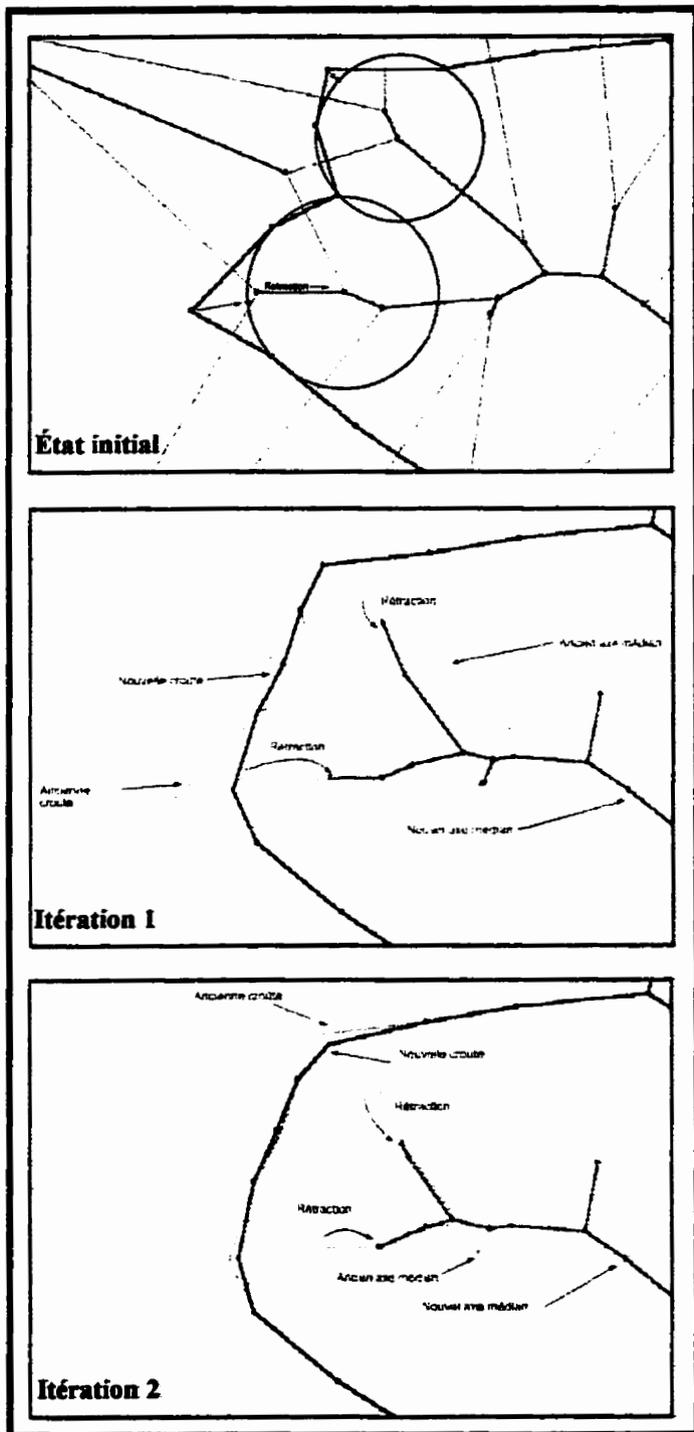
L'objectif est de développer un algorithme de généralisation cartographique rendant impossible les chevauchements d'objets. Ceci présuppose donc l'emploi d'une structure topologique, dans notre cas nous avons utilisé la structure Voronoï/Delaunay couplée avec la structure de donnée Quad-Edge. Comme nous l'avons mentionné précédemment, la structure de donnée Quad-Edge permet les mises à jour locales de la topologie, ce qui est une caractéristique essentielle au développement d'un tel algorithme.

## **3.2 Algorithme de généralisation par rétraction de l'axe médian**

### **3.2.1 Présentation**

Alt et al. (1995) ainsi que Blum (1967) ont montré que les feuilles (vertex terminaux de l'axe médian) correspondent aux endroits de courbure maximale de la croûte. Pour une courbe suffisamment échantillonnée, ceci signifie que trois (3) vertex adjacents sont cocirculaires, avec leur centre localisé aux feuilles de l'axe médian. Si nous souhaitons simplifier l'axe médian et par le fait même généraliser la croûte, nous devons rétracter les vertex terminaux vers leurs vertex parents respectifs. Ceci entraîne l'ajout d'un quatrième vertex sur le cercle. La rétraction

Figure 3.1



Ensemble du processus de généralisation par rétraction de l'axe médian.

en cours.

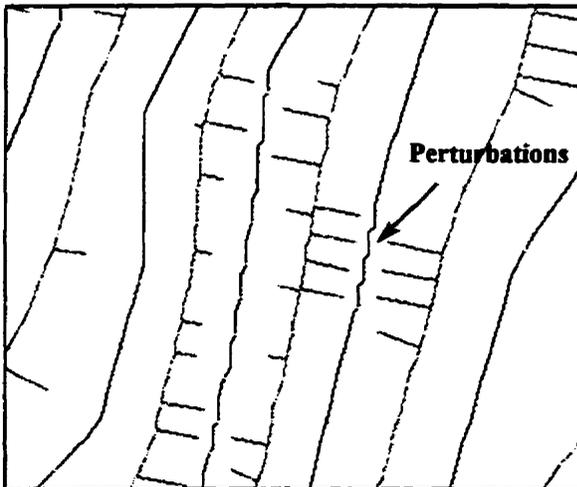
est appliquée en prenant le vertex central parmi les trois définissant la feuille de l'axe médian et en le déplaçant vers le vertex parent jusqu'à ce qu'il touche l'arc du cercle circonscrit, centré au vertex parent. En fait, la rétraction est un processus basé sur la cocircularité. Elle est appliquée itérativement de chaque côté de la croûte jusqu'à que tous les vertex soient situés sur les arcs des cercles. Cet état d'équilibre n'est atteint qu'après un nombre indéterminé d'itérations qui varient en fonction du nombre de vertex composant la croûte, mais aussi en fonction du degré de perturbations observables sur celle-ci. La figure 3.1 illustre l'ensemble du processus de généralisation. L'état initial montre la frontière d'un polygone (trait noir), les cercles circonscrits (en bleu) pour deux vertex parents de l'axe médian (en rouge) et la destination finale, après rétraction, de deux vertex de la croûte (en violet). Après chaque itération, on observe que la frontière du polygone a été adoucie de chaque côté. À l'itération 1 et 2, le trait en grisé et le trait en rouge pâle indique respectivement la frontière du polygone et la forme de l'axe médian avant l'itération

Le produit final est un axe médian simplifié qui se conforme bien au descripteur de forme décrit par Blum (1967). Quant à la croûte, elle est maintenant généralisée, les angles aigus ont été adoucis et les perturbations ont disparu. Toutefois, il peut arriver que certains vertex situés à des endroits stratégiques (coin d'un immeuble, par exemple, ou intersection de deux lignes) ne doivent pas être déplacés. Dans ce cas, on applique une étiquette à ces points avec la mention "ne pas bouger".

### 3.2.2 Caractéristiques de l'algorithme

L'algorithme de généralisation par rétraction de l'axe médian élimine les perturbations mineures le long des courbes (figure 3.2) et adoucit les angles. Les perturbations peuvent avoir

**Figure 3.2**



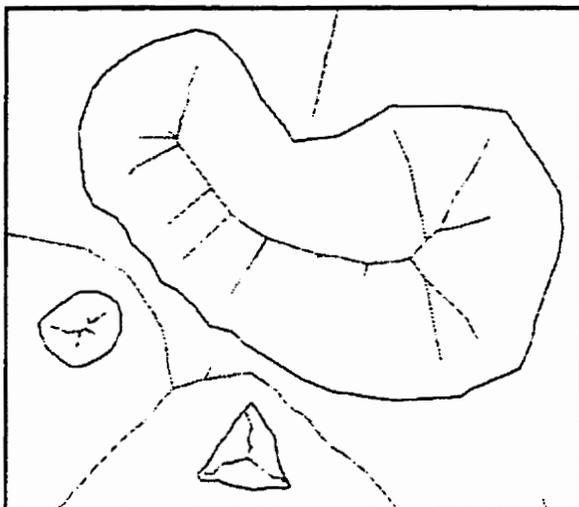
L'algorithme de généralisation par rétraction de l'axe médian élimine les perturbations mineures le long des courbes.

été provoquées par du bruit lors de la vectorisation d'une carte ou par la fatigue d'un opérateur qui numérise manuellement une carte. Pour que la généralisation puisse donner de bons résultats, la courbe ou croûte doit être suffisamment échantillonnée et donc ne présenter aucune rupture.

La superposition des objets traités par l'algorithme s'avère impossible du fait que chacun d'eux, qu'ils soient ligne, point ou polygone, est séparé par l'axe médian. L'axe médian fait office de "barrière" (figure 3.3). Au même titre,

les lignes qui sont adjacentes (routes, courbes de niveau, rivières, etc.) sont également séparées par l'axe médian (figure 3.2). Toutefois, la structure Voronoï/Delaunay ne garantit pas le maintien de la connexion (intersections) entre les lignes ni les polygones. Dans ce cas, il suffit de spécifier à l'algorithme que les vertex d'intersection (noeuds) ne doivent pas être déplacés.

L'algorithme procède par la mise à jour locale du diagramme de Voronoï/Delaunay et n'est

**Figure 3.3**

L'axe médian externe (exo-squelette) joue le rôle de frontière en isolant chacun des objets cartographiques.

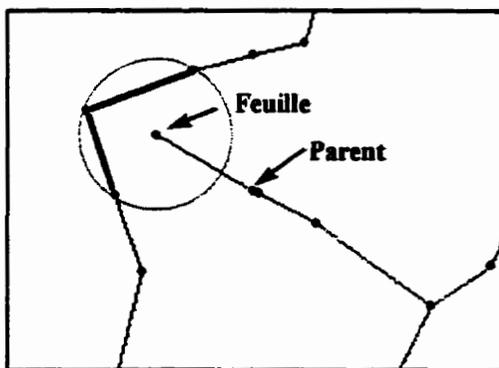
Leur méthode ne s'applique cependant qu'aux données matricielles et utilise une tolérance pour contrôler la réduction de l'axe médian. À l'inverse, la nôtre traite les données vectorielles et n'utilise pas de tolérance.

aucunement affecté par la densité et la distribution des objets sur le plan. De plus, il n'utilise aucune tolérance métrique pour le déplacement des points, ce qui signifie qu'il fonctionne indépendamment de l'échelle des données et qu'il ne requiert, à titre d'exemple, aucune modification pour migrer du traitement d'un jeu de données à l'échelle du 1 : 10 000 à un autre, à l'échelle du 1 : 1 000 000.

Ogniewicz (1994), de même que Ogniewicz et Ilg (1990), ont développé une méthode de simplification des objets utilisant l'axe médian.

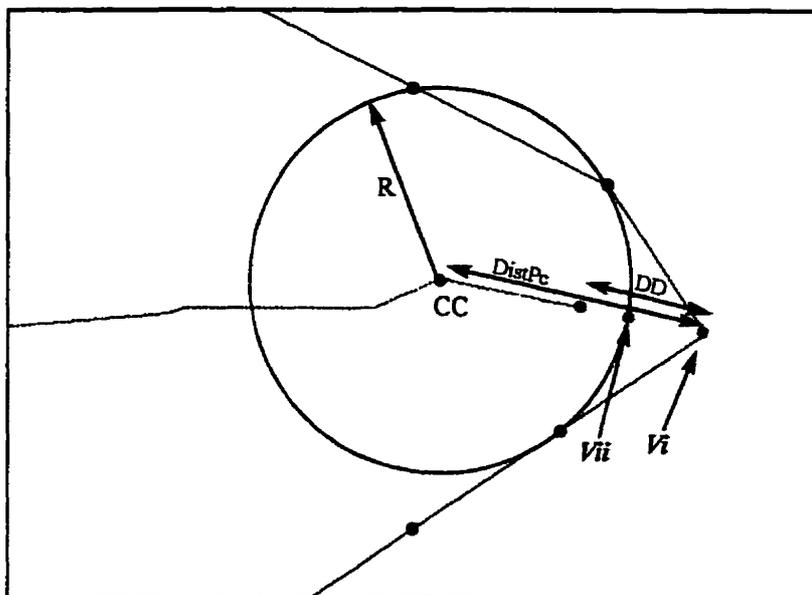
### 3.2.3 Description du code de programmation

La première opération réalisée par l'algorithme consiste à créer deux listes nommées *MovingPtList* et *NewPtList* qui serviront à stocker des vertex. Leur utilisation sera détaillée plus loin. On démarre la généralisation en appelant l'algorithme *Scan* qui est utilisé pour parcourir la structure Voronoï/Delaunay. Chaque segment traité par *Scan* est acheminé à une procédure nommée *Generalization*, qui va d'abord tenter de trouver un tronçon de croûte valide. Sur la figure 3.4, un tronçon de croûte (en bleu) est constitué de deux segments adjacents appartenant à la croûte (déterminé à l'aide de la fonction *Crust*) et dont leurs trois vertex sont

**Figure 3.4**

Un tronçon de croûte est constitué de deux segments adjacents appartenant à la croûte et dont leurs trois vertex sont cocirculaires indiquant ainsi l'emplacement d'une feuille de l'axe médian.

Figure 3.5



Variables utilisées pour le calcul de la nouvelle position d'un vertex sur l'arc de cercle.

Une fois qu'un tronçon valide de croûte a été repéré, la coordonnée du vertex central nommé  $V_i$  (parmi les 3 vertex cocirculaires) est acheminée à une autre fonction nommée *Circle*. Celle-ci définira la nouvelle position du point  $V_i$  sur l'arc du cercle centré au vertex parent de la feuille (figure 3.5). La nouvelle position de  $V_i$  soit  $V_{ii}$  est obtenue par les équations ci-dessous.

Soit les variables (figure 3.5):

$CC$  : Coordonnée du centre du cercle centré au vertex parent

$R$  : Rayon du cercle centré au vertex parent.

$DistPC$  : La distance entre le vertex  $V_i$  et  $CC$ .

$DD$  : La distance de déplacement, soit la différence entre  $DistPC$  et  $R$ .

$Lambda$  :  $DD / DistPC$

Les équations pour obtenir la coordonnée de  $V_{ii}$  sont respectivement:

Pour le X :  $((Lambda * CC.x) + (1-Lambda) * V_i.x)$

Pour le Y :  $((Lambda * CC.y) + (1-Lambda) * V_i.y)$

Si la coordonnée de  $V_i$  est différente de celle de  $V_{ii}$  et que  $V_i$  n'a pas été étiqueté précédemment comme un vertex à ne pas bouger, alors la fonction *Circle* placera la coordonnée de  $V_i$  dans la

cocirculaires, indiquant ainsi l'emplacement d'une feuille de l'axe médian.

Au cours de ce même passage, *Scan* apposera une étiquette aux vertex de jonction (noeuds) afin qu'ils ne soient pas déplacés ultérieurement, au risque d'altérer la topologie.

Une fois qu'un tronçon valide de croûte a été repéré,

liste nommée *MovingPtList* et celle de *Vii* dans *NewPtList*. Une fois le parcours de *Scan* complété, les deux listes sont envoyées à une dernière fonction nommée *ChangePt*. Celle-ci remplacera les vertex de *MovingPtList* avec ceux contenus dans *NewPtList* par une succession d'opérations d'effacement et d'insertion dans le maillage. La version actuelle de l'algorithme ne permet pas le déplacement des vertex d'une position A à une position B, mais plutôt l'effacement du vertex à la position A et l'insertion d'un nouveau vertex à la position B. Bien que l'algorithme pour le déplacement de vertex dans un maillage Voronoï/Delaunay existe à ce jour (Gold et al., 1998), aucun essai n'a cependant été tenté dans le but de comparer l'efficacité de cette procédure par rapport à celle actuellement employée.

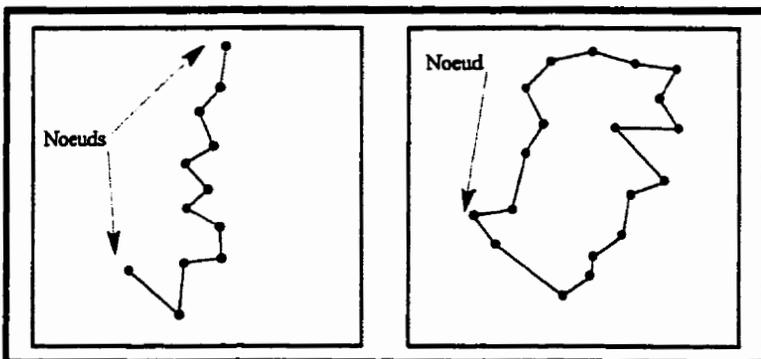
L'algorithme est ensuite relancé au besoin par l'utilisateur, s'il souhaite généraliser davantage la croûte. À chaque itération, de nouveaux vertex sont déplacés sur les arcs des cercles et un nombre croissant de vertex deviennent ainsi cocirculaires. Une fois que tous les vertex de la croûte ont été déplacés sur les cercles, à l'exception des nœuds, la généralisation est complétée car les deux listes de vertex demeurent alors vides.

## 4. APPLICATIONS

### 4.1 Applications utilisant les courbes

À l'intérieur d'un SIG, les courbes sont utilisées pour représenter de l'information sous forme

Figure 4.1



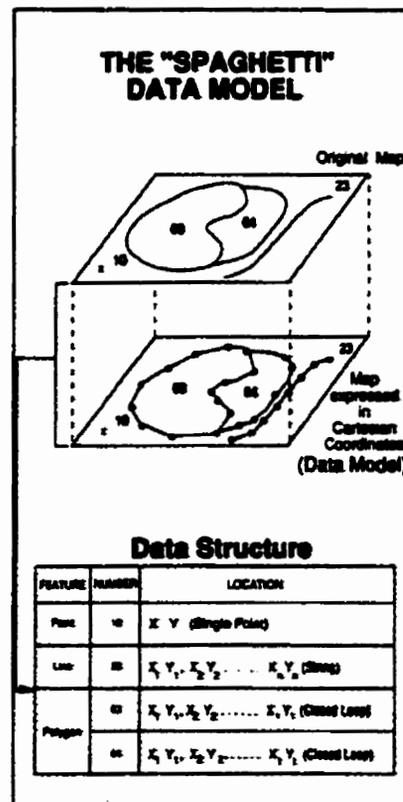
Exemples de courbes fermées et ouvertes.

d'objets dans l'espace selon la résolution des coordonnées utilisées.

Ainsi, une courbe est formée d'un nœud de départ et d'un nœud de fin définissant ses extrémités, mais également d'une suite de coordonnées nommées vertex, qui constitue "l'ossature" de la courbe. Les courbes permettent la représentation de structures linéaires, tels les réseaux hydrographiques et routiers, mais également les structures fermées que sont les polygones. La différence entre ces deux structures tient au fait que, dans la seconde, la courbe est refermée sur elle-même, c'est-à-dire que le nœud de départ est ancré sur le nœud de fin. La figure 4.1 présente un exemple de courbe fermée et ouverte. Dans le modèle vectoriel, on peut distinguer deux sous-modèles, le modèle spaghetti et le modèle topologique. Le premier est essentiellement une collection de chaînes de coordonnées sans structure inhérente (figure 4.2). Par contre, le second permet d'encoder les relations spatiales entre les

vectorielle. Brièvement, rappelons que l'information vectorielle est constituée par des séries de coordonnées géographiques ou autres, liées les unes aux autres selon un ordre défini dans le fichier de données. Par conséquent, le modèle vectoriel de données pourra permettre le positionnement très précis

Figure 4.2



La modèle spaghetti ou le vectoriel non structuré topologiquement.  
Source: Aronoff, 1989

entités graphiques. C'est ce dernier modèle qui est utilisé à l'intérieur des SIG.

Au cours des quelques pages qui suivent, nous présenterons des exemples d'applications utilisant les courbes et nous démontrerons comment l'emploi d'une structure topologique s'avère crucial afin de connecter les courbes les unes aux autres pour ainsi former un ensemble cohérent d'objets cartographiques.

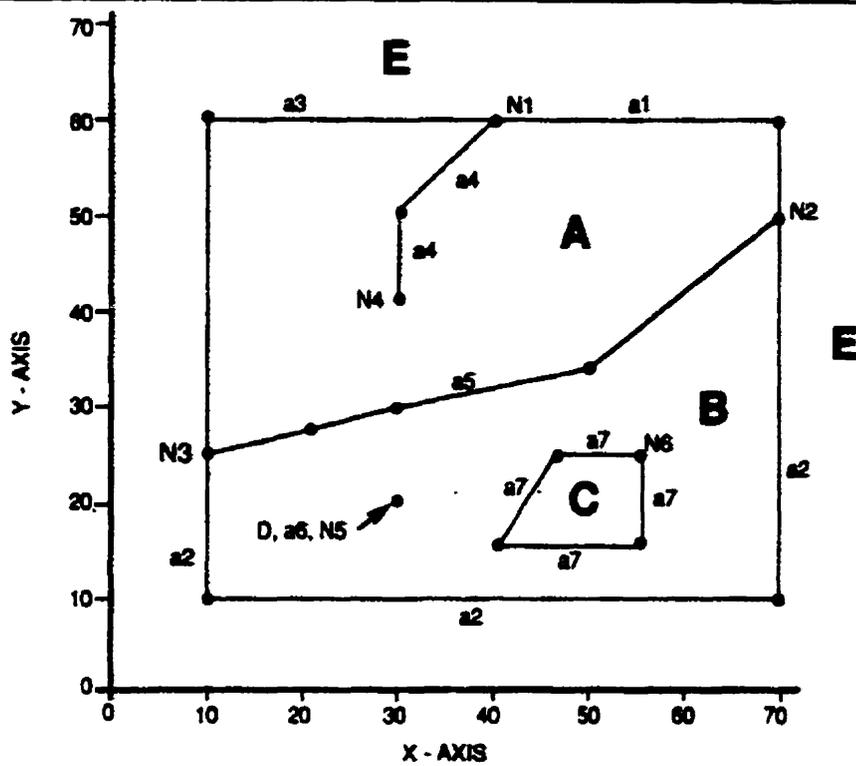
#### 4.1.1 Construction de cartes polygonales

Les polygones constituant les cartes polygonales sont utilisés pour représenter toute entité de nature surfacique, qu'il s'agisse d'un lac, d'un peuplement forestier, d'un lot cadastral, etc. Les polygones sont la représentation vectorielle de la deuxième dimension à l'intérieur des SIG.

L'élaboration d'une carte polygonale se fait essentiellement à partir d'un ensemble de courbes communément appelées "Arc" (terme provenant de la terminologie Arc/Info). La connexion et la fermeture des courbes, en vue de constituer les polygones, présupposent l'emploi de pointeurs topologiques. En fait, seul les pointeurs peuvent assurer la fermeture d'une courbe sur elle-même car en raison de la précision arithmétique limitée des ordinateurs, rien ne garantit qu'une fois fermée, le nœud de départ de la courbe sera exactement superposé sur son nœud de fin.

Diverses structures topologiques peuvent être employées afin de connecter les courbes et ainsi former les polygones. Parmi celles-ci, mentionnons la structure *Arc-Node*, utilisée, entre autres, par le SIG Arc/Info et présentée au tableau 4.1. Cette structure utilise quatre tables pour établir les relations topologiques entre les points, les lignes et les polygones. Une première table contenant la topologie des polygones où sont énumérés les arcs composant chacun d'eux. Une seconde table pour la topologie des nœuds où sont énumérés tous les arcs partageant un nœud donné. La troisième table est celle de la topologie des arcs. Pour chacun des arcs, on énumère leurs nœuds de départ et de fin, ainsi que les polygones situés à leur gauche et à leur droite, en regard au sens de numérisation de l'arc. Enfin, la quatrième et dernière table est celle des nœuds et vertex intermédiaires composant chacun des arcs.

Tableau 4.1



**SPATIAL DATA ENCODING**

POLYGON TOPOLOGY		NODE TOPOLOGY		ARC TOPOLOGY				
POLYGON	ARCS	NODE	ARCS	ARC	START NODE	END NODE	LEFT POLYGON	RIGHT POLYGON
A	a1, a5, a3	N1	a1, a3, a4	a1	N1	N2	E	A
B	a2, a5, 0, a6, 0, a7	N2	a1, a2, a5	a2	N2	N3	E	B
C	a7	N3	a2, a3, a5	a3	N3	N1	E	A
D	a6	N4	a4	a4	N4	N1	A	A
E	area outside map coverage	N5	a6	a5	N3	N2	A	B
		N6	a7	a6	N5	N5	B	B
				a7	N6	N6	B	C

ARC COORDINATE DATA			
ARC	START X,Y	INTERMEDIATE X,Y	END X,Y
a1	40, 60	70, 60	70, 50
a2	70, 50	70,10; 10,10	10, 25
a3	10, 25	10,60	40, 60
a4	40, 60	30,50	30, 40
a5	10, 25	20,27; 30,30; 50,32	70, 50
a6	30, 20		30, 20
a7	55, 27	55,15; 40,15; 45,27	55, 27

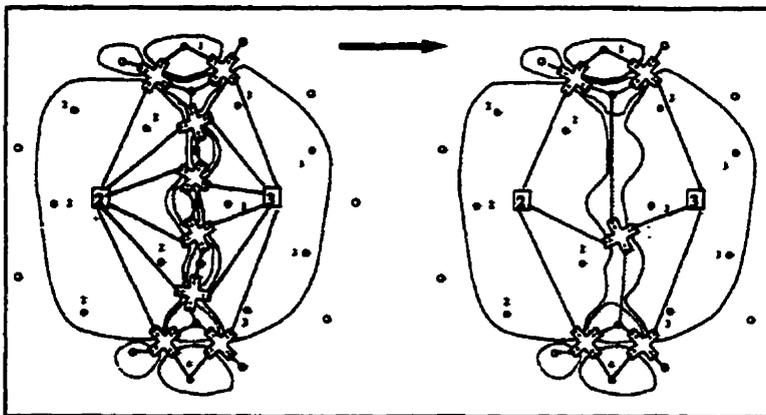
La structure topologique Arc/Node.  
Source: Aronoff, 1989

Du fait qu'elle a été implantée dans le SIG le plus utilisé sur le marché, la structure *Arc-Node* fait aujourd'hui figure de référence pour la construction de la topologie. Gold (1998) a développé une structure topologique nommée Quad-Arc, issue d'une version modifiée de la structure de données Quad-Edge. Elle permet la connexion des courbes les unes aux autres, mais aussi la gestion des relations de voisinage entre les objets. En fait, les edges qui pointaient vers les segments Voronoï pointent dorénavant vers les polygones.

*"The Quad-Arc approach may be significant in Geomatics as it provides a simple data structure that suits the needs of the data input process, preserves the map topology, and may be implemented within small-scale PC-based software for topological map querying."* (Gold, 1999).

La structure Quad-Arc permet de séparer complètement la topologie des attributs, en ce sens que les pointeurs topologiques sont entièrement dans la structure de données. Nul besoin donc d'utiliser des tables de données externes, comme c'est le cas pour la structure Arc/Node. La

**Figure 4.3**



Transformation Quad-Edge à Quad-Arc.  
Source: Gold, 1998a

figure 4.3 montre le résultat d'une transformation Quad-Edge à Quad-Arc pour deux polygones portant les numéros 2 et 3. On remarque que les nombreux Quad-Edges de l'image de gauche ont été remplacés par un seul Quad-Arc sur l'image de droite.

Nous n'entrerons pas davantage dans les détails des structures topologiques car ceci dépasse le cadre de cette recherche. Le lecteur intéressé à en savoir plus pourra consulter l'ouvrage de Laurini et Tompson, 1992.

### 4.1.2 Construction de réseaux

Les courbes, de par leur géométrie, sont conçues *a priori* pour représenter des entités linéaires tels les réseaux routiers, les réseaux hydrographiques et les courbes d'élévation, pour ne citer que ceux-là. À l'intérieur d'un réseau linéaire connecté, que l'on peut aussi appelé graphe, ces courbes sont rattachées les unes aux autres par des nœuds. Comme nous l'avons mentionné précédemment, les nœuds sont des éléments d'une importance critique lorsqu'il s'agit d'assurer l'intégrité topologique d'un réseau.

Un réseau voué à des applications nécessitant des opérations de déplacements, comme un réseau routier, par exemple, doit impérativement posséder une structure topologique. Ceci est primordial afin de pouvoir interroger le réseau et obtenir des réponses aux questions suivantes. Quel(s) tronçon(s) du réseau est connecté(s) à tel autre ? Quelles courbes sont rattachées à un nœud donné ? Quel est le chemin le plus court ou celui qui minimise le temps de parcours entre le point A et le point B? À noter que les courbes portent le nom de tronçon lorsque qu'utilisées dans un contexte de réseau routier. Pour pouvoir répondre à ces requêtes, l'intégrité topologique du réseau doit être parfaite, c'est-à-dire qu'il ne doit pas y avoir de redondance au niveau des nœuds et des courbes. En plus de la topologie, on doit également posséder pour chacun des tronçons l'information relative au sens de navigation autorisé sur celui-ci. On distingue ainsi les tronçons du réseau à sens unique de ceux qui permettent les déplacements dans les deux directions.

Mais il n'y a pas que les réseaux de transport qui utilisent ce type d'information, les réseaux hydrographiques le font également. En effet, pour être capable de spécifier quelle section d'un cours d'eau se situe en amont d'une autre ou quels sont les cours d'eau qui se déversent dans une rivière donnée, la présence d'une structure topologique est requise ici aussi. De plus, le sens d'écoulement des eaux pour chacune des sections du réseau hydrographique doit être précisé.

### 4.1.3 Construction de courbes de niveau

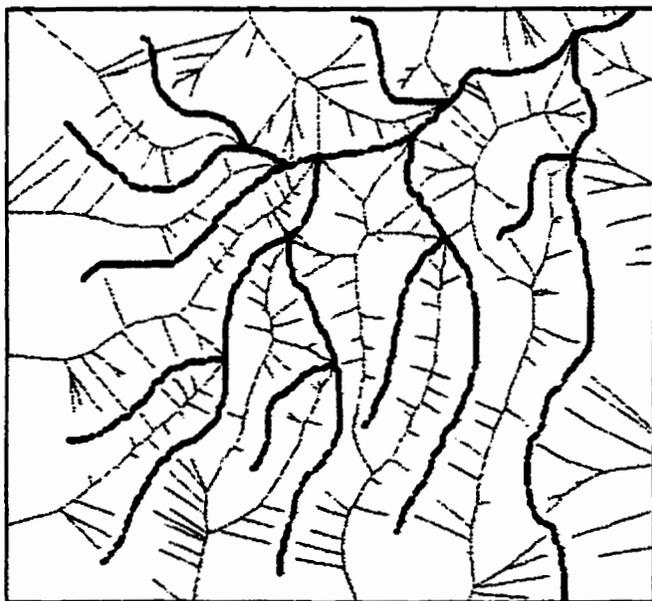
Lorsque les courbes sont utilisées pour la confection de carte d'élévation (isohypses), aucune structure topologique n'est requise. En revanche, les courbes doivent adéquatement être fermées lors de la numérisation en utilisant une fonction de type "snap". Également, on s'assurera de rattacher une étiquette d'élévation à chacune des courbes durant le processus de numérisation

## 4.2 Applications utilisant l'axe médian

À la section 2.3.1, nous avons présenté l'axe médian et ses propriétés, voyons maintenant quels types d'applications peuvent l'utiliser.

En regard des raisons qui ont poussé Blum (1967) à définir l'axe médian, nous pouvons affirmer que l'application première de l'axe médian est celui de descripteur de forme. En effet, il est utilisé dans de nombreuses applications de reconnaissance artificielle de forme (*OCR - Optical Characters Recognition*) de même qu'en robotique car il fournit, de part ses propriétés, une description simplifiée de l'objet, donc plus facile à interpréter, mais toute aussi complète.

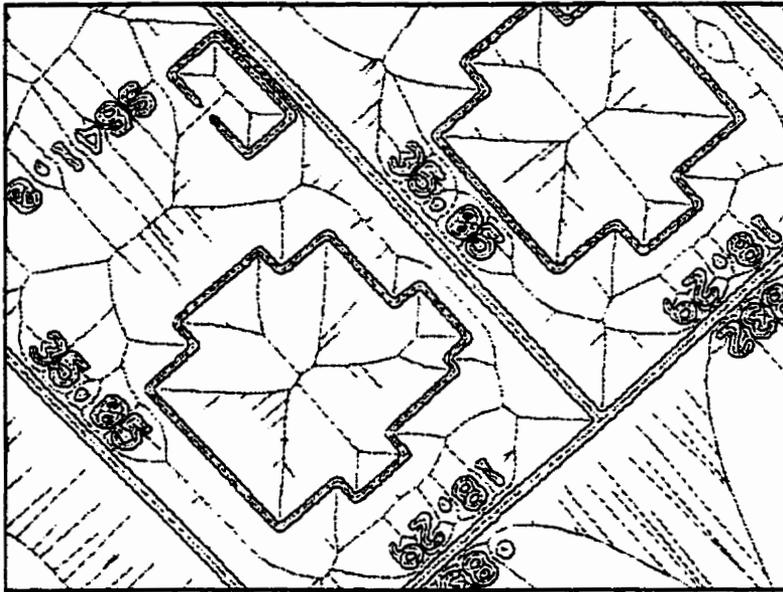
**Figure 4.4**



Axe médian d'un réseau hydrographique.

L'axe médian pourrait également être utilisé pour le stockage de données spatiales, telles que les cartes polygonales. En effet, nous avons montré à la section 2.3.1 qu'en ne possédant que l'axe médian comme source d'information, il est possible de reconstruire l'objet (la croûte) dans son intégralité (figure 2.14).

À partir d'un réseau hydrographique dont les vertex ont été triangulés, l'axe médian, une fois extrait, fournit une bonne appro-

**Figure 4.5**

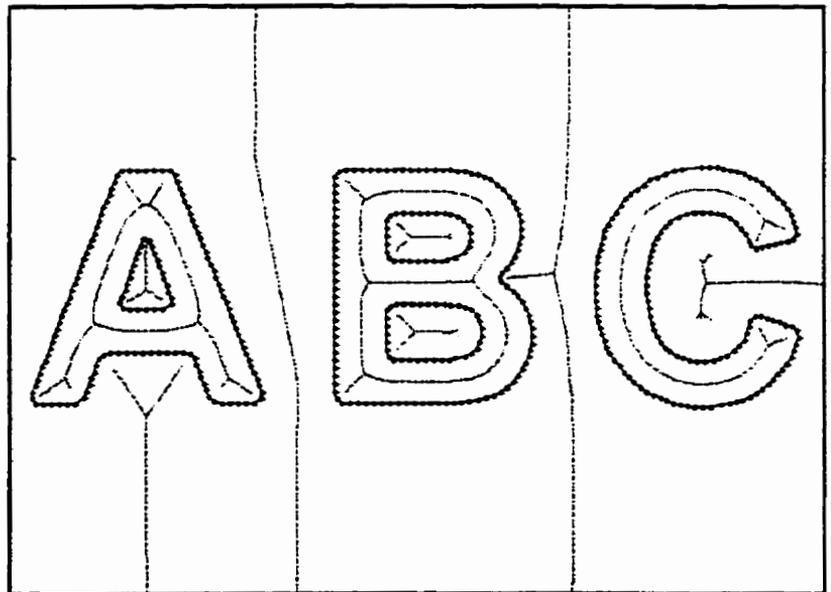
Axe médian pour une carte urbaine.

tionnons aussi que les branches de l'axe médian donnent une bonne indication sur l'orientation de la pente du terrain. Nous reviendrons sur ce point un peu plus loin.

L'axe médian peut également être utilisé pour la manipulation d'éléments textuels en cartographie, plus précisément pour la détection et la mise en place de toponymes ou tout autre symbole textuel (Gold, 1999). La figure 4.5 montre une portion d'une carte urbaine. On remarque que chacune des lettres est séparée par l'axe médian, qui définit ses relations spatiales avec les autres objets cartographiques de son voisinage.

L'axe médian interne des caractères (endo-squelette) permet de définir leur forme,

ximation du bassin versant comme le montre la figure 4.4. Cependant, la forme du patron de drainage obtenu est celle se conformant à un terrain homogène, ce qui n'est pas toujours le cas dans la réalité. Cette approche a été utilisée avec succès en Colombie-Britannique pour l'estimation préliminaire des bassins versants. Toutefois, des corrections par observations directes sur le terrain s'avéraient nécessaires. Men-

**Figure 4.6**

Axe médian de trois caractères de l'alphabet.

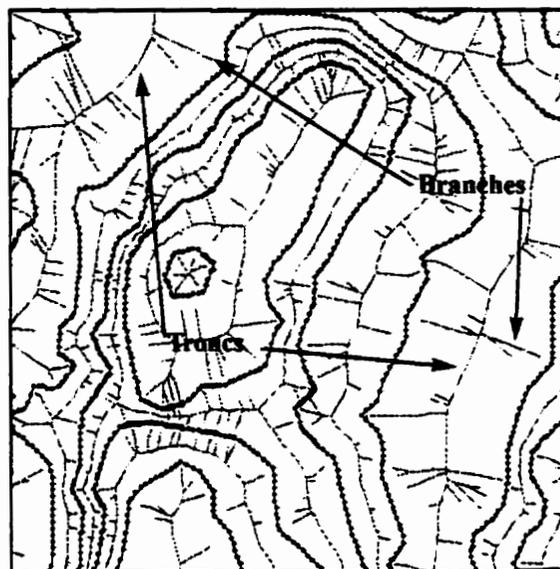
alors que l'axe médian externe (exo-squelette) permet de définir les relations spatiales entre eux (figure 4.6). Par conséquent, il s'avère donc possible de connaître la position d'une lettre par rapport à ses voisins et ce, à l'intérieur même d'un mot. D'autres recherches ont été menées sur l'extraction d'éléments textuels dans les cartes, notamment par les chercheurs Burge et Monagan (1995a et 1995b).

La modélisation du terrain est aussi un domaine où les propriétés de l'axe médian peuvent être appliquées. Après avoir triangulé les vertex des courbes de niveau, l'axe médian fournit l'information suivante. D'une part, les branches de l'axe indiquent l'orientation de la pente du terrain, car chacune d'elle est perpendiculaire au segment de courbe auquel elle est associée (figure 4.7).

Également, les branches indiquent les endroits de plus forte courbure sur les courbes. Lorsqu'un modèle numérique d'élévation est produit à partir de courbes de niveau, c'est dans les zones de forte courbure que le phénomène des triangles plats survient. Cette situation problématique, que nous aborderons en détail à la section 6, provoque un effet de terrassement à la surface du modèle, du fait que l'élévation est la même sur tous les sommets du ou des triangle(s).

Enfin, le tronc de l'axe médian indique l'emplacement des courbes intermédiaires. L'élévation le long du tronc est la moyenne des valeurs des deux courbes voisines (figure 4.7).

Figure 4.7



Axe médian d'un ensemble de courbes de niveau.

## 5. MODÉLISATION DE TERRAIN

### 5.1 Revue des travaux sur la modélisation du terrain

L'intégration de données topographiques à l'intérieur d'un système d'information à référence spatiale requiert généralement la conception d'un modèle numérique de terrain (MNT). *“Digital terrain models (DTMs) have been in use for more than three decades and have become a major constituent of geographic information processing”* (Weibel et al, 1990). Ce type de modèle fournit une représentation de la réalité terrain “utilisable” à l'intérieur d'un tel système, contrairement à celle offerte par les courbes de niveau. *“In GIS, DTMs provide an opportunity to model, analyse and display phenomena related to topography or other surfaces”* (Weibel et al. 1990). En réalité, l'avantage du MNT est qu'il fournit une représentation numérique et continue de l'élévation.

De nombreux projets de recherche ont porté sur les MNT au cours des dernières années. Citons à ce titre, les travaux de Carrara et al. (1997), qui ont porté sur l'évaluation de quatre logiciels commerciaux spécialisés dans la production de MNT à partir de courbes de niveau, en regard de cinq critères de qualité préalablement définis. Également, Eklundh et al. (1995) ont démontré qu'un gain appréciable dans le temps de traitement nécessaire à la production d'un MNT pouvait être obtenu lorsque l'on utilisait un échantillonnage régulier de points plutôt que des courbes de niveau. Leurs conclusions ont montré que l'usage de point permet d'obtenir une meilleure distribution statistique des valeurs, que la précision de ces valeurs est comparable à celle obtenue par les courbes de niveau, malgré le fait que la taille des données soit plus réduite. Enfin, Li (1994) a produit une étude comparative afin d'évaluer la précision des MNT par rapport à quatre modèles de données. Il en est venu à la conclusion que l'emploi de courbes de niveau donne de meilleurs résultats en terrain à pente forte et en terrain plat, contrairement à une grille régulière de points.

### 5.2 Le modèle grille versus le modèle à base de triangles (TIN)

La représentation du terrain sous forme de grille découpe l'espace en cellules d'égale dimension. À l'intérieur de l'ordinateur, cette structure est stockée sous forme d'une matrice à deux

**Figure 5.1**

12	12	13	13	14	15	14	14	14
12	12	12	13	14	15	14	14	13
12	12	12	13	13	14	14	13	12
12	12	12	13	13	13	13	12	12
12	12	12	12	13	13	13	12	12
12	12	12	12	12	13	12	12	12
12	12	11	12	12	13	12	12	11
11	11	12	12	12	12	12	11	10
12	12	12	12	12	12	12	10	9

Modélisation du terrain à l'aide d'une grille.

Source: Weibel et al., 1990

(2) dimensions, au même titre qu'un fichier image est constitué de lignes et de colonnes de pixels. Chaque cellule de la grille possède une valeur d'élévation, ce qui signifie que l'élévation est la même sur toute la surface d'une cellule (figure 5.1). La résolution de la grille est également fixe, sauf pour certaines applications d'infographie où l'on utilise des techniques permettant de faire varier la résolution en fonction de la position de l'observateur par rapport au modèle. Dans ce cas, certaines cellules seront divisées en 4, 8, 16 ou plus pour accroître la résolution à un endroit bien précis.

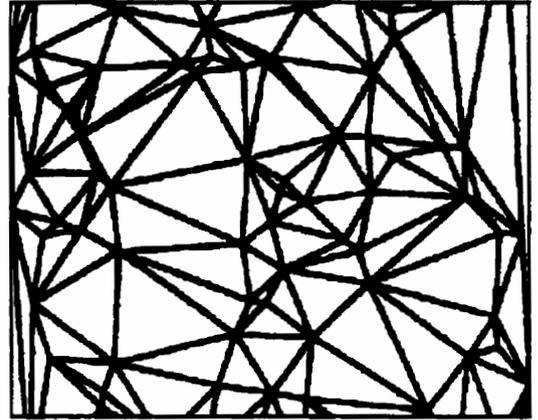
La résolution de la grille est normalement définie en fonction des besoins de l'utilisateur, mais parfois aussi en fonction de la superficie à représenter et des caractéristiques du terrain. Si la zone couverte est de faible dimension, on pourra s'accommoder d'une résolution plus élevée. Également, si la topographie du terrain est très accidentée, on préconisera une résolution élevée afin d'obtenir un niveau de détail satisfaisant. Mais la taille de la grille relève avant tout des besoins de l'utilisateur, car rien ne sert de produire un modèle très détaillé si on ne désire qu'une représentation grossière du terrain. On doit garder à l'esprit qu'un modèle de haute résolution demande plus de temps de calcul pour être produit, en raison d'un plus grand nombre de points à interpoler, et qu'il sera, par conséquent, plus lourd à manipuler lors de traitements ultérieurs.

L'avantage du modèle grille est qu'il est facile à implanter dans un système informatique en raison de sa structure simple. Malgré tout, ce n'est pas un modèle qui s'adapte à la complexité du relief. En effet, le nombre de cellules par unité de surface, qui servira à modéliser une surface plane, sera le même pour représenter une vallée aux parois escarpées. Une grille peut donc difficilement décrire les structures topographiques du terrain (Weibel et al., 1990). De plus, ce type de modèle ne conserve pas les points de données originaux car ceux-ci sont, en quelque sorte, "fondus" dans la grille. Par conséquent, la grille n'épouse pas exactement l'emplacement des points de données.

Le réseau de triangles irréguliers, communément appelé TIN, est une structure permettant de

représenter le relief terrestre au moyen de triangles de type Delaunay (figure 5.2). Contrairement à la grille, le TIN est plus difficile à implanter dans un système informatique car il requiert la présence d'une structure topologique pour la mise à jour de la triangulation lorsqu'un nouveau point y est inséré. Cependant, il a l'avantage de pouvoir adapter sa résolution de façon locale en fonction de la topographie du terrain. Peu de triangles seront ainsi nécessaires pour modéliser un terrain plat; par contre, un canyon sera modélisé par un nombre plus élevé de triangles. Par son caractère adaptatif, le TIN stocke ainsi moins d'informations superflues que le modèle grille. La résolution du modèle est accrue par subdivision des triangles en sous-triangles.

**Figure 5.2**



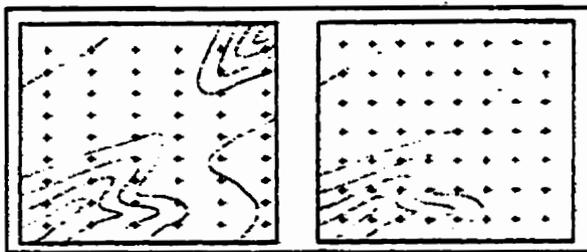
Modélisation du terrain par un réseau de triangles irréguliers.  
Source: Weibel et al., 1990

Un autre avantage du TIN est qu'il conserve les points de données originaux dans son maillage. Les sommets de certains triangles passeront donc exactement aux points de données.

### 5.3 Les diverses sources de données des MNT

Les sources de données les plus couramment utilisées pour la confection d'un MNT sont le semis de points d'altimétrie obtenu par photogrammétrie et le fichier d'isohypses (courbes de niveau).

**Figure 5.3**



Exemples de captage photogrammétrique par échantillonnage régulier pour deux types de densité.  
Source: Weibel et al., 1990

Le captage de point d'altimétrie par photogrammétrie se base sur l'interprétation par stéréoscopie de photos aériennes ou d'images satellitales. L'échantillonnage est en général régulier, c'est-à-dire que les points sont saisis à intervalles fixes (figure 5.3). Selon la densité de l'échantillonnage, le modèle numérique résul-

tant aura une précision plus ou moins grande. Mais ce type d'échantillonnage est avant tout réservé aux terrains homogènes, à topographie peu ou pas accidentée car sinon, trop de points

**Figure 5.4**



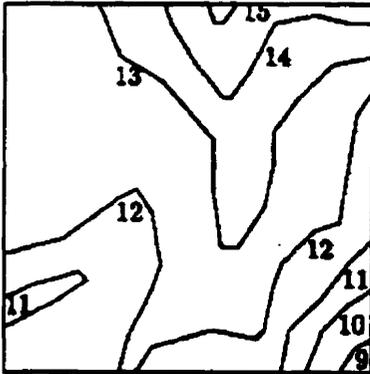
Exemples d'échantillonnage progressif permettant d'adapter la densité de l'échantillonnage en fonction de la complexité du terrain.

Source: Weibel et al., 1990

seront saisis en zone plane et un nombre insuffisant en zone accidentée. Dans certains cas particuliers, il est alors nécessaire de faire varier l'échantillonnage selon la complexité du terrain; plus de points seront ainsi saisis aux endroits de relief accidenté et moins en zone plane. Cette technique appelée échantillonnage progressif (figure 5.4) a été développée par Makarovic (1973).

En somme, l'avantage d'obtenir les points d'élévation par photogrammétrie est qu'on obtient une couverture complète et relativement homogène de la région. Ceci facilite beaucoup la tâche des algorithmes d'interpolation qui auront à estimer de nouvelles valeurs d'élévation entre les points de données. Malheureusement, cette technique de saisie est coûteuse et n'est réservée qu'aux fins de grands projets. (Weibel et al, 1990)

**Figure 5.5**



Les isohypses ou courbes de niveau sont les données les plus fréquemment utilisées pour la production de MNT.

Source: Van Kreveld, 1997

*"The direct derivation of the DTM from traditional photogrammetric analysis of stereo aerial photographs at the time of map production can certainly provide the best results, but the cost involved would be unaffordable when dealing with high-resolution terrain models over large regions"* (Carrara et al., 1997).

Les isohypses ou courbes de niveau sont les données les plus fréquemment utilisées pour la conception de MNT (figure 5.5). Les cartes d'isohypses sont largement disponibles (agences nationales ou militaires) et par le fait même peu coûteuses.

*"In many developed countries, detailed, recently produced, topographic maps are available. By exploiting automatic scanning and vectorization technology, contours can be cost-effective"*

*tively digitized and used as source data for DTMs*" (Carrara et al., 1997).

Malgré que ce type de donnée soit populaire, les isohypses sont avant tout une forme de visualisation du relief et ne sont pas particulièrement adaptées pour la conception de MNT (Weibel et al., 1990). En effet, trop de points sont échantillonnés sur les courbes et aucun entre celles-ci. De plus, lorsque qu'utilisées pour produire un TIN, les isohypses ont tendance à produire un effet de terrassement, communément appelé "triangles plats", aux endroits où les courbes sont incurvées. Nous abordons ce problème à la section 5.6. Conséquemment, les MNT produits avec ce type de données ont un degré d'exactitude plutôt limitée.

## **5.4 L'interpolation**

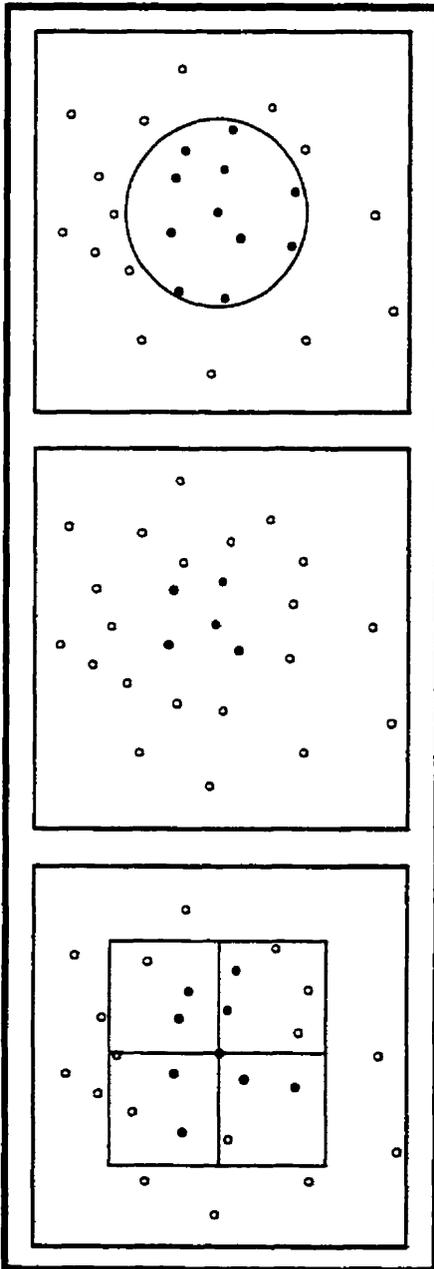
### **5.4.1 Introduction**

L'interpolation est une technique permettant de rendre continue une distribution discrète de valeur. Elle permet dès lors d'estimer la valeur d'une donnée à un endroit où aucune n'existe. De nombreuses méthodes d'interpolation existent et la technique traditionnellement employée consiste à superposer une grille sur le plan contenant les données. Ensuite, à l'aide de certains critères de recherche des points voisins, les valeurs sont interpolées entre les points de données. L'interpolation par krigeage et par plus proches voisins procèdent, entre autres, de cette façon. D'autres méthodes utilisent des fonctions polynomiales. Bien qu'elles soient différentes les unes des autres, les techniques d'interpolation s'intéressent tous au même problème, qui est ici central, soit la recherche des points candidats (points de données). Peu importe la méthode employée, la qualité de l'interpolation dépend largement du choix de ces points. Deux approches existent pour la recherche des points candidats. La première est de type métrique et les méthodes qui l'utilise (krigeage et plus proche voisin par exemple) interpolent à partir de points choisis arbitrairement dans l'espace. La seconde approche procède différemment. Les points candidats sont sélectionnés en fonction des relations de contiguïté qui existent entre les objets. La méthode d'interpolation "aire volée" appartient à cette seconde famille dite d'interpolation par voisin naturel. Pour une vue d'ensemble des méthodes d'interpolation existante, le lecteur pourra consulter les ouvrages de Burrough (1986), Lam (1983) et Watson (1994).

### 5.4.2 Interpolation à partir de points arbitraires

L'interpolation à partir de points arbitraires est basée sur le concept newtonien de l'espace (Gold, 1989). Les points candidats sont habituellement sélectionnés en utilisant les critères suivants ou leur combinaison (Chongjiang, 1996) : un critère de distance, un critère de nombre

**Figure 5.6**



Critères de recherche des points candidats. Au haut, le critère de distance. Au milieu, le critère de nombre. Au bas, le critère basé sur des secteurs.

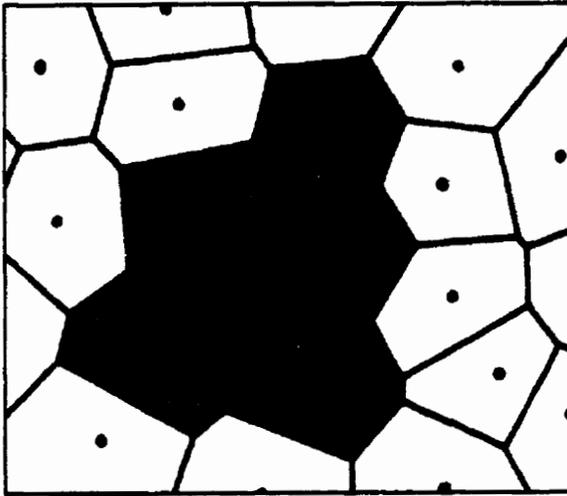
et un critère basé sur des secteurs (figure 5.6). Le critère de distance base la sélection des points candidats en fonction de leur distance au point d'estimation. Par exemple, on ne retient que les points situés à l'intérieur d'un cercle de diamètre  $X$  centré au point d'estimation. Le critère de nombre retient que les  $N$  points les plus près du point d'estimation. Enfin, le critère basé sur des secteurs divise la surface autour du point d'estimation en un nombre égal de secteurs (quadrants, octants) pour ne retenir ensuite que les deux à quatre points les plus près du point d'estimation pour chacun des secteurs.

Le problème avec ces méthodes de recherche, c'est qu'elles ne permettent pas d'obtenir les vrais voisins au sens spatial du terme. De plus, elle génère des discontinuités de surface aux endroits où le groupe de points candidats change. À ce sujet, Gold (1988) mentionne:

*"It is particularly difficult to ensure that a data point's influence (weighting) has decreased to zero before it is discarded from the neighbourhood set; if this condition is not met, discontinuities in the surface will occur."*

### 5.4.3 Interpolation par voisin naturel : la méthode "aire volée"

Figure 5.7



Le critère Voronoï pour la sélection des points candidats.

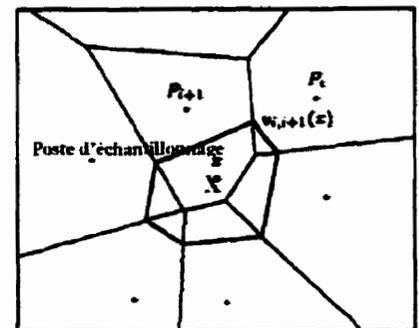
La méthode d'interpolation "aire volée" (Gold, 1989) exploite cette définition de la contiguïté spatiale offerte par le diagramme de Voronoï, d'une part pour la recherche des points candidats et, d'autre part, pour le calcul des poids (facteurs de pondération) qui seront attribués à chacun des points candidats. Ainsi, un point est le voisin d'un autre si et seulement si sa cellule Voronoï lui est adjacente (figure 5.7). Selon Chongjiang (1996), ceci constitue le critère Voronoï pour la sélection des points candidats.

Il mentionne à ce sujet: *"Since the nearest neighbours obtained by this criterion are directly nearest the grid node, we can say that they are the best in the sense that they make direct contributions to the grid node. Another advantage of the Voronoi criterion is that the obtained nearest neighbours are not collinear."*

Avec l'interpolation par "aire volée", le poids attribué à un point voisin décroît jusqu'à atteindre zéro lorsque celui-ci cesse d'être voisin. À l'inverse, plus le point voisin est proche du point à estimer, plus le poids qui lui est attribué est fort, en d'autres mots, plus son influence est grande.

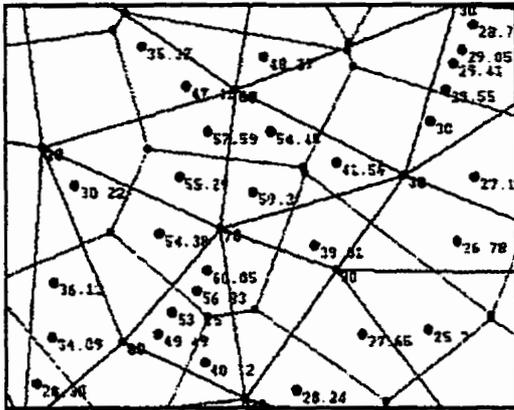
Voici comment fonctionne l'interpolation "aire volée". Lorsqu'un point est inséré dans la structure Voronoï, une nouvelle cellule est créée, "volant" ainsi une portion de l'espace aux cellules voisines (figure 5.8). Cela s'explique par le fait que la structure Voronoï couvre entièrement l'espace; il n'existe aucun vide entre les cellules. Par conséquent, la création d'une nouvelle cellule doit forcément se faire au dépend de ses voisins. C'est la proportion d'aire volée à chacun des voisins qui servira de facteur de

Figure 5.8



Interpolation "aire volée".  
Source: Gold, 1994

Figure 5.9



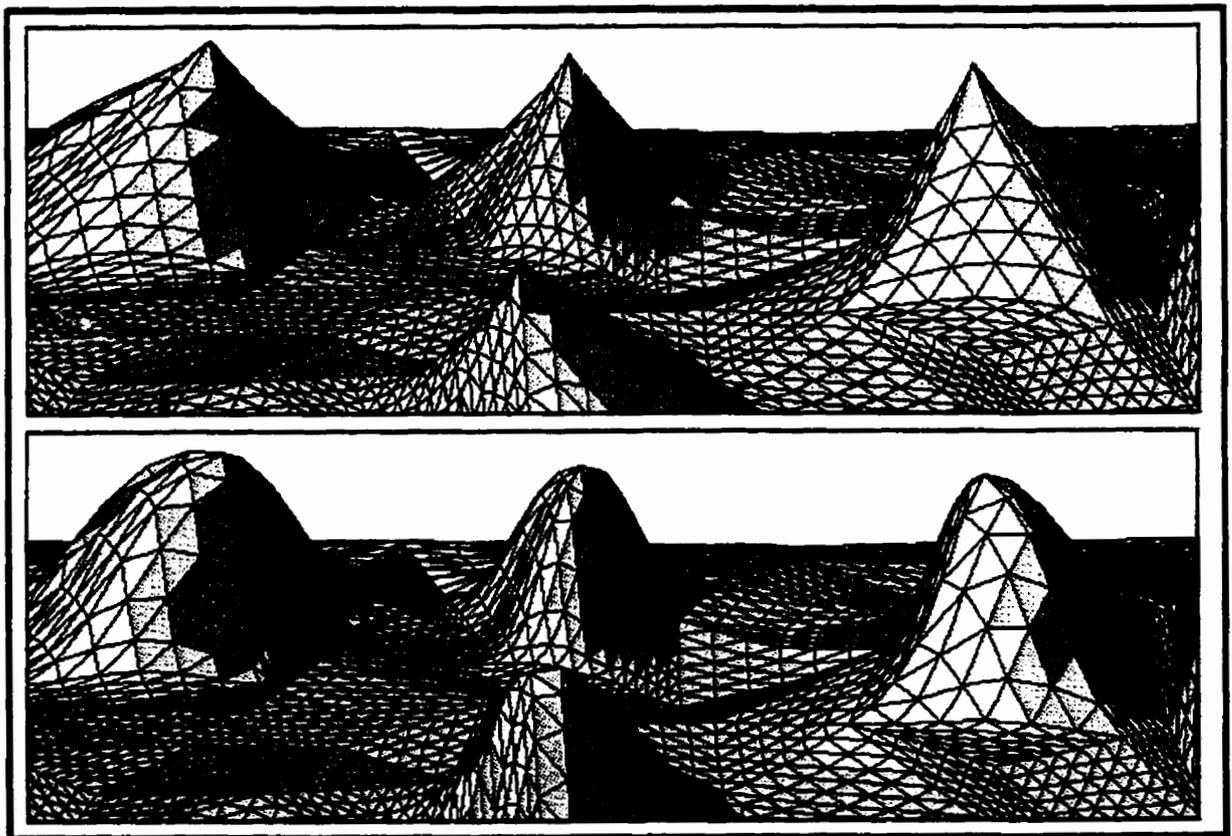
Valeurs obtenues par interpolation "aire volée".

pondération. Plus celle-ci est importante, plus le poids sera élevé et plus la valeur du point candidat aura de l'influence sur le calcul de la valeur au poste d'échantillonnage. La figure 5.9 illustre les points de données en noir et les valeurs obtenues par interpolation en vert.

La séquence d'exécution de l'algorithme d'interpolation est la suivante :

- 1) Insérer et localiser le poste d'échantillonnage;
- 2) Repérer les points candidats;
- 3) Calculer l'aire des cellules Voronoï des points candidats;
- 4) Détruire la cellule Voronoï du poste d'échantillonnage;
- 5) Recalculer l'aire des cellules Voronoï de chaque points candidats;

Figure 5.10



Exemple d'un lissage hermitien. Le modèle au bas est la version lissée du modèle présenté au haut.

- 6) Faire la différence des aires calculées en 3 et 5;
- 7) Faire la sommation des aires volées;
- 8) Diviser l'aire volée d'une cellule Voronoï par la somme des aires volées. Ceci donne le poids (compris entre 0 et 1) pour un point candidat donné;
- 9) Multiplier le poids obtenu en 8 par la valeur  $Z$  du point candidat;
- 10) Sommer toutes les valeurs obtenues en 9. On obtient ainsi la valeur interpolée.

Une fois l'interpolation réalisée, le rendu visuel du modèle obtenu peut parfois donner l'impression d'avoir été taillé au couteau en raison de l'effet de facette provoqué par les triangles. Un lissage hermitien lors de l'interpolation permet de réduire quelque peu cet effet en adoucissant le relief. Ce lissage est appliqué au moyen d'une polynomiale toute simple.  $W$  qui est le poids associé à chacun des points voisins devient  $W_i$  après qu'on lui ait appliqué la formule  $3W^2 - 2W^3$ . La figure 5.10 présente un exemple de lissage hermitien. Le modèle du haut est le MNT original, tandis que celui du bas est sa version lissée.

La programmation de cet algorithme à l'aide d'un langage Orienté-Objet tel que Delphi se fait sans trop de difficultés et ne requiert qu'un peu plus de deux pages de code. En fait, la structure du langage se prête très bien au codage de ce genre d'algorithme.

Les avantages de la méthode d'interpolation "aire volée" sont nombreux. D'abord, elle garantit que la surface générée passera par tous les points de données. Elle exempte la surface de tout artefact ou discontinuité spatiale quelconque. Elle permet de traiter les distributions de données fortement anisotropiques. Enfin, elle assure une sélection adéquate de points candidats, même dans le cas d'une distribution de données fortement dispersées (Gold, 1989). Pour toutes ces raisons, cette méthode d'interpolation présente un potentiel d'applicabilité considérable.

### **5.5 TIN produit à partir d'un semis de points obtenu par photogrammétrie**

Prenons un ensemble de points  $N$  possédant chacun une valeur d'élévation. Pour générer un TIN, il suffit de produire une triangulation Delaunay de ces points. La triangulation Delaunay est préférée car elle produit des triangles dont la longueur des arêtes est minimisée, ce qui con-

fère au modèle un rendu visuel plus agréable.

*“Évidemment la densité et l’emplacement des points échantillonnés sur la surface du terrain doivent faire en sorte qu’ils génèrent un modèle fidèle de celui-ci en fonction de la précision désirée pour l’application envisagée”*(Boulianne, 1999).

Pour densifier le semis de points, on aura recours à l’interpolation. L’interpolation linéaire offerte à même la triangulation Delaunay peut être utilisée, mais également d’autres méthodes telle l’interpolation “aire volée”.

### **5.6 TIN produit à partir de courbes d’élévation**

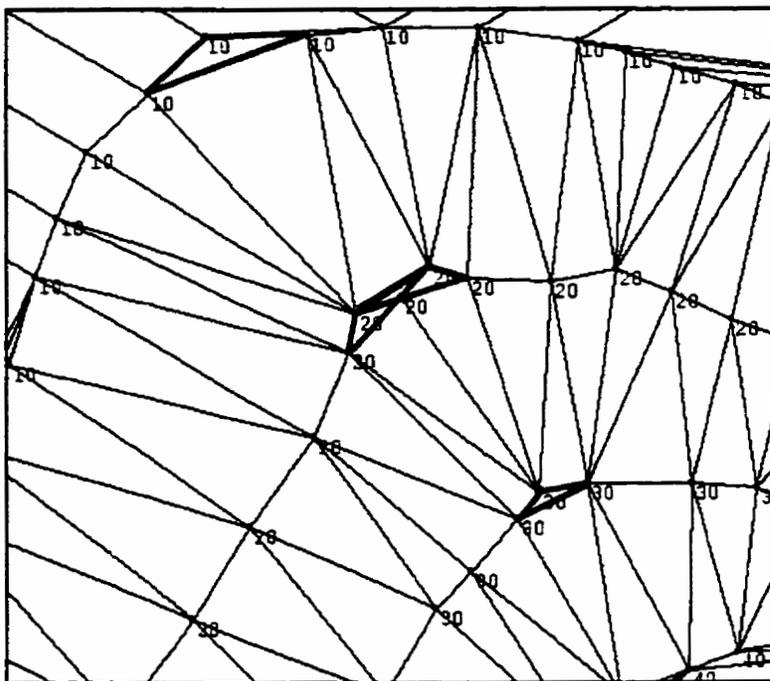
Les courbes de niveau sont la forme la plus fréquente d’information altimétrique utilisée pour la conception d’un TIN. Ceci s’explique en fonction de la grande accessibilité à cette source de données et au faible coût associé à son utilisation.

Pour produire un TIN, il suffit de générer une triangulation à partir des points composant les courbes. Cependant, la plupart des algorithmes utilisés pour produire une triangulation requièrent un ensemble de points comme donnée d’entrée et non des courbes de niveau. En réalité, les courbes ne doivent pas être considérées au même titre qu’un ensemble de points, mais plutôt comme des polygones superposés les uns sur les autres. Van Kreveld (1997) mentionne : *“Each region (contour lines) can be seen as a polygon with holes, [...]”*

Garcia et al. (1992) et Christensen (1987) ont élaboré certaines approches pour trianguler efficacement les courbes de niveau. Toutefois, un problème demeure avec ce type de donnée. Il y a sur-échantillonnage de données sur les courbes et aucun point entre celles-ci. Le sur-échantillonnage peut toutefois être corrigé en employant un filtre permettant de réduire le nombre de points sur la courbe. En ce qui concerne l’ajout de points entre les courbes, un algorithme d’interpolation linéaire peut très bien s’acquitter de cette tâche. Mentionnons également qu’aux endroits où les courbes s’incurvent plus fortement, il se produit une série de triangles plats dû au fait que tous les sommets des triangles sont situés sur la même courbe de niveau. Ce pro-

blème bien connu est appelé le problème des “triangles plats” ou “l’effet du gâteau de mariage” (figure 5.11). Devant un tel cas, les méthodes d’interpolation linéaire sont incapables d’estimer de nouvelles valeurs sur la surface de ces triangles. Ceci donne un effet particulièrement

**Figure 5.11**



Exemple du problème des “triangles plats”.

indésirable au TIN lorsque visualisé avec un effet d’éclairage virtuel (*hill shading*). Cependant, quelques techniques ont été développées pour corriger ce problème dont celle de Garcia et al. (1992). Nous avons également développé la nôtre dans le cadre de cette recherche et elle est présentée à la section 6. Notre méthode a l’avantage d’être très simple et de garantir la suppression complète des triangles plats et ce, même à l’intérieur de la courbe de niveau supérieur (sommet). Contrairement

à la méthode développée par Garcia et al. (1992), notre méthode enrichit du même coup le modèle TIN avec de l’information topographique supplémentaire provenant de nouvelles valeurs interpolées le long des branches et du tronc de l’axe médian. Ces nouvelles valeurs, une fois insérées dans le modèle, offriront une représentation beaucoup plus juste des lignes de crête et des talwegs, là où justement étaient formées les triangles plats.

## **6. ENRICHISSEMENT DE MODÈLES NUMÉRIQUES D'ÉLEVATION (TIN) PAR INTERPOLATION LE LONG DE L'AXE MÉDIAN**

### **6.1 Pourquoi enrichir un modèle numérique d'élévation ?**

Quelques auteurs, dont Aumann et al.(1991), se sont intéressés à l'enrichissement de modèles numériques d'élévation. En règle générale, on choisit d'enrichir un modèle afin d'y inclure de l'information topographique additionnelle tels que des lignes de faille, des lignes de crête, des talwegs, etc. Lorsque l'ajout de telles structures topographiques se fait dans un modèle de type TIN, on parle alors de "triangulation Delaunay contrainte". En fait, on force le réseau de triangles à s'ajuster à ces nouvelles structures.

Il existe également d'autres raisons pour enrichir un modèle numérique d'élévation, notamment lorsque l'on veut éliminer le problème des "triangles plats" dans un modèle de type TIN. Van Kreveld (1997) souligne ce problème :

*"In the GIS literature, a couple of approaches to triangulate between contour lines have been described. One of the problems with the constrained Delaunay triangulation and some of the other methods is that they may create horizontal triangles. [...]. It is especially undesirable when visualizing the terrain with the use of hill shading."*

Il serait donc particulièrement souhaitable de développer une méthode permettant de solutionner ce problème et, du même coup, d'inclure dans le modèle de l'information topographique additionnelle. On ferait ainsi une pierre deux coups. La technique d'enrichissement qui suit le permet. Elle a été développée dans le cadre de cette recherche et utilise l'axe médian.

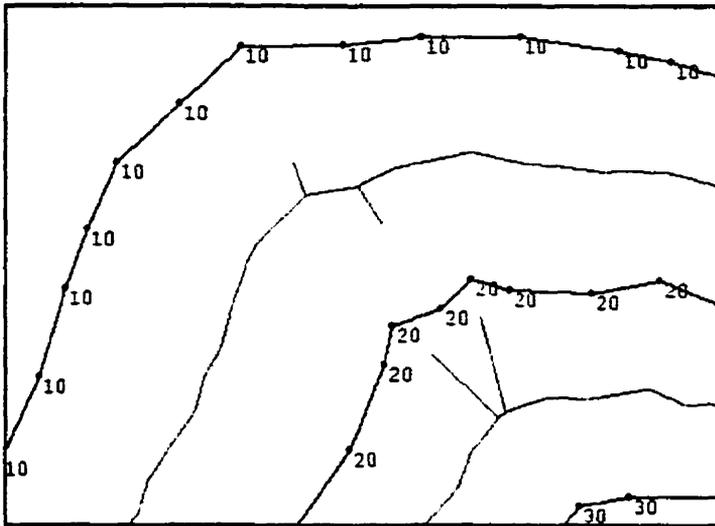
## **6.2 Méthode d'enrichissement par interpolation le long de l'axe médian**

### **6.2.1 Présentation**

À partir d'un fichier de courbes de niveau, on produit la triangulation Delaunay et le diagramme Voronoï de tous les vertex des courbes. Ensuite, en utilisant l'algorithme de Gold (1999), on isole la croûte (courbes de niveau) ainsi que l'axe médian. Pour que la croûte soit

extraite sans cassure, l'échantillonnage doit répondre aux critères définis par Amenta et al. (1998). On remarque sur la figure 6.1 que l'axe médian sépare chacune des courbes de niveau et qu'à certains endroits, là où les courbes s'incurvent, des branches apparaissent sur le tronc de cet axe médian. Elles indiquent l'emplacement de talus ou de petites dépressions suivant

**Figure 6.1**



Les branches de l'axe médian indiquent l'emplacement de talus ou de petites dépressions suivant l'élévation des courbes de niveaux supérieur et inférieur.

l'élévation des courbes de niveaux supérieur et inférieur. C'est à ces endroits que les cas de "triangles plats" surviennent.

L'idée maîtresse de cette méthode d'enrichissement est d'interpoler de nouvelles valeurs d'élévation sur les vertex de l'axe médian, grâce à une méthode d'interpolation spécialement développée, et d'insérer ensuite ces nouvelles valeurs dans la triangulation. Ceci permettra d'abord de corriger le problème des "triangles plats"

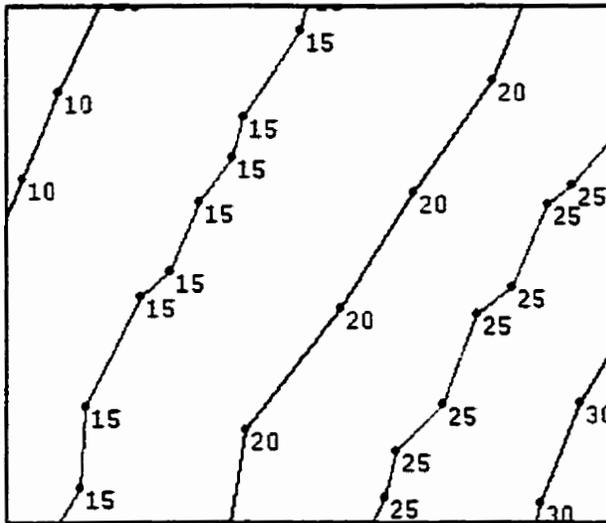
car de nouveaux triangles seront créés dans les zones critiques. Également, l'interpolation permettra d'ajouter dans le modèle de l'information topographique additionnelle et, de ce fait, de mieux prendre en compte les éléments structuraux du terrain, telles les lignes de crête et les vallées.

L'algorithme développé pour l'interpolation réalise deux opérations principales. D'abord, il calcule les valeurs d'élévation sur le tronc de l'axe médian, puis il estime ensuite les valeurs sur ses branches. Ces deux opérations sont distinctes l'une de l'autre car les calculs réalisés sont totalement différents.

Pour estimer les valeurs d'élévation sur le tronc de l'axe médian, rien de plus facile. Comme le tronc sépare chaque paire de courbes de niveau, on assigne aux vertex du tronc la moyenne des élévations des deux courbes de niveau voisines (figure 6.2).

L'estimation de valeurs sur les branches de l'axe est une opération plus complexe en soit. On ne peut interpoler avec une méthode traditionnelle comme l'aire volée car les valeurs d'élévation au sommet des triangles sont tous les mêmes. Pour permettre l'interpolation dans cette zone de triangles plats, une méthode particulière utilisant des ratios de cercles a été développée. On part du principe que tout au long d'une branche, il y a une relation linéaire entre le

**Figure 6.2**



Interpolation de valeurs sur le tronc de l'axe médian représenté ici en rouge.

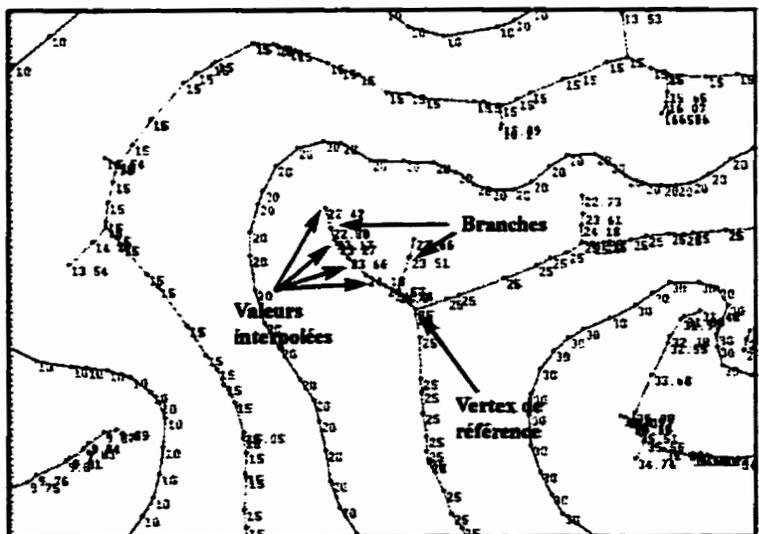
zone de dépression ou d'élévation.

Le cercle de référence utilisé est celui du vertex Voronoï, localisé à la jonction du tronc et de la branche de l'axe médian. La valeur de ce vertex est connue car elle a été obtenue lors de la première étape d'interpolation. Le reste de la procédure consiste à visiter chaque vertex de la branche en commençant par le vertex de jonction et en progres-

rayon des cercles circonscrits aux vertex de la branche et la valeur d'élévation associée à ces vertex. En d'autres mots, plus le rayon du cercle circonscrit à un triangle est grand, plus la valeur d'élévation au vertex Voronoï associé (centre du dit cercle) sera élevée.

Pour pouvoir estimer des valeurs à n'importe quel vertex le long d'une branche, il faut définir un vertex de référence dont la valeur d'élévation est connue (figure 6.3). Il faut aussi connaître la valeur de la courbe de niveau située à l'extrémité de la branche afin de déterminer si cette branche représente une

**Figure 6.3**

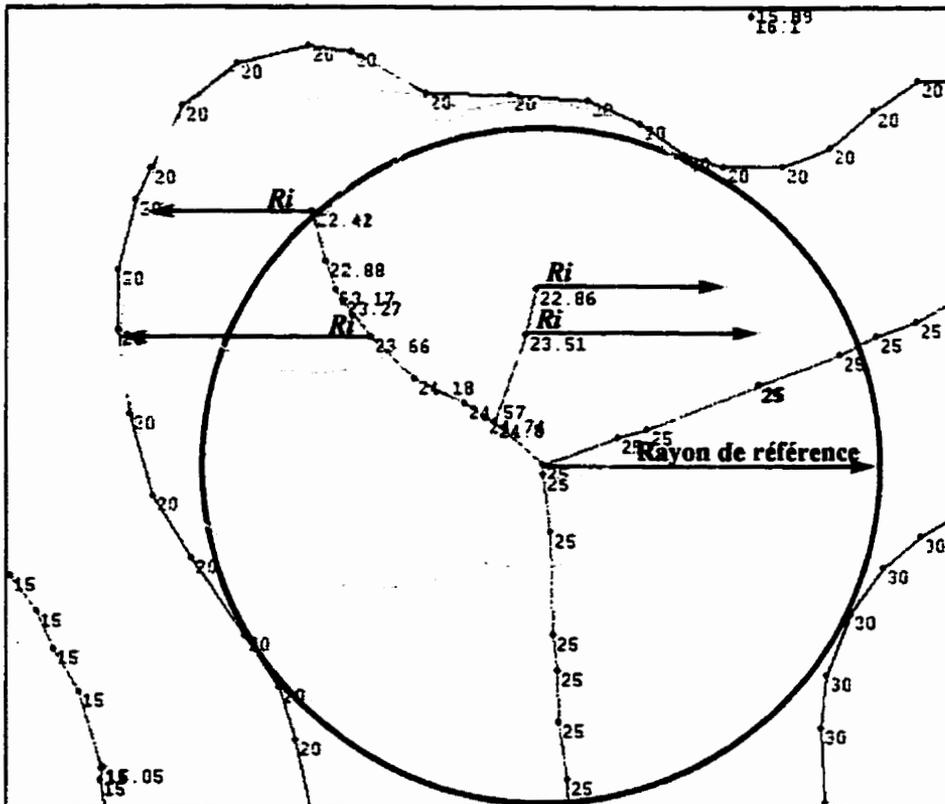


Première approche pour l'interpolation de valeurs sur les branches de l'axe médian.

sant vers les feuilles. On compare alors le rayon du cercle associé à chaque vertex nommé  $R_i$  au rayon du cercle de référence. La figure 6.4 montre les cercles pour une branche donnée. Par souci de lisibilité, seulement quelques cercles sont présentés.

Ceci est la première méthode que nous avons considérée pour l'interpolation de valeurs sur les branches de l'axe médian. Nous avons également exploré une seconde approche afin de vali-

**Figure 6.4**

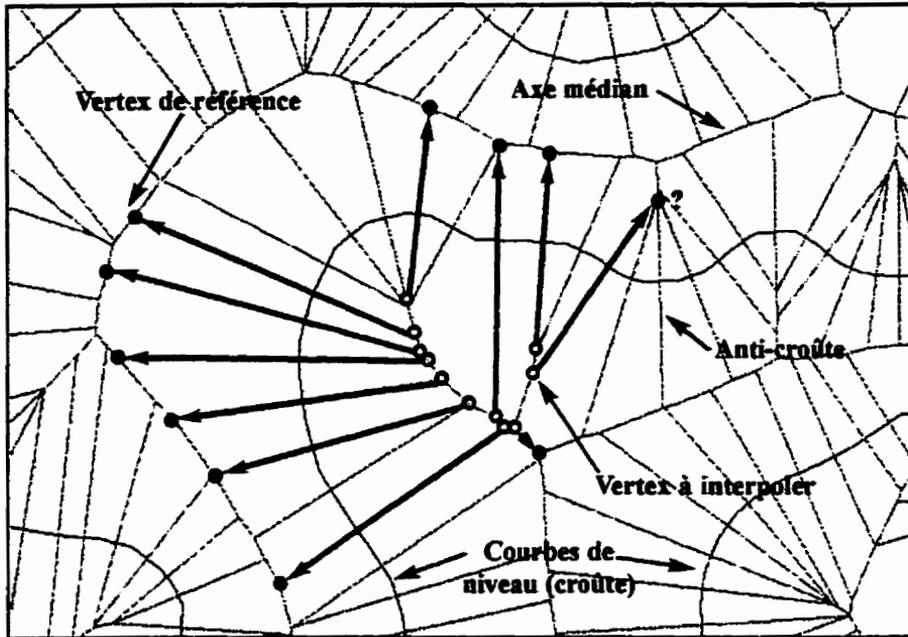


Les cercles de la première approche d'interpolation sur les branches de l'axe médian.

Cette deuxième approche utilise un cercle de référence différent pour chaque vertex de la branche. La progression dans la branche, à partir du point de jonction tronc-branche, demeure la même. Toutefois, à mesure que l'on change de vertex, on choisit un nouveau cercle de référence. Les vertex de référence sont situés soit sur le tronc de l'axe médian à laquelle la branche se rattache, soit sur le tronc passant entre les deux courbes de niveau, situées face aux feuilles de la branche (figure 6.5). La figure 6.6 présente les cercles de référence utilisés et ceux des vertex interpolés. Par souci de lisibilité, nous avons limité ici le nombre de cercles présentés.

der les résultats obtenus par la première méthode. Cette deuxième approche ne permet toutefois pas d'interpoler sur tous les vertex des branches. Nous expliquerons pourquoi un peu plus loin. Nous avons néanmoins décidé de l'utiliser pour comparer les résultats de l'interpolation.

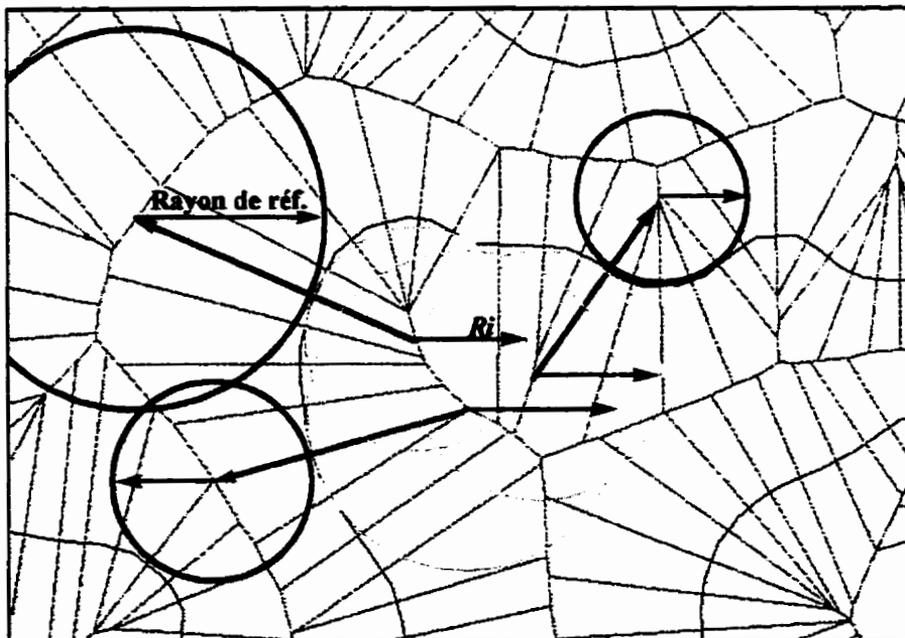
Figure 6.5



Deuxième approche pour l'interpolation de valeurs sur les branches de l'axe médian.

qui servira de référence. Son rayon sera comparé au vertex de la branche et une valeur d'élévation pourra ensuite être calculée (figure 6.6).

Figure 6.6



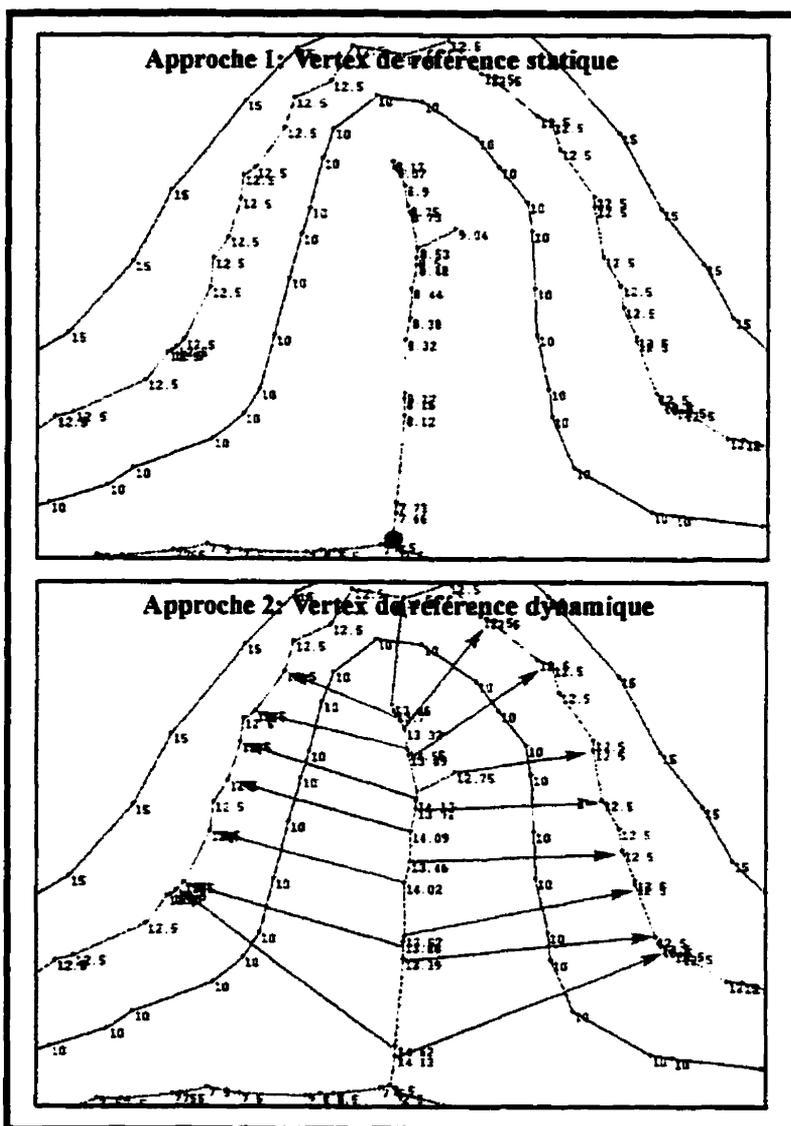
Les cercles de la deuxième approche d'interpolation sur les branches de l'axe médian.

Pour chaque vertex de la branche, on évalue si un segment Voronoï, ne faisant pas parti de l'axe médian (Gold, 1999 le nomme "anti-croûte" du fait qu'il croise la croûte), connecte le vertex à estimer à un autre vertex situé sur le tronc et dont la valeur d'élévation est connue. C'est ce vertex situé sur le tronc

Comme nous l'avons mentionné, certains vertex des branches ne peuvent être interpolés. Ceci est dû au fait que la valeur des vertex de référence qu'ils utilisent n'est pas connue, car les segments "anti-croûte" ne relient pas le vertex à estimer au tronc, mais plutôt à une branche de l'axe

médian. Dans un tel cas, il s'avère donc impossible d'estimer une valeur d'élévation. La figure 6.7 compare les valeurs obtenues avec les deux méthodes. Sur l'image du haut, le point vert indique le vertex de référence utilisé. Sur l'image du bas, les flèches vertes montrent les vertex de référence utilisés pour chacun des vertex de la branche.

**Figure 6.7**



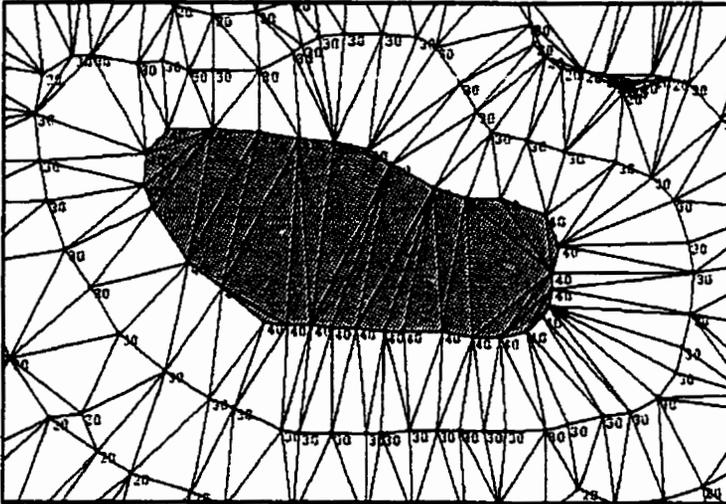
Comparaison des valeurs obtenues à l'aide des deux approches d'interpolation pour une branche de l'axe médian représentant un talweg (ligne qui relie les points les plus bas d'une vallée).

Après observation, nous pouvons affirmer que les valeurs estimées sont tout à fait farfelues, lorsque nous employons la deuxième approche et que nous considérons la branche (figure 6.7) comme représentant un talweg (ligne reliant les points les plus bas d'une vallée). On remarque que les valeurs de la branche dépassent l'élévation de la courbe supérieure de 10 mètres, ce qui est illogique car les valeurs devraient plutôt être comprises en 7.5 mètres (valeur à la base de la branche) et 10 mètres (valeur de la courbe supérieure). En somme, la première approche s'avère être la seule qui puisse fournir des résultats vraisemblables et c'est elle qui a été retenue pour cette recherche.

Il existe un cas spécial d'interpolation dont nous n'avons pas encore fait mention. Il s'agit de l'interpolation sur la portion de l'axe médian localisée à l'intérieur de la courbe de plus haute

élévation. En l'absence d'un point coté, toute la zone située au sommet du modèle constitue une zone critique où les triangles sont plats (figure 6.8). L'objectif est donc d'estimer des valeurs afin de produire une représentation naturelle du sommet. La technique que nous avons développée emploie la même méthode utilisée pour l'interpolation le long des branches, à la

**Figure 6.8**

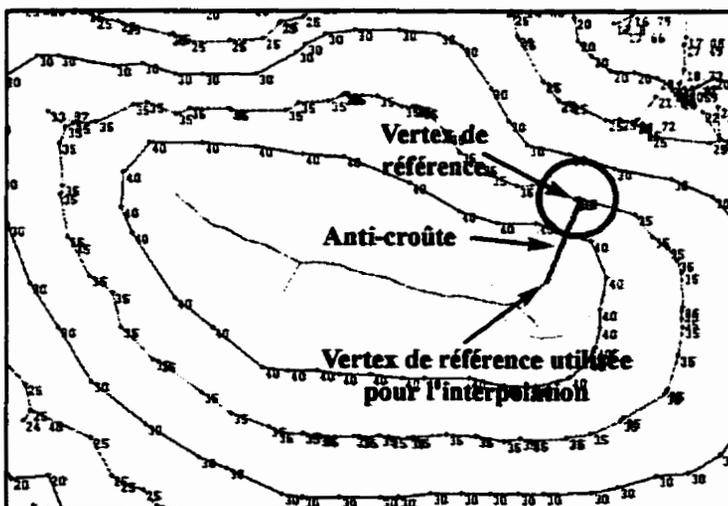


En l'absence d'un point coté, tout le sommet du modèle constitue en fait une zone critique où les triangles sont plats.

différence qu'aucun vertex de référence n'existe dans ce cas-ci. Nous devons donc d'abord calculer la valeur d'un vertex de référence. Celui-ci doit être situé à une des extrémités (feuille) de cet axe médian. La valeur du vertex de référence est obtenue par ratio de cercles, en utilisant un vertex situé à l'extrémité du segment "anti-croûte" et dont la valeur est connue. Pour cette raison, cette étape de l'interpolation doit être réalisée

une fois seulement que les valeurs auront été calculées sur le tronc et toutes les autres branches de l'axe médian. La figure 6.9 présente la procédure. Une fois la valeur du vertex de référence obtenue, on pourra ensuite estimer les valeurs d'élévation tout au long de l'axe, comme s'il s'agissait d'une simple branche.

**Figure 6.9**



Calcul de la valeur du vertex de référence pour l'interpolation au sommet du modèle.

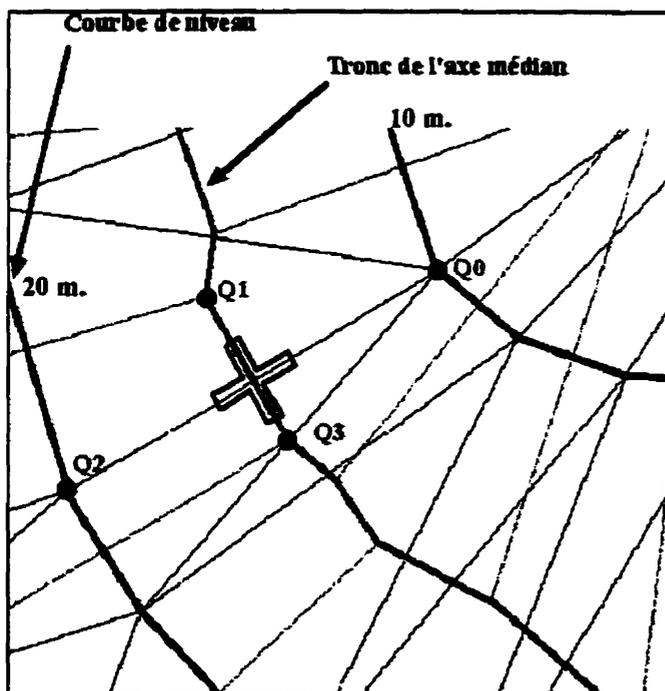
### 6.2.2 Description du code de programmation

Une fois les courbes de niveau triangulées et le diagramme de Voronoï produit, on débute l'enrichissement du modèle en choisissant la commande *Enrich TIN with skeleton* du menu *Edit* à l'intérieur

de l'application *Voronoi Toolkit*. Cette commande lance la série de procédures et de fonctions nécessaires à l'estimation des nouvelles valeurs. La première opération consiste à créer deux listes nommées *SkeletonPt* et *QList*. *SkeletonPt* contiendra les coordonnées des nouveaux vertex interpolés avant leur sauvegarde dans un fichier de sortie, tandis que *QList* contiendra les Quads situés à la base des branches de l'axe médian. On utilise ensuite *Scan* pour parcourir le maillage en vue d'interpoler la première série de valeurs sur le tronc principal de l'axe et de déterminer les Quads situés à la base de chacune des branches. La procédure utilisée pour cette opération se nomme *ScanningSkeletonPt* et reçoit, comme paramètre d'entrée, un edge provenant de *Scan*.

L'interpolation sur le tronc de l'axe est une opération simple en soit. À partir d'un edge envoyé par *Scan*, on applique le test de la croûte afin de déterminer si le dual de ce edge fait partie de l'axe médian. Le lecteur peut s'aider de la figure 6.10 pour suivre la démarche des opérations subséquentes. Sur cette figure, Q0, Q1, Q2 et Q3 sont des pointeurs vers les vertex de la courbe de niveau ou de l'axe médian. Leurs équivalents d'après la terminologie Quad-Edge sont: Q0= Edge.V; Q1= Edge.R.V; Q2= Edge.Sym.V; Q3= Edge.Sym.R.V.

Figure 6.10

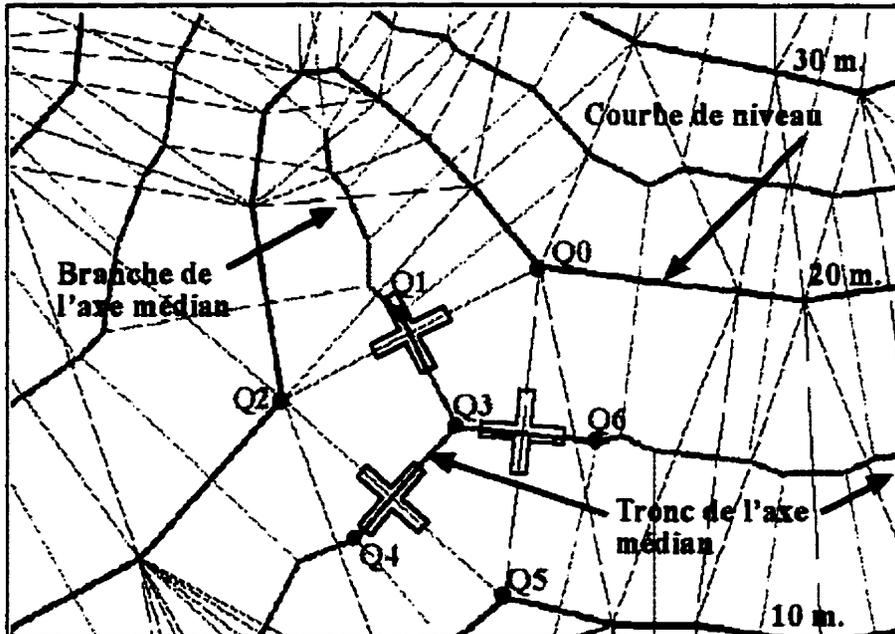


Repérage d'un segment du tronc de l'axe médian pour interpolation.

Débutons en appliquant le test de la croûte sur le segment  $Q0 \Rightarrow Q2$ . Si le test retourne faux, alors le segment Voronoï dual  $Q1 \Rightarrow Q3$  fait partie de l'axe médian. Mais pour appartenir au tronc de l'axe médian, les valeurs d'élévation aux extrémités du primal (Delaunay) doivent être différentes ( $Q0.Z \neq Q2.Z$ ) car le tronc sépare deux courbes de niveau. Si cette condition est remplie, comme sur la figure 6.10, alors le segment  $Q1 \Rightarrow Q3$  appartient bel et bien au tronc de l'axe médian. On assigne alors à  $Q1.Z$  et  $Q3.Z$  la moyenne des élévations des deux courbes de niveau voisines. Sur la figure

6.10, Q1.Z et Q3.Z vaudront 15. Une fois les valeurs assignées à Q1 et Q3, on leur applique un "flag" nommé "D1", indiquant qu'ils ont été traités. De plus, deux coordonnées de type *TvPoint* sont créées à partir de Q1 et Q3 et placées dans la liste *SkeletonPt*, en vue de leur

Figure 6.11



Reperage d'un vertex de jonction tronc-branche en vue de l'interpolation sur la branche.

sauvegarde future dans un fichier de sortie. On place également dans *SkeletonPt* les vertex Q0.V et Q2.V de la croûte car ils seront eux aussi inclus dans le fichier de sortie. L'objectif est de produire un modèle contenant les vertex originaux des courbes, auxquels on aura ajouté les vertex de l'axe médian. Men-

tionnons que le type *TvPoint* est défini par trois (3) champs de type réel étendu (X,Y,Z) et un champ de type entier (C) servant à stocker, au besoin, la valeur de couleur du vertex. Il est utilisé pour le stockage des coordonnées.

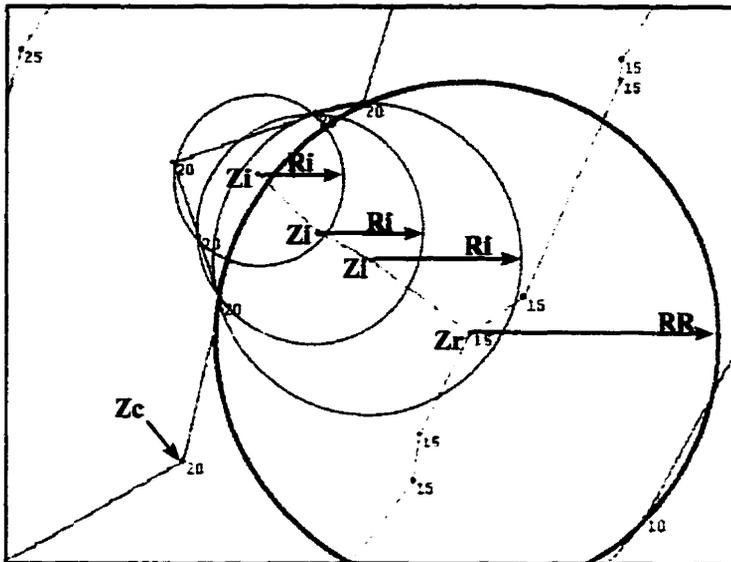
La procédure *ScanningSkeletonPt* extrait également au cours de cette opération les Quads situés à la jonction tronc-branche. Ceux-ci serviront de pointeurs de départ pour le parcours des branches. Le lecteur est invité ici à consulter la figure 6.11 pour une meilleure compréhension de la procédure de sélection des Quads. Mentionnons d'abord que, tout comme sur la figure 6.10, Q1, Q2, Q3, Q4, Q5 et Q6 sont des pointeurs vers des vertex des courbes ou de l'axe médian, le pointeur de référence est fixé à Q0 ou Edge. Quant aux autres pointeurs, voici leur définition d'après la terminologie Quad-Edge et par rapport à Q0. Q1=Edge.R.V; Q2=Edge.Sym.V ou Edge.Lnext.V; Q3=Edge.Sym.R.V; Q4=Edge.Lnext.Sym.R.V; Q5=Edge.Lnext.Sym.V ou Edge.Lnext.Lnext.V; Q6=Edge.Lnext.Lnext.R.V.

Pour trouver les vertex de jonction, on doit répondre aux critères suivants. Par rapport à la figure 6.11, si le test de la croûte pour les segments  $Q0 \Rightarrow Q2$ ,  $Q2 \Rightarrow Q5$  et  $Q0 \Rightarrow Q5$  retourne faux pour chacun, ceci assure que les segments Voronoï dual, soit  $Q3 \Rightarrow Q1$ ,  $Q4 \Rightarrow Q3$  et  $Q6 \Rightarrow Q3$ , se rattachent à un seul vertex. Dans ce cas-ci la jonction est  $Q3$ . Cette condition est toutefois insuffisante pour garantir que le vertex est bel et bien situé à la jonction du tronc et d'une branche. En fait, il pourrait très bien s'agir d'un vertex situé sur une branche de l'axe comportant deux embranchements. On doit ajouter à la condition précédente, les deux conditions suivantes. La valeur d'élévation au vertex  $Q0$  doit être égale à celle de  $Q2$  pour garantir que l'on est à l'entrée d'une zone de triangles plats, plus précisément où passe la branche. De plus, on doit garantir que le tronc de l'axe médian existe. Pour ce faire, la valeur de  $Q2.Z$  doit être différente de celle à  $Q5.Z$ , mais également entre  $Q5.Z$  et  $Q0.Z$ . Une fois le vertex de jonction tronc-branche trouvé ( $Q3$ ), on place le Quad (pointeur) situé à la base de la branche dans *QList*. Dans ce cas-ci, c'est le pointeur opposé à  $Q1$  qui est retenu, soit celui pointant vers  $Q3$ .

Avant de passer à la deuxième phase du processus, soit l'interpolation sur les branches de l'axe, on doit s'assurer qu'il n'y a pas de pointeurs dédoublés dans la liste *QList*. En effet, bien que *Scan* traite une seule fois chaque edge, un vertex demeure rattaché à deux edges et ce dernier a pu être traité à deux reprises. Une fonction nommée *RemoveDoubles* a été conçue afin de parcourir une liste et éliminer tous les enregistrements en double. Ce n'est qu'une fois *QList* traitée par cette fonction qu'on pourra procéder à l'interpolation de valeurs sur les branches de l'axe.

*QList* est acheminée à la procédure *InterpolateAlongBranches*. Les pointeurs sont extraits de la liste un à un. On réassigne le premier pointeur afin qu'il pointe en direction des feuilles de la branche et on le place sur une liste spéciale nommée *QStack* faisant office de pile. À cette étape, le rayon du cercle et la valeur d'élévation du vertex de jonction ont été placés dans les variables respectives *RR* et *Zr* que nous décrirons plus loin. On progresse ensuite dans la branche par rotation successive autour des vertex. Lorsque l'on identifie un embranchement, on place dans *QStack* le Quad situé à la base de cette nouvelle branche et on poursuit l'ascension dans la branche. Lorsque l'extrémité de la branche est atteinte, la fonction *FindTerminalNode* retourne vrai et on récupère un Quad dans *QStack*. L'opération se prolonge jusqu'à ce que la pile soit complètement vide, c'est à dire lorsque tous les vertex de la branche ont été visités. Pour le traitement d'une autre branche, on retourne dans *QList* chercher un

Figure 6.12



Variables utilisées pour l'interpolation sur les branches de l'axe médian.

deuxième pointeur et on reprend le processus énoncé précédemment.

L'interpolation des valeurs se fait en comparant le rayon du cercle circonscrit, centré à un vertex donné, avec le rayon du cercle centré au vertex de référence (jonction tronc-branche). Ceci implique que l'on doit connaître les valeurs suivantes avant de débiter le parcours de la branche (figure 6.12):

**Zr** qui est la valeur d'élévation du vertex de référence;

**Zc** qui est la valeur d'élévation de la courbe de niveau localisée face à la branche;

**RR** qui est le rayon du cercle de référence .

On doit déterminer pour chacun des vertex de la branche :

**Ri** qui est le rayon du cercle au vertex en cours d'évaluation;

**R** qui est le rapport entre **Ri** et **RR** ( $R_i/RR$ );

**Zi** est la valeur d'élévation au vertex en cours d'évaluation.

À chacun des vertex visités, on estime la valeur d'élévation par la formule suivante :

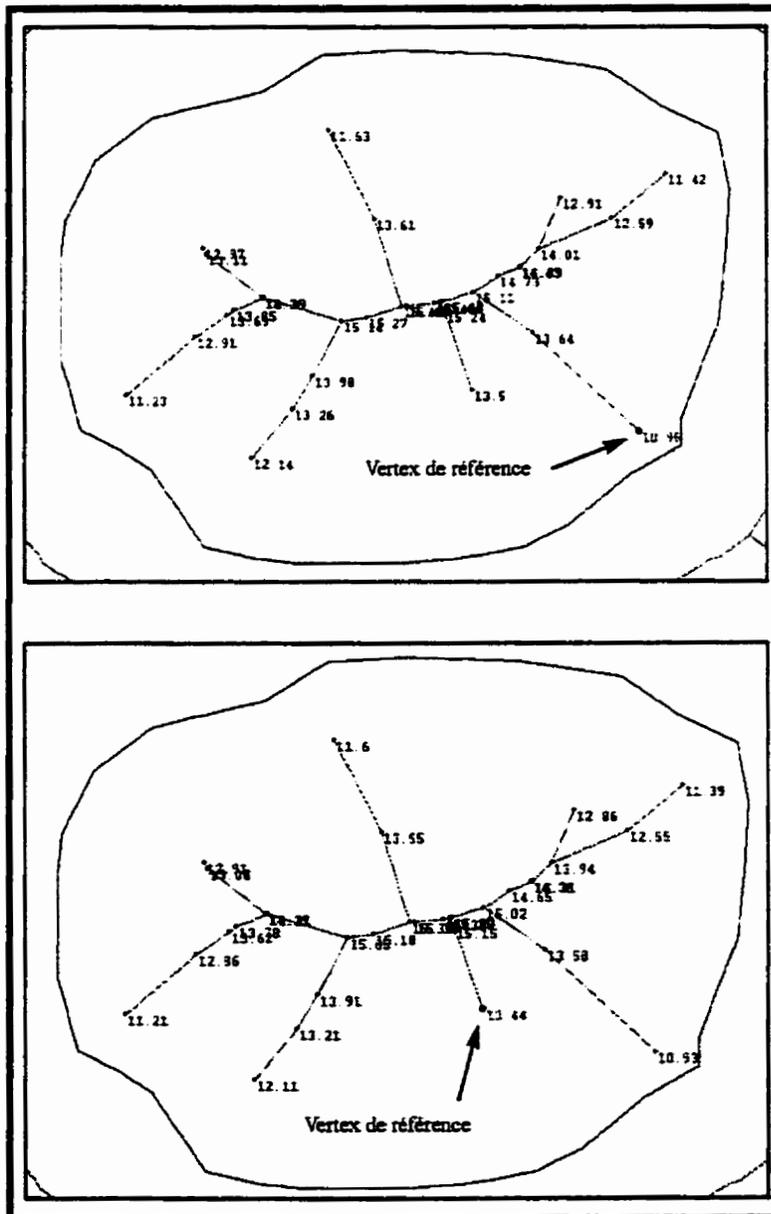
$$Z_i = (R * Z_r) + ((1-R) * Z_c)$$

Une fois la valeur d'un vertex obtenue, on lui appose une étiquette nommée "D2" pour indiquer qu'il a été traité. On crée également un objet de type *TvPoint* avec la coordonnée de ce point et on place ce nouvel objet dans la liste *SkeletonPt* en vue de sa sauvegarde dans le fichier de sortie.

À cette étape de l'enrichissement du modèle numérique, de nouvelles valeurs ont été inter-

polées tant sur le tronc que les branches de l'axe médian. En l'absence d'un point coté à l'intérieur de la courbe d'élévation supérieure, ce qui est fréquent, aucune valeur n'aura été calculée sur cette section de l'axe médian. Il en va de même pour les portions de l'axe médian situées à l'extérieur du modèle. Le cas échéant, un nouvel appel à *Scan* est requis. Ce problème a été abordé plus tôt (figure 6.8).

Figure 6.13



À peu de chose près, les valeurs obtenues par interpolation au sommet du modèle demeurent sensiblement les mêmes, quelque soit le vertex de référence utilisé.

*Scan* est donc appelé une seconde fois. Chaque edge qu'il traite est envoyé à une procédure nommée *ScanningHill-TopVertex* dont l'objectif est de trouver un vertex de référence (non traité) situé sur une feuille de l'axe médian et d'estimer sa valeur (figure 6.9). La procédure *FindTerminalNode* sera employée pour sélectionner cette feuille. Ce n'est qu'une fois la valeur du vertex obtenue que nous pourrons lancer la procédure *InterpolateAlong-Branchs* pour estimer les valeurs des autres vertex de la branche.

Nous aimerions attirer l'attention du lecteur sur le fait que le choix du vertex de référence n'a ici aucune importance. En fait, toutes les feuilles peuvent potentiellement être des vertex de référence et peu importe

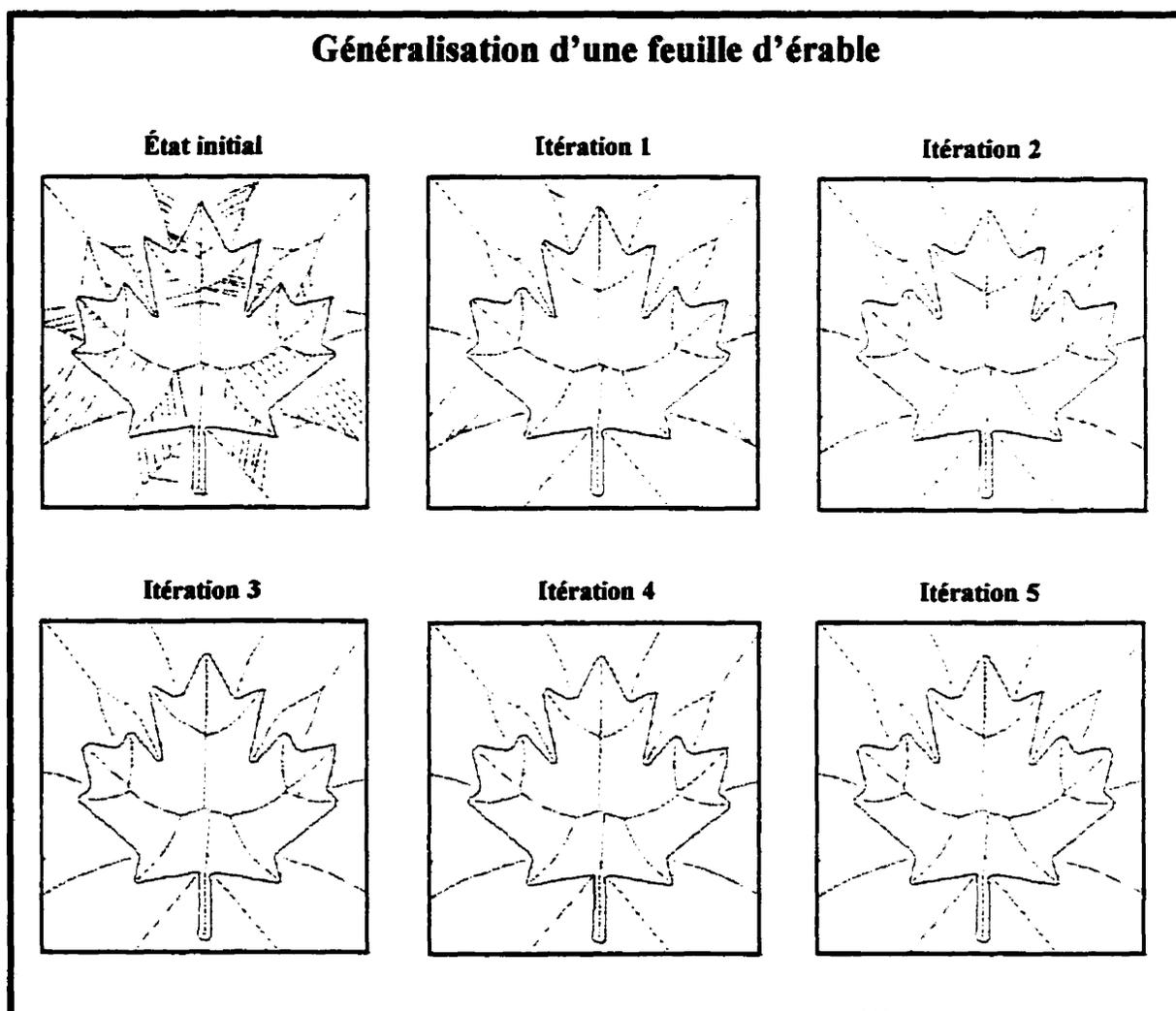
celui qui sera choisi, les valeurs obtenues sur l'ensemble de l'axe seront les mêmes. Ceci peut sembler pour le moins inusité, surtout si la valeur d'élévation et le rayon du cercle ne sont pas les mêmes d'un vertex de référence à l'autre. Il existerait donc une relation linéaire entre les valeurs des vertex de l'axe. Bien que la figure 6.13 tende à le démontrer, cette observation n'a toutefois pas été confirmée mathématiquement.

## 7. ÉVALUATION DES ALGORITHMES DÉVELOPPÉS

### 7.1 Généralisation par rétraction de l'axe médian

Les performances de cet algorithme ont été évaluées en fonction du temps requis pour déplacer sur les cercles un nombre déterminé de vertex. Le temps de traitement varie aussi avec la taille du fichier. En effet, plus le fichier est volumineux, plus l'algorithme *Scan* requiert du temps pour parcourir le maillage. Cependant, il n'y a pas de relation directe entre la taille d'un fichier et le nombre de vertex qui peuvent potentiellement être déplacés. Comme nous l'avons mentionné à la section 3, un vertex ne peut être déplacé que s'il répond à certains critères. D'abord, il doit faire partie de la croûte et, deuxièmement, il ne doit pas être un vertex de jonc-

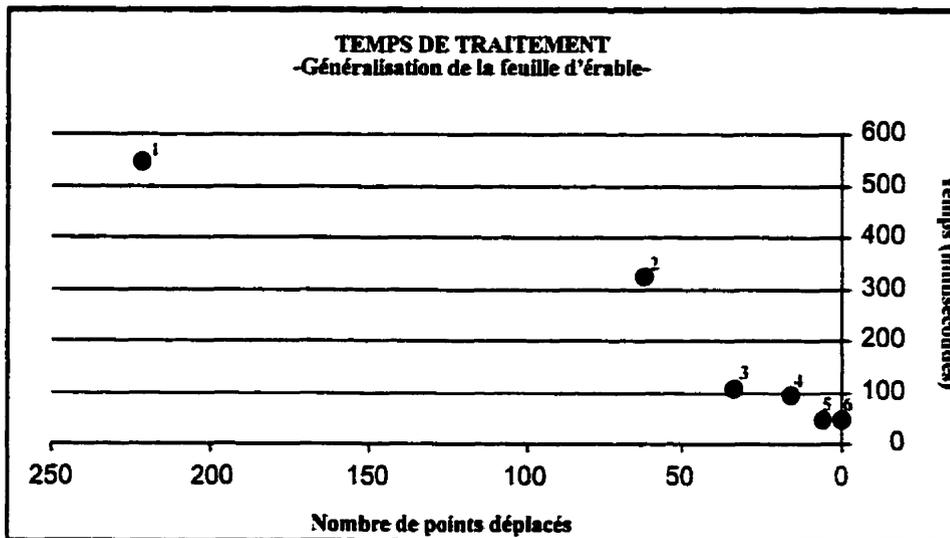
Figure 7.1



Étapes successives de la généralisation de la feuille d'érable. Les résultats obtenus après chacune des cinq itérations sont présentés.

tion (nœud) entre trois (3) segments de la croûte .

**Graphique 7.1**



Variation du temps de traitement et du nombre de vertex déplacés pour chacune des cinq itérations nécessaires à la généralisation complète de la feuille d'érable.

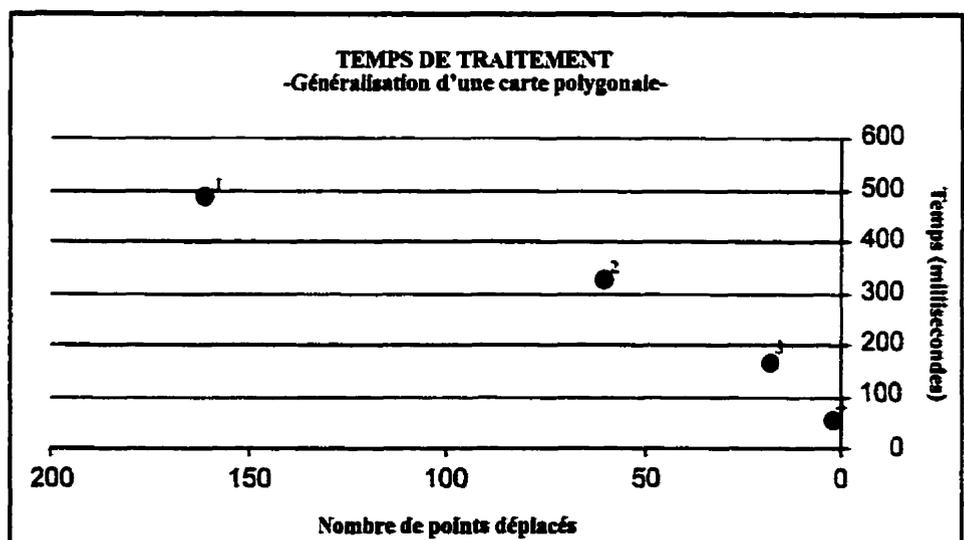
Les tests ont été réalisés sur un Pentium 200 MMX possédant 64 méga-octets de mémoire vive et 6.4 giga-octets d'espace disque. Les temps de calculs ont été obtenus à l'aide d'un composant

(VCL) Delphi nommé *Tchrono* intégré dans le programme. Ce composant permet de démarrer et d'arrêter un chronomètre à certains endroits précis dans le code.

Le premier fichier avec lequel l'algorithme a été évalué est une image représentant une feuille d'érable. Les coordonnées constituant l'image ont été extraites par filtrage à partir d'une version matricielle de la feuille d'érable. Le fichier comporte au total 350 vertex.

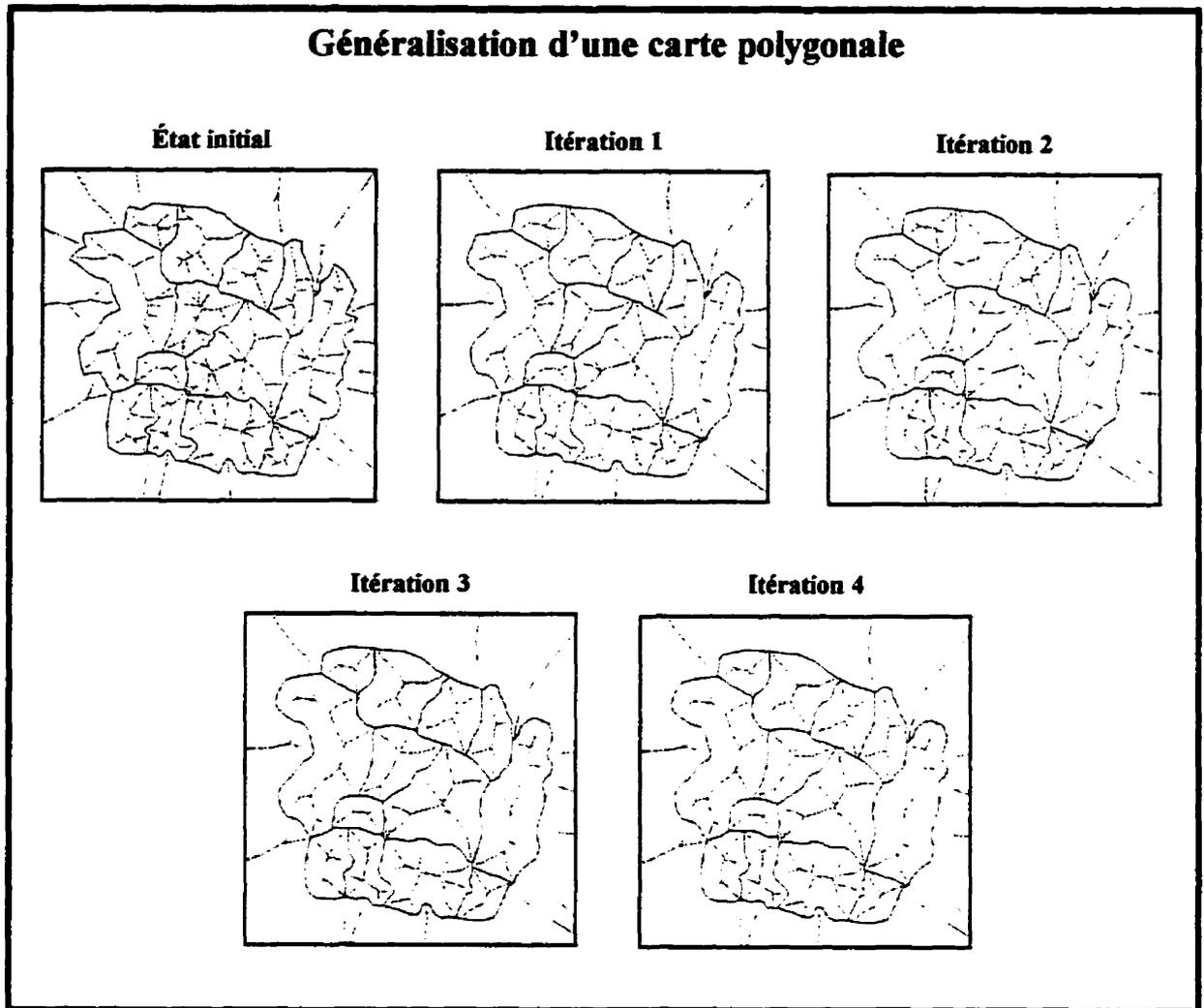
Comme l'algorithme de généralisation fonctionne de manière itérative, nous avons appliqué cinq itérations au fichier, soit jusqu'à ce qu'il n'y ait plus de vertex à déplacer. La figure 7.1 montre la

**Graphique 7.2**



Variation du temps de traitement et du nombre de vertex déplacés pour chacune des quatre itérations nécessaires à la généralisation complète d'une carte polygonale.

Figure 7.2



Étapes successives pour la généralisation d'une carte polygonale. Les résultats obtenus après chacune des quatre itérations sont présentés.

modification de la croûte après chacune des itérations. Le graphique 7.1 présente l'évolution des temps de calcul en fonction du nombre de vertex déplacés. On peut observer que les temps de traitement diminuent après chacune des itérations, du fait que le nombre de vertex cocirculaires s'accroît.

Au cours de la première itération, 220 vertex sont déplacés en 550 millisecondes, 61 vertex en 330 millisecondes pour la deuxième itération, 33 vertex en 110 millisecondes pour la troisième itération, 14 vertex en 100 millisecondes pour la quatrième itération, 4 vertex en 50 millisecondes pour la cinquième itération et, finalement, aucun vertex à la sixième itération n'a été déplacé. Le temps écoulé lors de cette dernière itération a été de 50 millisecondes, ce qui cor-

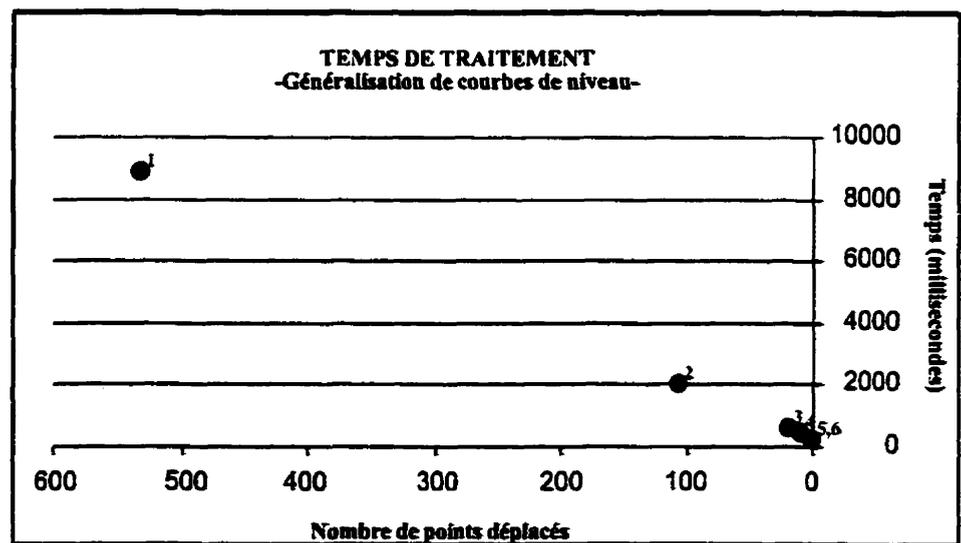
respond approximativement au temps nécessaire pour parcourir la triangulation.

Nous avons également évalué l'algorithme sur une petite carte polygonale conçue par l'auteur et constituée de 344 vertex. Le graphique 7.2 présente les temps de traitements qui ont été nécessaires à sa généralisation. Au cours de la première itération, 161 vertex ont été déplacés en 490 millisecondes, 60 vertex en 330 millisecondes lors de la deuxième itération, 18 vertex en 170 millisecondes lors de la troisième itération et, enfin, 1 vertex a été déplacé à la quatrième itération dans un temps de 60 millisecondes. La figure 7.2 montre les modifications apportées à la croûte au cours de chaque itération.

Le dernier fichier utilisé pour évaluer l'algorithme est une carte de courbes de niveau de la région du Lac Killarney (Missouri, États-Unis), tirée de Miller et al.(1989). Ce fichier comporte au total 2169 vertex. Il a été obtenu avec le même procédé que pour la feuille d'érable, soit par numérisation et ensuite extraction des coordonnées par filtrage. Bien que ce fichier ait été traité en totalité, nous n'en présenterons qu'une portion en raison de sa grande taille, ceci afin de faire ressortir clairement les modifications apportées par la généralisation. Le graphique 7.3 présente les temps de calcul pour chaque itération, alors que la figure 7.3 montre les résultats de la généralisation. En détail, 532 vertex ont été déplacés en 8950 millisecondes au cours de la première itération, 107 vertex en 2040 millisecondes lors de la deuxième itération, 19 vertex en 600 mil-

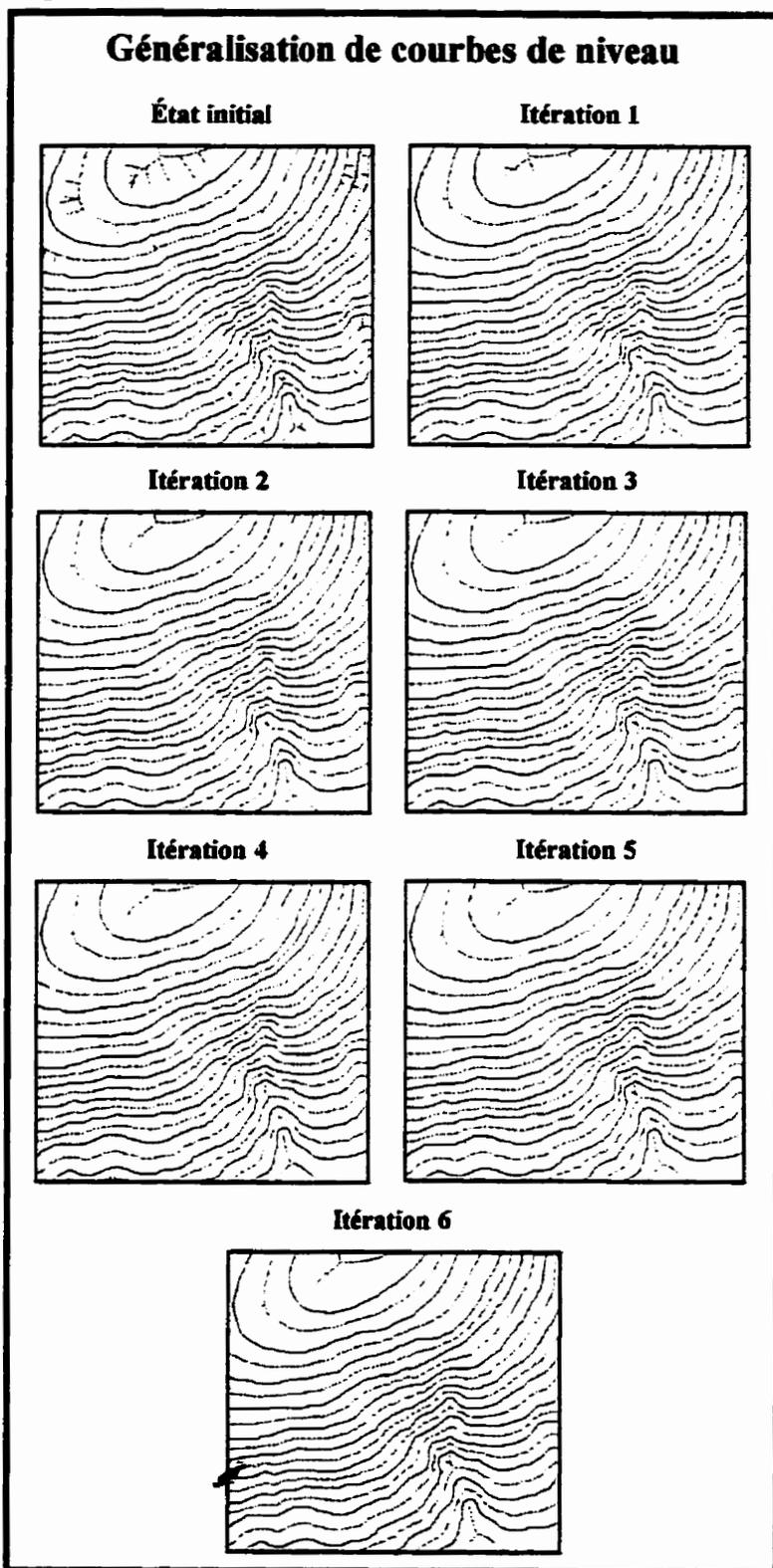
lisecondes à la troisième itération, 9 vertex en 440 millisecondes à la quatrième, 2 vertex en 270 millisecondes à la cinquième et enfin, encore 2 vertex à la sixième.

**Graphique 7.3**



Variation du temps de traitement et du nombre de vertex déplacés pour chacune des six itérations nécessaires à la généralisation complète d'un fichier de courbes de niveau.

Figure 7.3



Étapes successives pour la généralisation d'un fichier de courbes de niveau. Les résultats obtenus après chacune des six itérations sont présentés.

Source de la carte: Miller et al., 1989

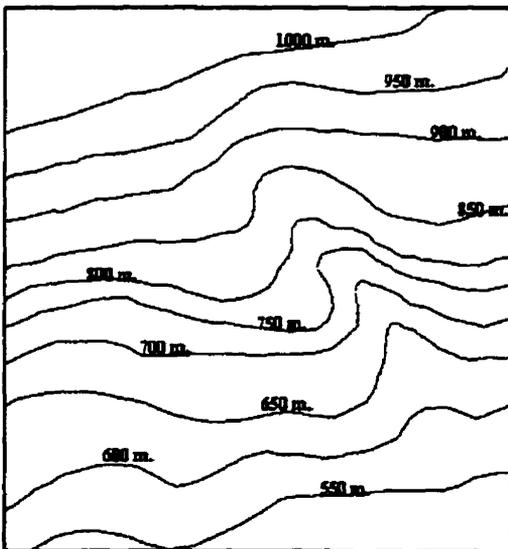
On note qu'entre la cinquième et la sixième itération, aucun nouveau vertex n'a été déplacé. En fait, il existe des cas où des vertex ne peuvent être placés sur les arcs de cercle parce qu'il existe déjà dans le maillage un autre vertex à l'intérieur de la tolérance fixée par une fonction appelée *Hit*. Cette fonction s'assure, lors de l'insertion d'un vertex, que celui-ci n'entrera pas en collision avec aucun autre. La tolérance est fixée à 0,001 unité à l'intérieur de notre code. Ceci explique également que le nombre total de vertex ait diminué après la généralisation. Par exemple, en employant la tolérance habituelle de 0.001, la feuille d'érable, qui possédait 350 vertex à l'origine, en a perdu 72 une fois la généralisation complétée (5 itérations). Avec une tolérance de 0.0001 cette fois, le nombre de vertex abandonnés diminue à 13, toujours après 5 itérations. En réalité, ce qui se produit, c'est qu'un vertex donné est retiré de la triangulation, mais ne peut être réin-

séré sur l'arc d'un cercle à cause de la tolérance fixée. Le bon côté de la chose, c'est qu'en employant une tolérance de 0.001, on réduit du même coup le nombre de points nécessaires pour représenter la courbe. Cependant, si le processus de généralisation était automatisée, c'est-à-dire si toutes les itérations étaient exécutées les unes après les autres, un tel cas engendrerait une boucle sans fin à l'intérieur de l'algorithme qui est prévu pour se terminer une fois que tous les vertex sont sur les cercles. Mais, dans le cas présent, les itérations sont déclenchées les unes après les autres par l'utilisateur. Il revient donc à ce dernier de mettre fin à la généralisation lorsqu'il le juge nécessaire.

Enfin, le lecteur devra comprendre que l'algorithme a été développé dans un cadre de recherche universitaire et que les performances affichées sont satisfaisantes d'un point de vue académique. Dans sa version actuelle, le code n'est pas optimisé pour satisfaire à des besoins industriels et/ou commerciaux.

## 7.2 Enrichissement de modèles numériques d'élévation (TIN) par interpolation le long de l'axe médian.

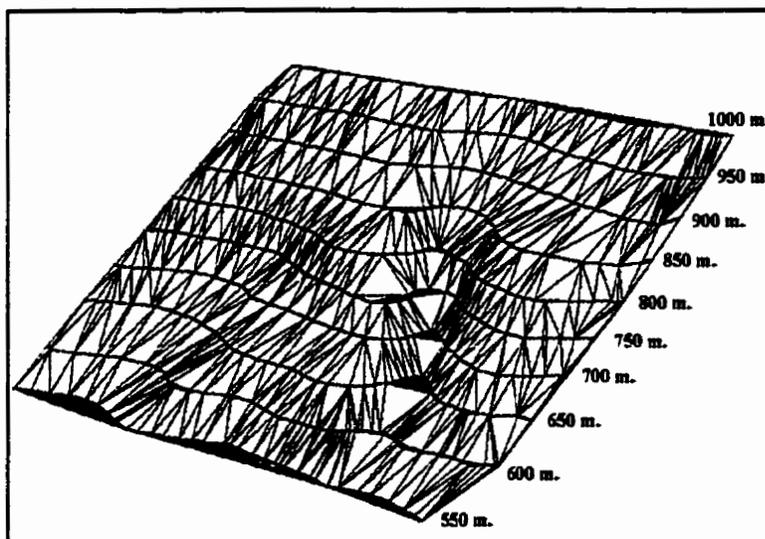
Figure 7.4



Extrait d'une carte topographique de la région du mont Albert (Québec, Canada).  
Source: Base nationale de données topographiques du Canada.

La technique d'enrichissement sera évaluée tout d'abord en analysant les résultats obtenus suite au traitement d'un fichier de courbes de niveau (figure 7.4) extrait d'une carte de la BNDT et présentant un secteur de la région du mont Albert (Québec, Canada). À noter que les courbes ont été généralisées au préalable avec l'algorithme présenté à la section 3 de ce mémoire. Le fichier comporte 469 vertex répartis sur 10 courbes de niveau à intervalles de 50 mètres. L'objectif général est d'évaluer la qualité de l'enrichissement en comparant la version originale du modèle à celle obtenue après enrichissement. Le lecteur remarquera que les triangles constituant les modèles présentés n'ont pas tous la même forme et la

Figure 7.5



Vue d'ensemble du modèle non enrichi et non interpolé.

même taille, ce qui peut parfois entraîner une mauvaise perception de la topographie.

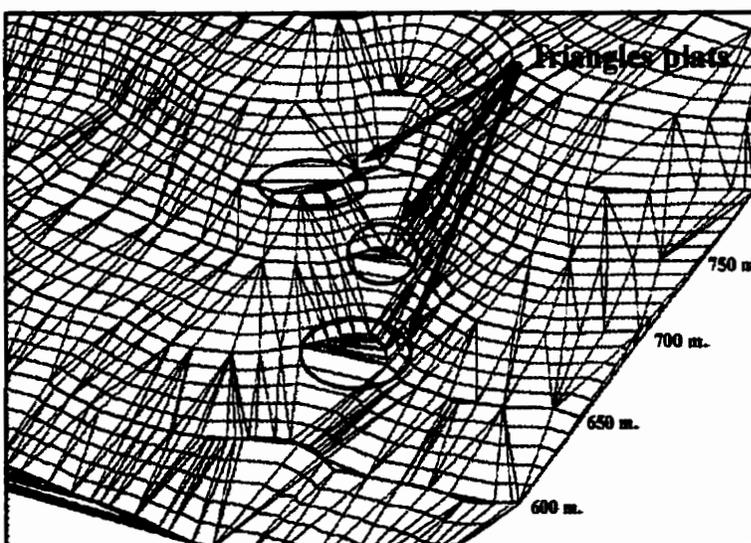
La figure 7.5 présente une vue d'ensemble du fichier. Les paramètres de visualisation pour les figures 7.5 à 7.8 inclusive-ment sont: rotation sur l'axe Z de 50 degrés vers l'est, angle vertical de 20 degrés et perspective de 100. À noter qu'aucun facteur d'exagération verticale

n'a été appliqué sur les modèles afin de respecter les proportions du terrain. Mentionnons aussi que les modèles n'ont pas été interpolés afin de mieux faire ressortir la présence des triangles plats.

L'outil utilisé pour visualiser les modèles numériques a été développé par l'auteur dans le cadre de cette recherche. Son fonctionnement et ses caractéristiques techniques sont présentés en détail à la section 8 de ce mémoire.

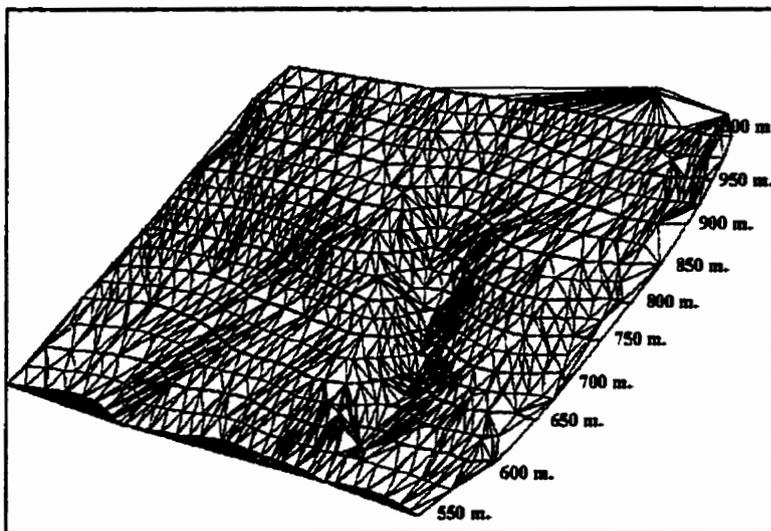
La figure 7.6 présente une vue rapprochée de la zone où la formation de triangles plats peut être observée. On voit clairement les plateaux formés entre les courbes de 650 et 750 mètres. Comme nous l'avons mentionné à la section 6.2, l'ajout des vertex de l'axe médian dans la triangulation permettra de solutionner ce problème.

Figure 7.6



Vue rapprochée de la figure 7.5 avec identification des zones de triangles plats et courbes de niveau intermédiaire à équidistance de 10 mètres.

Figure 7.7



Vue d'ensemble du modèle enrichi mais non interpolé.

Après avoir interpolé de nouvelles valeurs d'élévation sur les différents segments de l'axe médian, 695 vertex ont été insérés dans le fichier portant le nombre total de vertex à 1121.

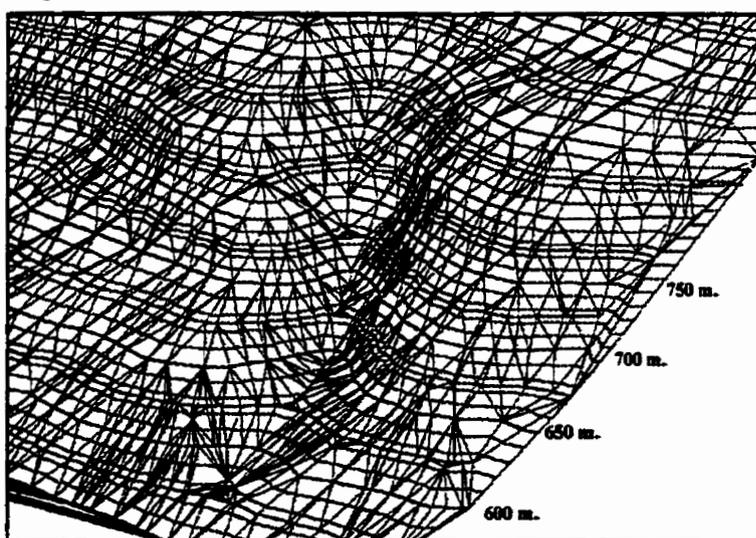
La figure 7.7 présente la version enrichie du modèle, comme en témoigne sa densité accrue. La figure 7.8 montre une vue rapprochée de la zone où se trouvaient auparavant les triangles

plats. On peut constater qu'ils ont été éliminés. L'axe médian des courbes de niveau a été superposé au modèle et il est représenté en violet. On observe que les branches de l'axe médian suivent l'orientation de cette vallée localisée au centre du modèle.

Également, la reconstitution des courbes à partir du modèle enrichi montre que celles-ci passent aux mêmes endroits que sur le fichier original avant l'enrichissement. En fait, ce ne sont que les portions du terrain situées en zone de triangles plats qui ont réellement été modifiées par l'insertion des vertex. Les valeurs

obtenues sur les branches de l'axe médian sont-elles exactes et reflètent-elles la réalité terrain? Difficile de l'affirmer. Rappelons que l'interpolation par ratio de cercles, qui a été développée pour cette application, est une interpolation de type linéaire. En effet, la progression des valeurs d'élévation est constante sur toute la longueur d'une branche. Après

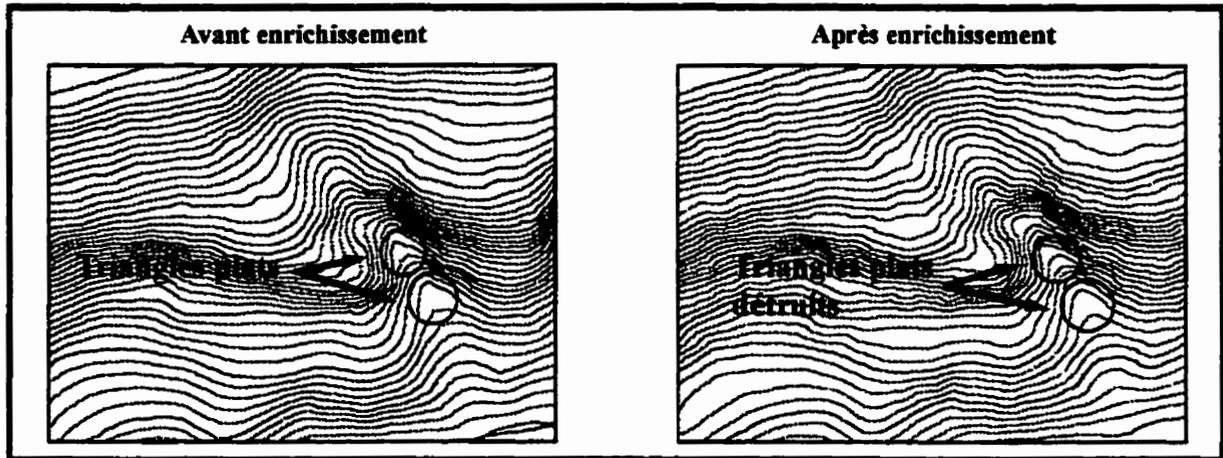
Figure 7.8



Vue rapprochée de la figure 7.7 avec superposition de l'axe médian et courbes de niveau intermédiaire à équidistance de 10 mètres.

insertion des vertex des branches dans la triangulation, les zones de dépression et les crêtes auront une pente constante dans l'axe de la branche. Cependant, nous considérons qu'en l'absence d'information topographique supplémentaire, tels des points cotés, les valeurs d'éléva-

**Figure 7.9**



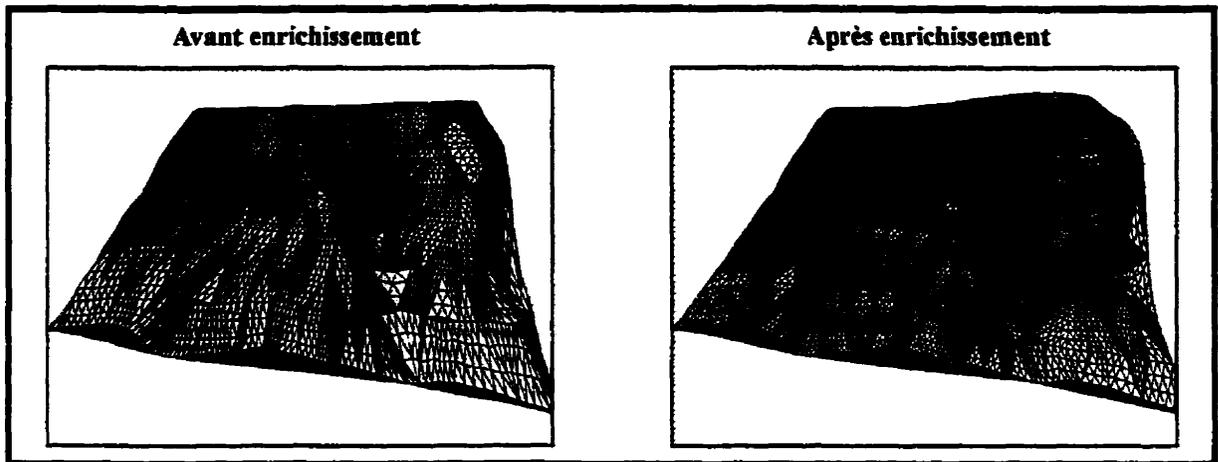
Distribution des courbes de niveau intermédiaires avant et après l'enrichissement.

tion le long des branches se conforment certainement plus à la réalité terrain que les triangles plats d'origine, ce qui confère au modèle numérique une nette amélioration au plan de sa qualité.

La figure 7.9 montre les courbes maîtresses et intermédiaires extraites avant et après enrichissement du modèle. Les courbes sont présentées depuis un angle de vue vertical (nadir) et après une légère rotation de 30 degrés vers l'ouest. Sur l'image de gauche, les zones de triangles plats sont bien visibles alors que sur l'image de droite, la pénétration des courbes de niveau intermédiaire dans la zone de triangles plats confirme que ces derniers sont disparus. La figure 7.10 montre le modèle interpolé auquel un effet d'ombrage a été appliqué. Un facteur d'interpolation de 3, un angle de vue vertical de 30 degrés, une rotation de 20 degrés vers l'est et un indice de perspective de 15 ont été choisis pour produire les deux versions de ce modèle.

Afin de bien démontrer l'apport et la validité d'une telle méthode d'enrichissement pour la confection de modèles numériques, nous allons maintenant comparer nos résultats à ceux obtenus avec un logiciel commercial d'interpolation nommé *Surfer* (produit par *Golden Software Inc.*) et ce, pour un cas extrême de triangles plats. Ce cas consiste en trois courbes de niveau très

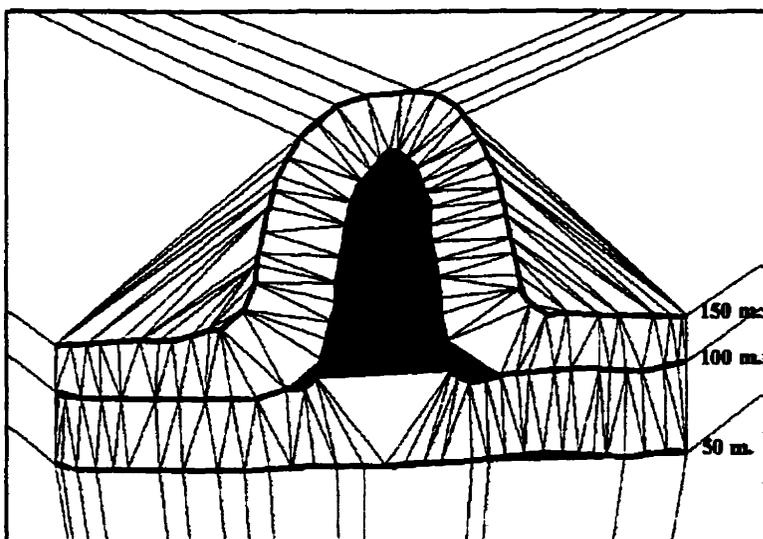
Figure 7.10



Modèles interpolés présentés avec effet d'ombrage avant et après l'enrichissement.

fortement infléchies, au centre desquels tous les triangles sont plats (figure 7.11). Nous verrons ainsi comment se comportent les algorithmes d'interpolation lorsque confrontés à une pareille situation. Deux modèles ont été conçus dans *Surfer* à partir des courbes de niveau original (fichier non enrichi). Le premier modèle a été produit avec un algorithme de krigeage et le second avec un algorithme de distance inverse à la puissance 2. Ces modèles sont les deux meilleurs que *Surfer* a pu produire en employant deux algorithmes différents d'interpolation. Enfin, nous présenterons le modèle produit à partir des courbes de niveau original auquel nous

Figure 7.11

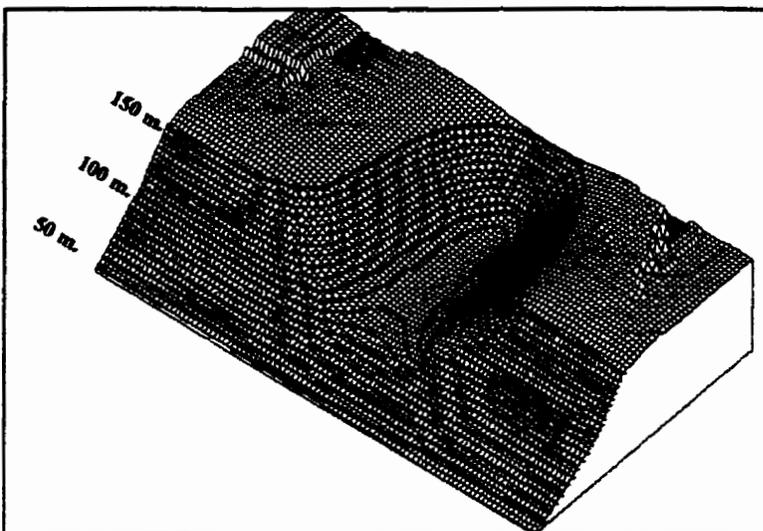


Cas extrême de triangles plats.

avons ajouté les valeurs interpolées sur l'axe médian. La confection de ce troisième modèle a été réalisée dans notre application maison (*Voronoi Toolkit*) à l'aide du module de visualisation 3D. L'interpolation par "aire volée" fut utilisée. Les paramètres pour l'interpolation des différents modèles sont présentés respectivement au bas de chaque figure.

L'interpolation par krigeage (figure 7.12) semble, à première vue, avoir donné de bons résultats. En observant plus attentivement, on se rend cependant compte que l'algorithme n'a pu

Figure 7.12



Modèle produit dans *Surfer* avec interpolation par krigeage à partir d'un fichier non enrichi. Paramètres d'interpolation par défaut choisis, à l'exception des suivants: type de recherche: octant; ellipse de recherche: 20-20; points par secteur: 6; nombre minimum de points: 10; minimum de secteur vide: 4.

d'élévation dans la zone de triangles plats. De plus, même les courbes de niveau original (en rouge) n'ont pu être adéquatement reconstituées.

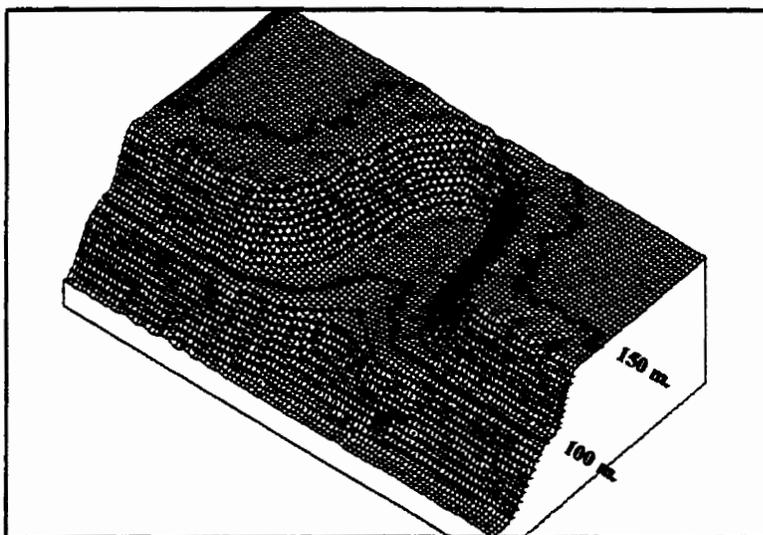
La figure 7.14 montre le modèle obtenu une fois que les valeurs interpolées sur l'axe médian ont été ajoutées. Comparativement à ce qui a été présentée à la figure 7.12, la vallée obtenue est plus profonde, comme en témoigne le plus grand nombre de courbes intermédiaires dans cette zone (4 courbes versus 2 sur le modèle de la figure 7.12).

Nous avons présenté une évaluation en regard des modèles

interpoler correctement le fond de cette vallée car une zone plane subsiste en son centre. Cette vallée devrait être un peu plus profonde si on se base sur le dénivelé existant entre la première (50 m.) et la deuxième courbe (100 m.)

Pour ce qui est du second modèle produit par *Surfer* (figure 7.13), les résultats obtenus sont de piètre qualité. L'interpolation par distance inverse n'a pu estimer correctement des valeurs

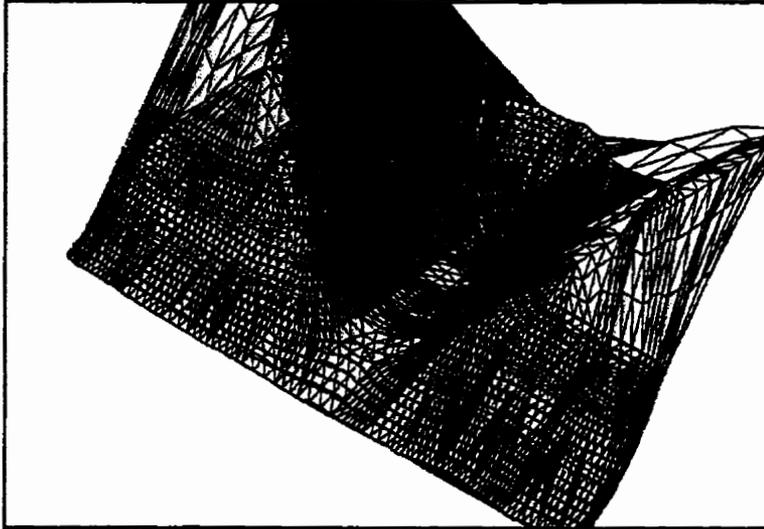
Figure 7.13



Modèle produit dans *Surfer* avec interpolation par distance inverse à la puissance 2 à partir d'un fichier non enrichi. Paramètres d'interpolation par défaut choisis, à l'exception des suivants: type de recherche: octant; ellipse de recherche: 30-30; points par secteur: 6; nombre minimum de points: 10; minimum de secteur vide: 4.

numériques d'élévation produits par la technique d'enrichissement. Mais qu'en est-il des performances de l'algorithme lui-même? Quel est le temps requis pour interpoler un nombre déterminé de vertex sur le tronc et les branches de l'axe médian? Une évaluation de l'algorithme est maintenant présentée en fonction du temps requis pour l'interpolation.

**Figure 7.14**



Modèle produit après enrichissement et avec interpolation "aire volée". Paramètres d'interpolation: facteur de subdivision des triangles de 3; aucun lissage appliqué.

Le micro-ordinateur utilisé est encore une fois un Pentium 200 MMX muni de 64 méga-octets de mémoire vive et d'un espace disque de 6.4 giga-octets. Le composant *Tchrono* a permis la mesure des temps de calcul.

Pour le fichier du mont Albert comprenant 469 vertex (figure 7.4), le temps obtenu est de 170

millisecondes pour l'interpolation des 695 vertex de l'axe médian. Ce résultat démontre la rapidité relative de l'algorithme. En effet, comme l'algorithme n'a pas été testé sur un jeu de données de grande dimension, comme l'ensemble des courbes de niveau d'une carte topographique, par exemple, on ne peut parler ici de réelle performance. De part la structure du programme *Voronoi Toolkit* à l'intérieur duquel l'algorithme a été implanté, et en raison des contraintes de mémoire, liées notamment à l'emploi de la structure Quad-Edge, il s'avérait impossible de traiter un jeu de données de grande taille.

## **8. LE MODULE DE VISUALISATION 3D**

### **8.1 Objectifs**

Parfois, il peut s'avérer nécessaire de représenter le relief d'une région donnée en utilisant un mode de représentation offrant d'abord un aperçu plus naturel du terrain, mais également une plus grande liberté de visualisation, c'est-à-dire de permettre l'observation d'une région selon différents angles et hauteurs de vue. Contrairement à la représentation du relief à l'aide de courbes de niveau, la modélisation du terrain en 3D (en réalité 2D<sup>1/2</sup>) permet d'observer le relief un peu comme si on y était. Ce type de visualisation permet d'embrasser d'un seul coup d'œil l'aspect général des lieux.

Dans le cadre de cette recherche, un module de visualisation 3D a été développé dans l'optique de valider la qualité des modèles produits à l'aide de la méthode d'enrichissement. Les fonctions offertes par ce module sont les suivantes: le déplacement de l'observateur autour des axes de rotation XYZ, la variation de la perspective (distance de l'observateur par rapport au modèle), l'interpolation "aire volée", l'exagération verticale afin de faire ressortir les faibles variations du relief et, enfin, l'illumination virtuelle du modèle, ce qui permet d'accentuer l'effet 3D et de mettre en valeur la topographie du terrain.

### **8.2 Développement**

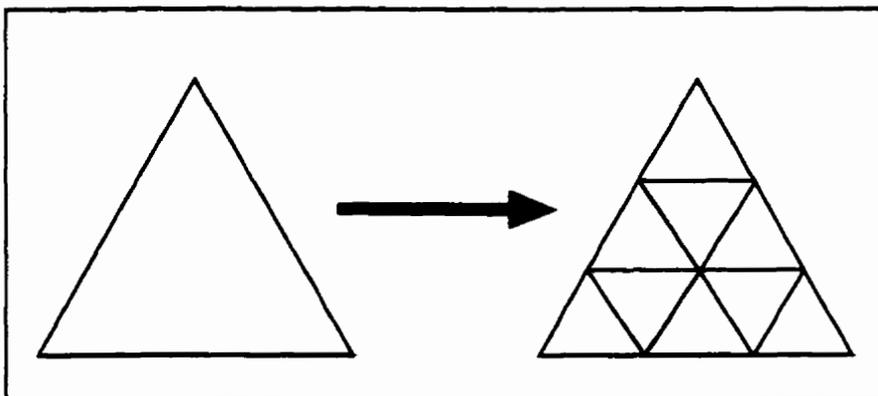
#### **8.2.1 Présentation**

Le module de visualisation a été développé à l'intérieur de l'application *Voronoi Toolkit*. Rappelons que cette application a été conçue en partie par l'auteur et par le Dr. Christopher Gold, professeur au département des Sciences Géomatiques, de même que par M. Mir Abofalz Mostafavi, étudiant au doctorat. Le développement du module a été réalisé en langage de programmation Delphi. Au cours des prochaines sous-sections, nous détaillerons son fonctionnement et décrirons les divers algorithmes qu'il utilise.

### 8.2.2 Densification et interpolation du TIN

L'interpolation est une étape cruciale dans la modélisation de terrain comme nous l'avons mentionné abondamment à la section 5. Rappelons qu'elle consiste à estimer de nouvelles valeurs entre les points de données afin d'enrichir le modèle. À l'intérieur du module, l'interpolation "aire volée" est utilisée pour l'estimation des nouvelles valeurs.

**Figure 8.1**



Densification du TIN par subdivision des triangles. Est présentée ici, une subdivision de facteur 3 qui engendre 9 triangles enfants ( $3^2$ ).

La densification du TIN est réalisée en subdivisant les triangles parents, ceux passant aux points de données, en un nombre X de triangles de type enfants dont les valeurs d'élévation aux sommets seront interpolées (fi-

gure 8.1). La technique de subdivision des triangles est décrite en détail dans Gold (1987). Le module de visualisation permet la subdivision des triangles d'un facteur de trois (3) jusqu'à trente (30). Un facteur de 3 divisera le triangle parent en 9 triangles enfants ( $3^2$ ), un facteur de 10 engendrera 100 nouveaux triangles ( $10^2$ ) et ainsi de suite. Bien que le module offre un facteur de subdivision aussi élevé que 30, un facteur de 15 suffit largement pour obtenir un réseau de triangles de bonne densité, à condition qu'au départ la densité des points de données soit suffisante.

### 8.2.3 Projection du modèle sur le plan

Pour être visualisées sur l'écran d'un ordinateur, les coordonnées 3D des triangles doivent être projetées, d'où l'appellation de représentation en 2D  $\frac{1}{2}$ . En effet, il ne peut s'agir de 3D car le médium de visualisation (l'écran) ne supporte pas la troisième dimension, il ne peut que la simuler.

**Figure 8.2**

1	0	0	0
0	1	0	0
0	0	1	0
0	0	-H	1

Matrice de translation.  
Source: Gold, 1998b

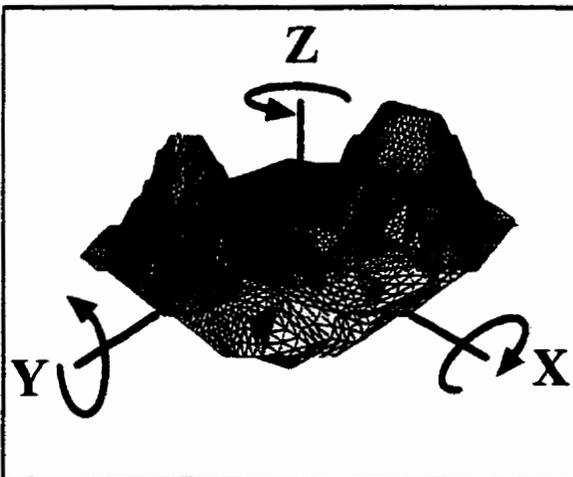
La projection des coordonnées se fait à l'aide de matrices de dimension 4X4, couramment utilisées en infographie. L'avantage d'employer les matrices est qu'on peut les combiner par multiplication afin d'obtenir une seule matrice de transformation que l'on appliquera ensuite aux coordonnées (Gold, 1998b). La projection de coordonnées sur le plan nécessite deux matrices: une matrice de translation et une matrice de perspective. La matrice de translation permet de déplacer le point d'observation du modèle à l'extérieur du plan à une distance  $D+1$  ou  $H$  de l'écran,  $D$  étant la distance entre l'écran et une coordonnée dans l'espace 3D. Cette translation doit être effectuée avant d'appliquer la perspective. La matrice de translation est présentée à la figure 8.2. La matrice de perspective, quant à elle, projette chacune des coordonnées sur le plan. Une fois projetées, les coordonnées ont tous un  $Z$  égale à  $D$ , soit la surface de l'écran. La figure 8.3 illustre la matrice de perspective. La valeur de  $H$  correspond à la distance entre l'observateur et le centre du modèle. En soustrayant 1 à  $H$ , on obtient la distance entre l'observateur et le plan sur lequel est projeté le modèle, en l'occurrence l'écran de l'ordinateur.

La projection des coordonnées se fait à l'aide de matrices de dimension 4X4, couramment utilisées en infographie. L'avantage d'employer les matrices est qu'on peut les combiner par multiplication afin d'obtenir une seule matrice de transformation que l'on appliquera ensuite aux coordonnées (Gold, 1998b). La projection de coordonnées sur le plan nécessite deux matrices: une matrice de translation et une matrice de perspective. La matrice de translation permet de déplacer le point d'observation du modèle à l'extérieur du plan à une distance  $D+1$  ou  $H$  de l'écran,  $D$  étant la distance entre l'écran et une coordonnée dans l'espace 3D. Cette translation doit être effectuée avant d'appliquer la perspective. La matrice de translation est présentée à la figure 8.2. La matrice de perspective, quant à elle, projette chacune des coordonnées sur le plan. Une fois projetées, les coordonnées ont tous un  $Z$  égale à  $D$ , soit la surface de l'écran. La figure 8.3 illustre la matrice de perspective. La valeur de  $H$  correspond à la distance entre l'observateur et le centre du modèle. En soustrayant 1 à  $H$ , on obtient la distance entre l'observateur et le plan sur lequel est projeté le modèle, en l'occurrence l'écran de l'ordinateur.

**Figure 8.3**

1	0	0	0
0	1	0	0
0	0	1	$1/H-1$
0	0	0	1

Matrice de perspective.  
Source: Gold, 1998b

**Figure 8.4**

Axes de rotation du modèle numérique d'élevation.

#### 8.2.4 Déplacement de l'observateur autour du modèle

Afin de visualiser le TIN de divers points de vue, il est nécessaire de permettre le déplacement de l'observateur autour du modèle. Ce déplacement s'effectue autour des axes de rotation XYZ tels qu'illustrés à la figure 8.4. À l'intérieur du module de visualisation, les mouvements sur les 3 axes sont permis, bien que seuls

les déplacements autour des axes Y et Z soient vraiment utiles. L'axe Y permet de faire varier la hauteur de l'observateur par rapport à l'horizon, alors que l'axe Z permet la rotation autour du modèle. La rotation autour de l'axe X est peu ou pas utilisée car elle a comme conséquence d'incliner la ligne d'horizon.

**Figure 8.5**

Matrice X				Matrice Y				Matrice Z			
1	0	0	0	$\text{Cos}(\theta)$	0	$-\text{Sin}(\theta)$	0	$\text{Cos}(\theta)$	$\text{Sin}(\theta)$	0	0
0	$\text{Cos}(\theta)$	$\text{Sin}(\theta)$	0	0	1	0	0	$-\text{Sin}(\theta)$	$\text{Cos}(\theta)$	0	0
0	$-\text{Sin}(\theta)$	$\text{Cos}(\theta)$	0	$\text{Sin}(\theta)$	0	$\text{Cos}(\theta)$	0	0	0	1	0
0	0	0	1	0	0	0	1	0	0	0	1

Les matrices de rotation X, Y et Z.  
Source: Gold, 1998b

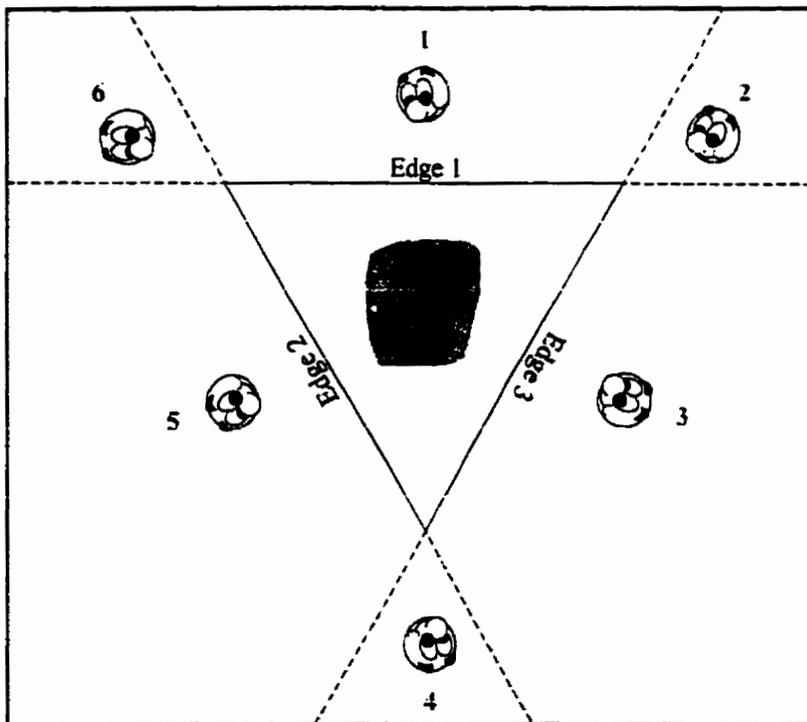
Les déplacements autour des axes sont réalisés à l'aide d'une série de 3 matrices qui, lorsqu'elles sont multipliées ensemble, fournissent la matrice complète de rotation. C'est cette matrice résultante qui est appliquée aux coordonnées. La figure 8.5 illustre chacune des 3 matrices de rotation.

### 8.2.5 Séquence d'envoi des triangles parents et ordre d'affichage des triangles enfants

La séquence d'envoi des triangles parents et l'ordre d'affichage des triangles enfants sont deux opérations distinctes. La première est appliquée lors de l'envoi des triangles pour subdivision et interpolation. La deuxième est l'ordre d'affichage d'un ensemble de triangles (enfants) pour un triangle (parent) donné. La section 8.3 présente ces deux opérations dans le flux procédural du module de visualisation.

En fonction des déplacements de l'observateur autour du modèle, la séquence d'envoi des triangles à l'écran se doit d'être ajustée et pour cause. En fait, les triangles qui sont les plus près de l'observateur seront les plus éloignés après rotation de 180 degrés autour du modèle. Ceci nécessite que le maillage de triangles soit dessiné selon un ordre fixé d'après la position de l'observateur. L'algorithme utilisé pour parcourir le maillage est encore une fois *Scan*. Tout d'abord, on spécifie un point de vue situé au nord de l'observateur et à l'extérieur du maillage. Ensuite, on précise le segment (d'un triangle) de départ; celui-ci doit être le plus éloigné de l'observateur. On traite ensuite les autres triangles en progressant vers l'avant, en direction de l'observateur. Cet algorithme a la particularité de ne traiter qu'une seule fois un triangle, ce qui nous assure qu'aucun ne sera dédoublé.

Figure 8.6

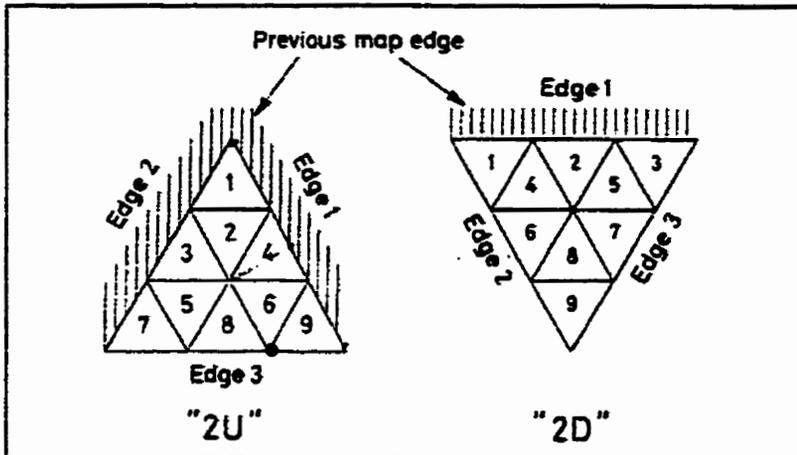


Les six séquences de parcours du maillage de triangles en fonction de la position de l'observateur.

Six séquences de parcours ont été définies afin d'assurer un affichage parfait du modèle. La figure 8.6 présente ces six (6) séquences. Chacune d'elle définit également l'ordre avec lequel les triangles seront envoyés pour subdivision, interpolation et affichage. Si l'observateur se situe en position 1, on parcourt le maillage avec *Scan* en débutant par l'edge 3, puis on relance *Scan*, mais cette fois à partir de l'edge 2. Chacun

des deux appels à *Scan* traitera la moitié du maillage. Lorsque l'observateur est en position 2, on appelle *Scan* une fois à partir de l'edge 2. En position 3, on lance *Scan* depuis l'edge 2 et ensuite l'edge 1. En position 4, on lance *Scan* une fois depuis l'edge 1. En position 5, on appelle *Scan* une fois de l'edge 1 et une fois de l'edge 3. Enfin, si l'observateur est à la sixième position, on parcourt une fois le maillage depuis l'edge 3.

Figure 8.7



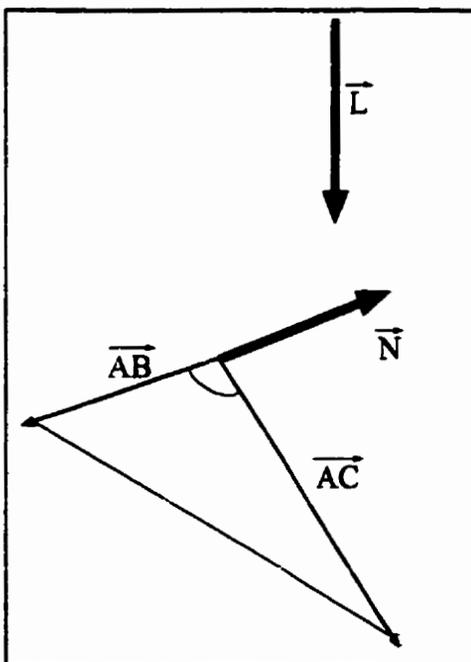
Ordre d'affichage des triangles enfants en fonction du type de triangle parent soit "2U" et "2D".

Source: Gold, 1987

"2U" (Two Up). Le second possède, à l'inverse, deux côtés orientés vers le bas (edges 2 et 3), par conséquent, on le nomme "2D" (Two Down). Cette technique assure que, à l'intérieur d'un triangle parent, aucun triangle enfant n'en masquera un autre. Ceci est particulièrement utile lorsque l'inclinaison des triangles enfants varie quelque peu.

Pour chacun des triangles traités par *Scan*, on doit également définir un ordre pour l'affichage des triangles enfants. Cet ordre dépend du type de triangle parent. Gold (1987) a développé la technique et a défini les deux types de triangle (figure 8.7). Le premier possède deux côtés orientés vers le haut (edges 1 et 2), on le nomme

Figure 8.8



L'illumination d'un triangle se fait en comparant le vecteur normal du triangle nommé  $\vec{N}$  au vecteur de la source lumineuse nommé  $\vec{L}$  dont la position est définie par l'utilisateur.

### 8.2.6 Illumination du modèle.

L'illumination d'un modèle numérique d'élévation par une source de lumière virtuelle est une opération qui vise à accentuer l'effet de la troisième dimension en faisant ressortir davantage les variations de pente du terrain. Ainsi, les surfaces planes, les dépressions et les zones surélevées sont mises en évidence. L'illumination d'un triangle se fait en comparant le vecteur normal ( $\vec{N}$ ) du triangle (vecteur perpendiculaire à la face du triangle) au vecteur de la source lumineuse ( $\vec{L}$ ) dont la position est définie par l'utilisateur (figure 8.8). Voici la liste des opérations requises pour calculer un coefficient d'illumination pour un triangle. À noter qu'on utilise

les coordonnées des triangles transformés par les matrices de rotation et de translation pour le calcul de ce coefficient.

- 1) À partir de deux vecteurs (AB et AC), qui composent deux (2) côtés d'un triangle, on calcule le produit vectoriel (vecteur normal) . Le vecteur ainsi obtenu est orthogonal à la surface du triangle.
- 2) On calcule ensuite le produit scalaire du vecteur normal par le vecteur de la source lumineuse qui varie en fonction de sa position, définie par l'utilisateur.
- 3) On divise ensuite le résultat du produit scalaire par le produit de la norme du vecteur de la source lumineuse avec la norme du vecteur calculé au point 2.
- 4) Ceci nous donne un coefficient variant entre  $-1$  et  $1$ . Lorsqu'il est de  $1$ , le triangle est éclairé directement par la source lumineuse, c'est-à-dire que les rayons lumineux sont perpendiculaires à sa surface. Lorsqu'il est de  $-1$ , la surface du triangle est dans l'obscurité totale.
- 5) Pour donner l'effet d'éclairage au TIN, il suffit d'assigner un dégradé de couleur (du blanc au noir par exemple) aux valeurs de coefficients obtenues.

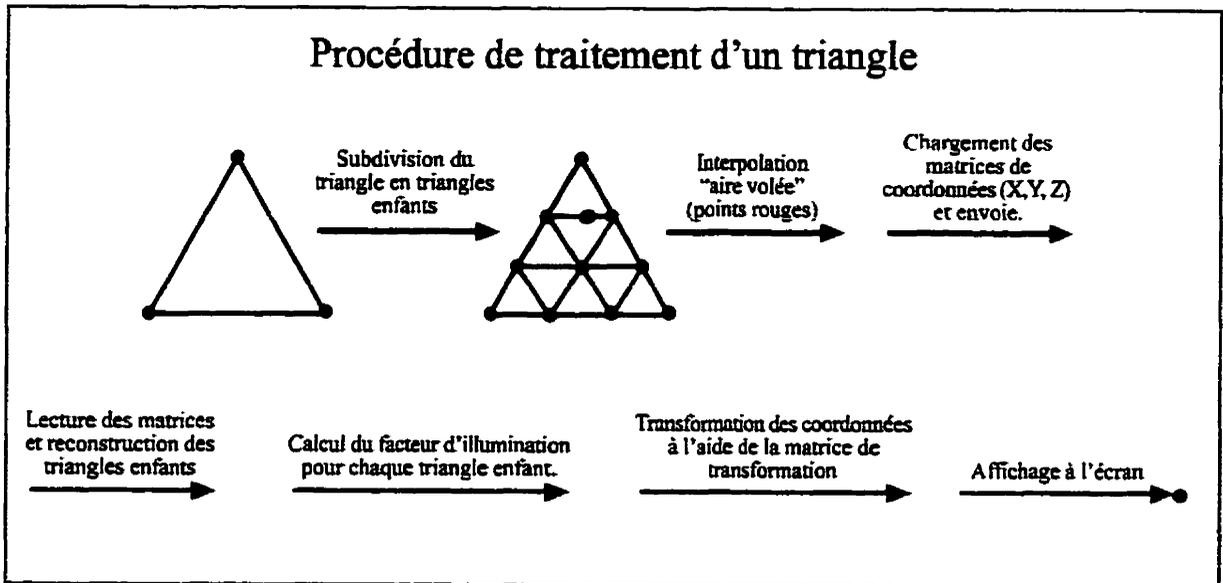
### 8.3 Flux des opérations à l'intérieur du module de visualisation 3D

Le flux des opérations montre la séquence d'appel des procédures et fonctions à l'intérieur du code. Il s'agit de l'ordre avec laquelle les triangles du TIN sont traités jusqu'à leur affichage à l'écran avec les effets de perspective, de rotation et d'illumination. La figure 8.9 schématise le flux des opérations.

La première opération consiste d'abord à produire la matrice de transformation et la matrice de perspective. La matrice de transformation, qui résulte du produit de la matrice de rotation (la matrice de rotation est le produit des matrices de rotation X, Y et Z) avec la matrice de translation, sera appliquée à chaque coordonnée pour le calcul du coefficient d'illumination des triangles. La perspective sera appliquée par après car on ne peut calculer un coefficient d'illumination qu'une fois que les coordonnées ont été projetées.

La seconde étape consiste à parcourir le maillage de triangles une seule fois afin de calculer un

Figure 8.9



Ordre séquentiel des opérations réalisées pour le traitement d'un triangle donné à l'intérieur du module de visualisation 3D

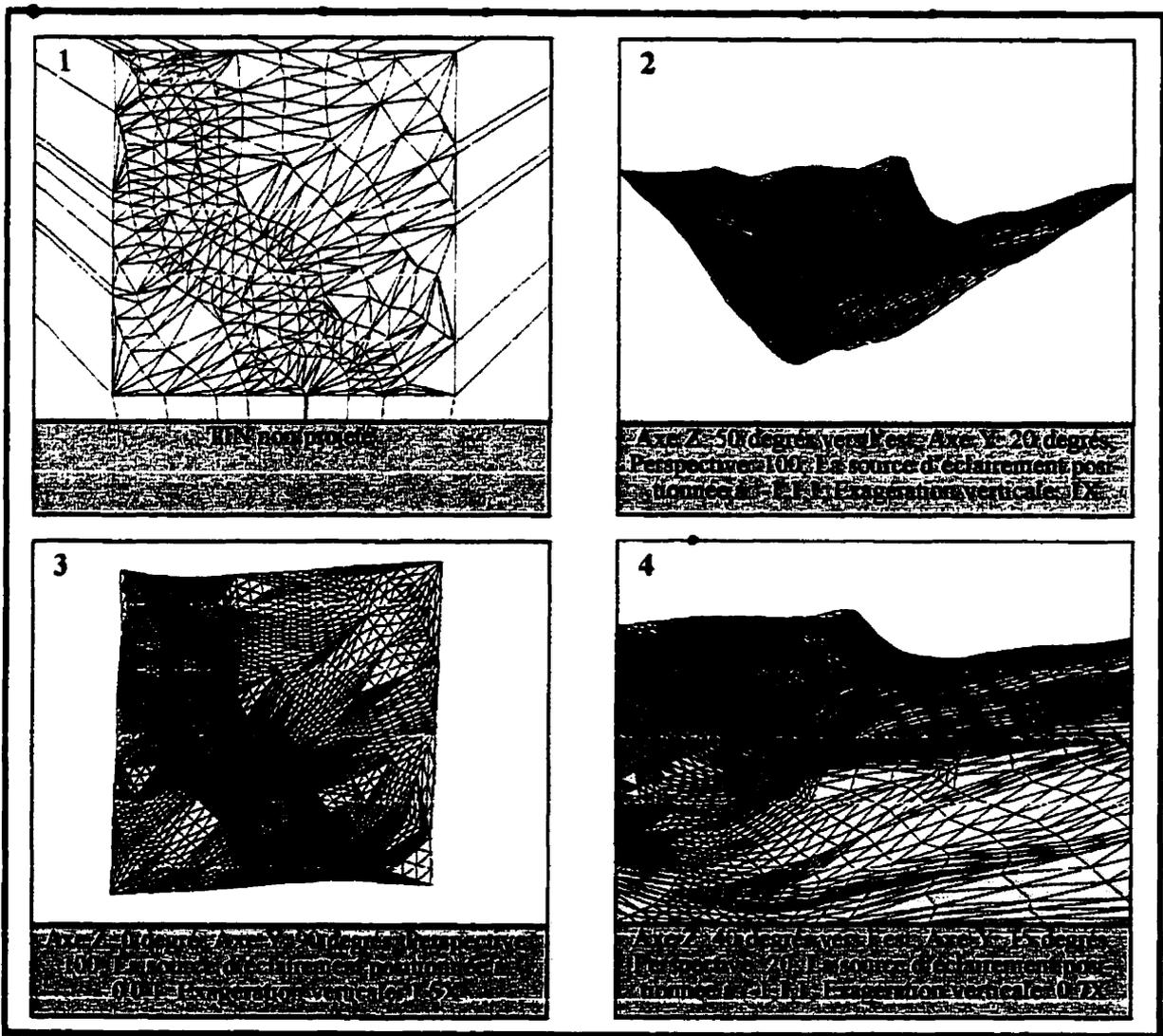
DeltaX (pente horizontale) et un DeltaY (pente verticale) pour chacun des points de données. Ceci sera nécessaire lors de l'interpolation par "aire volée".

À la troisième étape, on parcourt à nouveau le maillage de triangles, mais cette fois pour débiter la division des triangles en triangles enfants. Au cours de cette opération, un seul triangle est traité à la fois et divisé en un nombre de triangles enfants déterminé par la valeur du facteur de subdivision. Chacun des sommets des triangles enfants est envoyé à une fonction d'interpolation qui calculera une valeur d'élévation. Une fois la valeur Z obtenue pour un sommet, sa coordonnée est placée dans une série de trois (3) matrices: une pour les coordonnées X, une autre pour les coordonnées Y et une pour les coordonnées Z. Lorsque tout le triangle a été subdivisé et que les valeurs d'élévation des sommets des triangles enfants ont été calculées, les trois (3) matrices sont alors remplies. Elles seront reçues par une fonction qui va lire leur contenu et reconstruire les triangles enfants dans le bon ordre. Lorsqu'un triangle enfant est construit, on applique aux coordonnées de ses sommets la matrice de transformation pour calculer la valeur du gradient d'illumination et déterminer sa couleur de remplissage. Par après, on applique aux coordonnées résultantes la matrice de perspective et on dessine le triangle à l'écran.

### 8.4 Évaluation du module de visualisation 3D

Dans le cadre de cette recherche, ce module de visualisation aura rendu possible l'évaluation de la technique d'enrichissement présentée à la section 7. En effet, il s'avérait nécessaire, pour démontrer le potentiel de cette technique, de se doter d'un outil permettant de visualiser un TIN sous divers angles et hauteurs de vue. La possibilité de faire varier le positionnement de la source d'éclairage autour du modèle aura permis de renforcer visuellement les variations de la topographie du terrain. En somme, ce module est un outil qui s'est avéré d'une grande utilité. Sans lui, une partie considérable de cette recherche n'aurait pu être menée à terme.

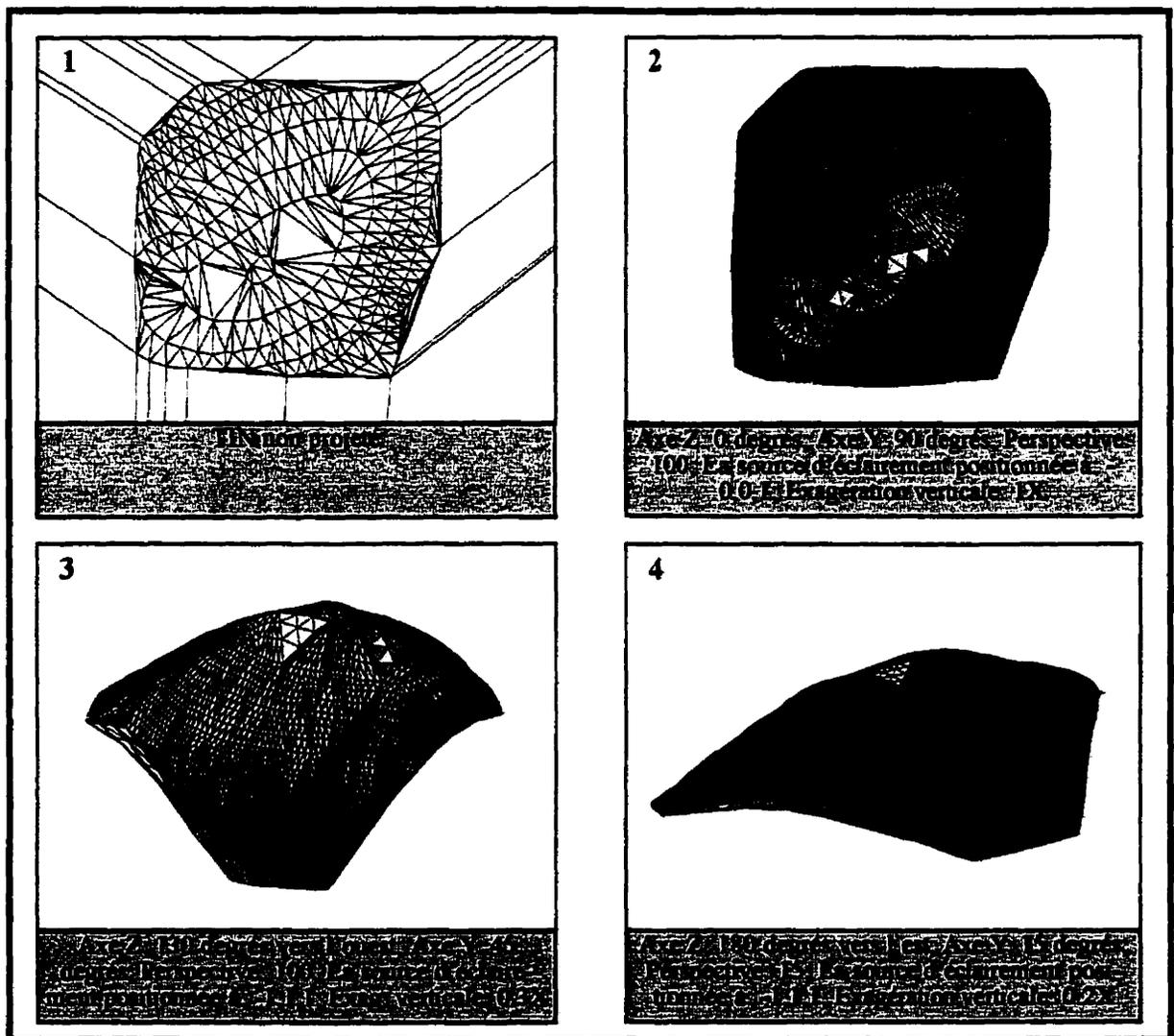
Figure 8.10



Visualisation d'un TIN représentant un fleuve encastré.

Passons maintenant à l'évaluation proprement dite du module. Plusieurs tests ont été réalisés avec divers jeux de données afin d'évaluer la qualité des modèles produits. Sauf indication contraire, tous les modèles ont été obtenus par interpolation "aire volée" avec lissage hermitien et un facteur de subdivision de 3. Le premier modèle présenté a été conçu à partir d'un fichier de points générés par l'auteur et représentant un fleuve encastré. La figure 8.10 présente le TIN avant interpolation ainsi que différentes représentations obtenues par rotation et inclinaison du modèle interpolé. Les paramètres de visualisation pour les images 2,3,4 sont présentés au bas de chacune d'elles. À noter qu'aucune rotation n'a été appliquée sur l'axe X. On distingue clairement entre les images 2 et 4, l'effet provoqué par une variation de la perspective; l'image

Figure 8.11



Visualisation d'un TIN représentant une montagne de la région du Lac Killarney au Missouri.  
Source de la carte: Miller et al., 1989

étant plus étirée lorsque la perspective diminue, l'observateur étant alors plus près du modèle. Mentionnons aussi que l'exagération a été abaissée sur la figure 4, ce qui se traduit par un contraste moins accentué entre les hautes et les basses élévations. Sur la figure 3, on a fait ressortir la pente du terrain en positionnant la source d'éclairage au nadir, à laquelle on a associé une exagération verticale plus élevée. Ainsi, plus la couleur d'un triangle est foncée, plus l'inclinaison de sa face est élevée.

Un deuxième jeu de données, constitué de courbes de niveau provenant d'une carte topographique de la région du Lac Killarney au Missouri (tirée de Miller et al. 1989) et représentant le sommet d'une montagne, est illustré à la figure 8.11. L'image 2 présente le modèle vu du nadir, avec source d'éclairage située également au nadir. Ceci permet, comme à l'image 3 de la figure 8.10, de mettre en évidence la pente du terrain. Comme le modèle n'a pas été enrichi, on peut remarquer que le sommet de la montagne est plat du fait que plusieurs triangles sont blancs. Les courbes originales ont été superposées au modèle afin de montrer la bonne correspondance entre le TIN original (image 1) et le modèle interpolé. Sur l'image 3, aucun lissage hermitien n'a été appliqué. Enfin, l'image 4 a été produite à partir d'une interpolation avec facteur de subdivision de 6, ce qui s'est traduit par un modèle dont la résolution est quatre fois supérieure aux autres modèles présentés.

Comme l'ont démontré les figures 8.10 et 8.11, les modèles semblent de bonne qualité, qu'ils aient été produits à partir de courbes de niveau ou d'un semis de points altimétriques. Dans le cas du TIN produit à partir de courbes de niveau, rappelons que la qualité des résultats auraient pu être supérieure s'il avait été enrichi au préalable.

À long terme et si les besoins le justifient, il serait judicieux d'apporter des modifications à ce module afin d'accroître, d'abord, l'étendue des fonctionnalités offertes et de corriger certains problèmes relatifs à l'aspect visuel des triangles après subdivision, mais aussi pour en optimiser le code, notamment en ce qui a trait à la gestion de la mémoire.

Il pourrait être envisagé de doter le module d'une fonction permettant la rotation du TIN en temps réel à l'aide de barres de défilement, plutôt que de recommencer la procédure en entier (interpolation et affichage) à chaque fois que l'on désire appliquer une rotation au modèle. Pour

ce qui est de la gestion de la mémoire, nous nous sommes rendus compte que lorsqu'un fichier de grande taille était traité, la mémoire vive devenait une denrée rare après seulement quelques minutes d'utilisation. Nous avons également remarqué que la subdivision des triangles engendre parfois de petits triangles visuellement inesthétiques. À titre d'exemple, un triangle possédant de longues arêtes et subdivisé par un facteur de 5 produira 25 petits triangles très allongés. À un niveau réduit d'affichage, il ne devient alors plus possible de distinguer l'intérieur des triangles, seule la couleur de leur contour est alors perceptible.

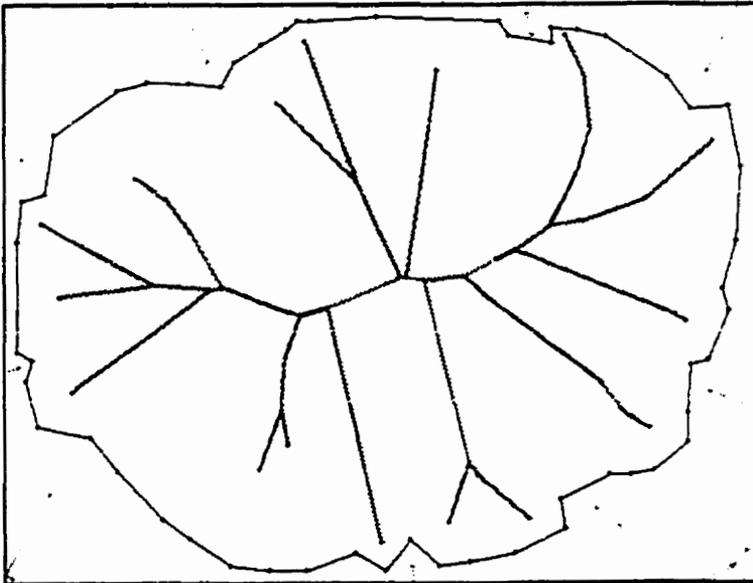
Mais en regard au mandat qui avait été donné à ce module, les résultats obtenus ont été jugés très satisfaisants dans l'ensemble.

## 9. TRAVAUX FUTURS

### 9.1 Généralisation des branches d'ordre 2

Lorsque l'algorithme de généralisation a été développé, l'objectif était de concevoir un outil permettant d'éliminer les perturbations mineures le long des courbes; une rétraction de premier ordre s'avérait alors suffisante. La version actuelle de l'algorithme de généralisation ne permet que la rétraction des branches de premier ordre de l'axe médian. On ne rétracte donc que les feuilles de l'axe médian vers leur nœud parent. Pour généraliser la croûte par rétraction des

Figure 9.1



Les divers ordres des branches de l'axe médian sont représentés par une couleur distinctive.

détruire ces feuilles et réassigner les nœuds parents comme étant maintenant les feuilles. Ensuite, on rétracte ces nouvelles feuilles vers leurs nœuds parents respectifs. Sur la figure 9.1, les divers ordres des branches de l'axe médian sont représentés par une couleur distinctive. Par exemple, les branches d'ordre 1 sont affichées en vert alors que celles d'ordre 2 sont affichées en bleu foncé.

La rétraction des branches d'ordre 2 généralisera davantage la croûte et réduira encore plus le nombre de points sur la courbe. En effet, pour chaque feuille qui sera détruite, trois vertex de la courbe seront retirés du même coup. Le même processus peut être répété pour une généralisation d'ordre 3, 4, etc.

L'ajout de la rétraction d'ordre 2 soulève toutefois certaines questions concernant le maintien de la topologie. En effet, plus on rétracte les branches, plus la forme de la croûte sera altérée. Que se passera-t-il, par exemple, si on applique une généralisation d'ordre 2 à un fichier de courbes de niveau? Est-ce que les lignes et les connexions entre celles-ci seront préservées? En

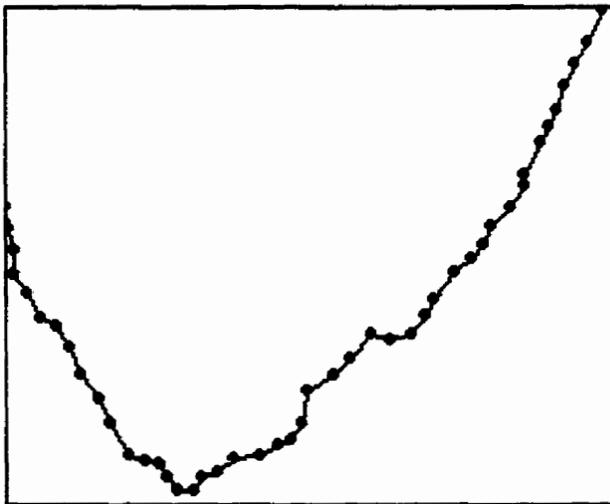
fait, il ne s'agit plus ici d'éliminer le bruit sur la courbe mais bien de modifier sa forme générale. Ces questions devront obtenir une réponse avant implantation car on se doit de prendre les mesures nécessaires pour préserver l'intégrité topologique et la morphologie des objets.

## 9.2 Problèmes d'échantillonnage sur les courbes

Pour cette recherche, à moins d'une indication contraire, les courbes ont été construites dans le programme *Voronoi Toolkit* à partir d'une image en format RAW où les lignes étaient représentées en blanc (pixel de valeur 1) sur fond noir (pixel de valeur 0). Un programme développé par madame Josée Fitzback, anciennement étudiante au département des Sciences Géomatiques, a permis d'extraire les coordonnées de tous les pixels blancs de l'image par filtrage et d'écrire ces valeurs dans un fichier texte. Ensuite, ce fichier était lu par notre application et les courbes étaient reconstruites sous forme vectorielle.

Lors de la construction des courbes, la proximité des points est dépendante, en bonne partie, de la tolérance fixée par la fonction nommée *Hit* qui s'assure qu'aucun point n'entrera en collision avec un autre. Plus la tolérance est faible, plus les points vont être rapprochés les uns des

**Figure 9.2**



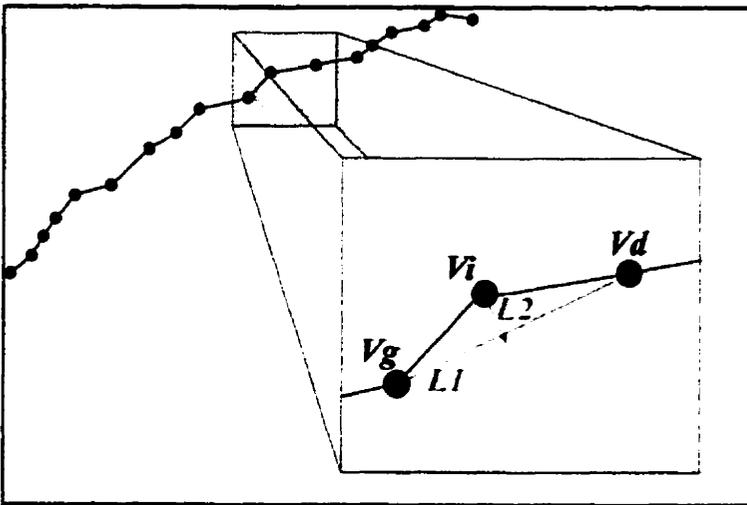
Exemple d'une courbe sur-échantillonnée pour laquelle on devrait réduire le nombre de vertex qui la constituent.

autres sur la courbe et inversement. Mais qu'advient-il si l'on possède déjà des courbes sur-échantillonnées et que l'on souhaite plutôt réduire le nombre de points en fonction de la complexité graphique de ces courbes? Une méthode simple permettant de réaliser cette réduction est présentée ici. Si elle s'avère efficace, elle pourrait être implantée éventuellement dans une version prochaine du *Voronoi Toolkit*.

Débutons par une courbe sur-échantillonnée (figure 9.2). L'algorithme utilisé consiste à évaluer si le retrait d'un vertex de la croûte

peut être fait sans que celle-ci ne se rupture. Si le retrait d'un vertex donné ne brise pas la croûte alors on le détruit. Chacun des vertex de la croûte, est testé successivement. Mais attention, cette méthode, si elle n'est pas correctement paramétrée, peut considérablement modifier la forme de la croûte. Ce que nous voulons, c'est réduire le nombre de points constituant la croûte, mais sans toutefois altérer sa forme. Certains vertex se doivent d'être préservés, comme ceux situés aux endroits de plus forte inflexion sur la courbe. Une tolérance appelée de "déviation" doit donc être ajoutée afin d'éliminer les points qui ne sont pas critiques au maintien de la forme de l'objet. On applique cette tolérance de la façon suivante.

**Figure 9.3**



Exemple d'une méthode pouvant être utilisée pour réduire le nombre de vertex sur une courbe.

Avant de détruire un vertex  $V_i$  sur la croûte, on calcule la longueur de la base du triangle formé par  $V_i$  avec ses deux voisins respectifs. Le vertex de droite à  $V_i$  est nommé  $V_d$  et le vertex de gauche à  $V_i$  est nommé  $V_g$  (figure 9.3). Nous appellerons cette première longueur  $L_1$ . On calcule ensuite la hauteur du triangle que l'on nommera  $L_2$ . Si  $L_2$  est plus grand que  $L_1$ , alors  $V_i$  est localisé à l'ex-

trémité d'une zone de forte inflexion et il se doit d'être préservé. Toutefois, si  $L_2$  est plus petit que  $L_1$  alors  $V_i$  peut potentiellement être retiré. On doit cependant s'assurer que son retrait n'entraînera pas une rupture de la croûte.

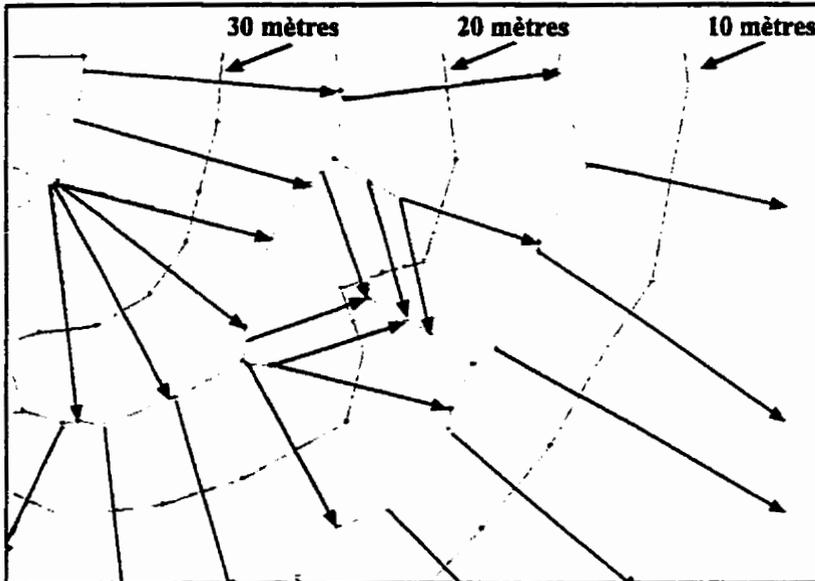
Nous tenons à aviser le lecteur que cette méthode est expérimentale et qu'elle n'a pas encore fait l'objet d'une quelconque implantation.

### 9.3 Modélisation du terrain

#### 9.3.1 Calcul de pente en travers des courbes de niveau

À partir d'une version triangulée des courbes de niveau, le diagramme de Voronoï associé per-

Figure 9.4



Les segments "anti-croûte" indiquent l'orientation de la pente du terrain et peuvent, par conséquent, servir à la confection d'un modèle d'écoulement.

met de conserver au sein de sa structure une définition explicite des pentes entre chacune des courbes. En fait, les segments "anti-croûte", soit les segments Voronoï croisant la croûte, peuvent être utilisés pour estimer un gradient de pente entre chacune des courbes (figure 9.4).

Comment définir une équation de pente pour chacun

des segments "anti-croûte"? Une fois que des valeurs d'élévation auront été interpolées aux vertex de l'axe médian par la méthode d'enrichissement, il sera ensuite possible de calculer un Delta Y et définir une équation de pente de la forme  $Ax+By+C$  pour chacun de ces segments en prenant les valeurs d'élévation situées à leurs extrémités. Ceci fournira la pente verticale du terrain à cet endroit précis. De plus, comme les segments de croûte sont perpendiculaires aux segments "anti-croûte", les équations de pente peuvent être attribuées aux vertex de la croûte. Ces équations pourront alors servir aux algorithmes d'interpolation pour l'estimation de nouvelles valeurs.

#### 9.3.2 Modèle d'écoulement

Comme les segments "anti-croûte" sont orthogonaux aux segments des courbes de niveau, ils indiquent l'orientation de la pente, mais également le sens d'écoulement de tout fluide qui

serait déversé sur ce terrain. Ils peuvent donc être utilisés en tant que vecteur dans l'élaboration d'un modèle d'écoulement. La figure 9.4 donne un exemple de l'emploi des segments "anti-croûte" pour la conception d'un tel modèle.

## 10. CONCLUSION

Nous avons présenté dans cette recherche quelques domaines d'applications utilisant les courbes et pouvant tirer profit des propriétés géométriques uniques de l'axe médian. Pour n'en souligner que quelques-uns, citons la reconnaissance de formes et de caractères, l'estimation de bassin versant, la conception de modèle d'écoulement, le placement d'éléments textuels, la généralisation cartographique et l'enrichissement de modèle numérique d'élévation. Ces diverses applications n'auraient pu être explorées avec autant de facilité sans l'algorithme d'extraction de la croûte et de l'axe médian développé par Gold (1999).

Nous avons orienté cette recherche tout particulièrement sur deux de ces applications et nous avons développé les algorithmes appropriés afin de les implanter dans un programme informatique. Ainsi, nous avons pu mettre au point des solutions potentielles pour résoudre deux problèmes rencontrés dans le domaine de la géomatique. Premièrement, les algorithmes de généralisation cartographique entraînent parfois des chevauchements d'objets lorsque ceux-ci sont près les uns des autres (Weibel, 1990). Deuxièmement, la génération d'une triangulation à partir de courbes de niveau produit des triangles plats aux zones d'inflexion prononcée (Van Kreveld, 1997). Nous avons aussi démontré que les solutions développées fonctionnent bien à l'intérieur d'un cadre académique.

Le développement de l'algorithme de généralisation par rétraction de l'axe médian a permis de faire ressortir certaines des lacunes propres aux algorithmes traditionnels de généralisation, notamment quant aux risques de chevauchement de lignes et à la perte de la topologie provoquée par la modification des objets cartographiques. Les résultats obtenus, suite au traitement de divers jeux de données (carte polygonale, feuille d'érable, etc.), ont clairement démontré le potentiel de cet algorithme dans l'élimination des perturbations sur les courbes ainsi que pour l'adoucissement des angles. D'un point de vue académique, les résultats peuvent être qualifiés de satisfaisants, tant au niveau du produit final que des temps de traitement. Cependant, il serait intéressant d'explorer et, éventuellement d'implanter, toujours dans l'optique d'accroître les performances de cette technique, un algorithme rendant possible le déplacement de vertex dans un maillage Delaunay/Voronoï, ceci afin de remplacer la procédure actuelle de positionnement des vertex sur les arcs des cercles. De plus, l'adaptation de cet algorithme, afin de permettre la

rétraction des branches de n'importe quel ordre, serait à envisager. Ceci accroîtrait considérablement l'éventail des applications pouvant utiliser cet d'algorithme.

La conception d'un module de visualisation 3D, en parallèle à cette recherche, aura permis, en bout de ligne, d'évaluer les qualités et les performances de la méthode d'enrichissement. Sans lui, l'évaluation des résultats aurait été difficilement possible, car nous tenions à utiliser l'algorithme d'interpolation "aire volée" pour concevoir les modèles. Dans la même veine, les résultats fournis par le logiciel *Surfer* ont permis de démontrer que l'enrichissement devrait être une étape indissociable de l'interpolation dans les cas où les fichiers de courbes de niveau seraient susceptibles de produire un grand nombre de triangles plats. Il est apparu que, aussi performant que puisse être l'algorithme d'interpolation, si le fichier de données n'est pas assez riche en information, le produit résultant ne peut être que de piètre qualité. En effet, un fichier bien enrichi auquel a été appliqué un simple algorithme d'interpolation linéaire, comme "l'aire volée", a fourni de meilleurs résultats qu'un algorithme de krigeage. La méthode d'enrichissement s'est donc avérée particulièrement efficace et nous souhaitons vivement que, dans un proche avenir, ses possibilités soient davantage explorées.

En somme, cette recherche a démontré que, dans le domaine des sciences géomatiques, le potentiel d'applicabilité de l'axe médian, mais aussi du diagramme Voronoï et de la triangulation Delaunay, est incontestable. Les applications qui ont été développées ici ne sont qu'un aperçu des nombreuses autres qui n'ont pas encore eu la chance d'être étudiées.

## RÉFÉRENCES BIBLIOGRAPHIQUES

- Alt, H. & O. Schwarzkopf. 1995. The Voronoi diagram of curved objects. Proc. 11th Annual ACM Symposium on Computational Geometry, pp 89-97
- Amenta, N., M. Bern & D. Eppstein. 1998. The crust and the beta-skeleton: combinatorial curve reconstruction. *Graphical Models and Image Processing*, 60, pp 125-135
- Aronoff, S. 1989. *Geographic Information Systems: A Management Perspective*, WDL Publications, Ottawa, Canada, 290 pages
- Aumann, G., H. Ebner & L. Tang. 1991. Automatic derivation of skeletons lines from digitized contours. *ISPRS Journal of Photogrammetry and Remote Sensing*, 46, pp. 259-268
- Aurenhammer, F. 1991. Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure. *ACM Computing Surveys*, v. 23 no. 3, pp. 345-405
- Aurenhammer, F. & R. Klein. 1996. *Voronoi Diagrams*, Technical Report 198, Department of Computer Science, Fern Universität Hagen, Germany, 92 pages
- Blum, H. 1967. A transformation for extracting new descriptors of shape, *Models for the Perception of Speech and Visual Form*, Cambridge, Mass.: M.I.T. Press pp. 153-171
- Boulianne, M. 1999. *Photogrammétrie avancée*, Note de cours, Département des Sciences Géomatiques, Université Laval
- Brassel, K.E. & Weibel, R. 1988. A review and conceptual framework of automated map generalization, *International Journal of Geographical Information Systems*, v. 2(3), pp. 229-244
- Bundy, G.L., Jones, C.B. & Furse, E. 1995. Holistic generalization of large-scale cartographic data, *GIS and Generalization - Methodology and Practice*. Édité par Muller, J.C., Lagrange J.-P. & Weibel, R. (London: Taylor and Francis Ltd.), pp. 106-119.
- Burge, M. & G. Monagan. 1995a. Using the Voronoi tessellation for grouping words and multi-part symbols in documents, *Proceedings, Vision Geometry IV*, pp. 116-123
- Burge, M. & G. Monagan. 1995b. Extracting Words and Multi-part Symbols in Graphics Rich Documents. Proc. 8th International Conference on Image Analysis and Processing, ICIAP, (San Remo, Italy), LNCS, Springer Verlag, pp. 533-538
- Burrough, P.A. 1986. *Principles of Geographical Information Systems for Land Resources Assesment*, Clarendon Press, Oxford, pp. 39-56

Carrara, A., G. Bitelli & R. Carla. 1997. Comparaison of techniques for generating digital terrain models from contour lines, *Int. J. Geographical Information Science*, Vol. 11, No. 5, pp 451-473

Chithambaram, R., K. Beard & R. Barrera. 1991, Skeletonizing polygons for map generalization, *Technical papers ACSM-ASPRS*, Vol. 2, pp 44-55

Chongjiang, D. 1996. An interpolation method for grid-based terrain modelling, *The computer journal*, Vol. 39, No. 10, pp. 837-843

Christensen, A.H.J. 1987. Fitting a triangulation to contours, In *Proc. Auto-Carto 8*, pp 57-67

Douglas, D.H. & Peucker, T.K. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, *The Canadian Cartographer*, pp 112-123

Eklundh, L & U. Martensson. 1995, *Int. J. Geographical Information Science*, Vol. 9, No. 3, pp 329-340

Garcia, A.B, C.G. Nicieza, J.B.O. Meré & A.M. Diaz. 1992. A contour line based triangulation algorithm, In *Proc. 5 th Int. Symp. on Spatial Data Handling*, pp 411-423

Gold, C.M. & U.M. Maydell. 1978 Triangulation and spatial ordering in computer cartography, *Proceedings of the Canadian Cartographic Association Annual Meeting (Vancouver: Canadian Cartographic Association)*, pp. 69-81

Gold, C.M, & S. Cormack. 1987. Spatially ordered networks and topographic reconstructions, *Int. J. Geographical Information Systems*, Vol. 1, no. 2, pp. 137-148

Gold, C.M. 1988. Point and area interpolation and the digital terrain model, *Proceedings, Second International Seminar on Trends and Concerns of Spatial Sciences*, (Fredericton: University of New Brunswick), pp. 133-147.

Gold, C.M. 1989. Surface interpolation, spatial adjacency and G.I.S, *Three Dimensional Applications in Geographic Information Systems (J.Raper, ed.)*, Taylor and Francis, Ltd., London, pp. 21-35.

Gold, C.M. 1994. An object-based method for modelling geological surfaces containing linear data. *Proceedings, Annual Meeting of the International Association for Mathematical Geology*, Mont Tremblant, Québec, October 1994, pp. 141-146.

Gold, C.M. J. Nantel & W. Yang. 1996. Outside-In: an alternative approach to forest map digitizing, *International Journal of Geographical Information Systems*, Vol. 10, No. 3, pp. 291-310

Gold, C.M. 1998a. The Quad-Arc data structure, Proceedings, 8th International Symposium on Spatial Data Handling, Vancouver, BC, pp. 713-724.

Gold, C.M. 1998b. Structures d'algorithmes et de données spatiales, Note de cours, Département des Sciences Géomatiques, Université Laval

Gold, C.M., P.R., Remmele & T. Roos. 1998. Fully dynamic and kinematic Voronoi diagrams in GIS (sous presse), Algorithmica.

Gold, C.M. 1999. Crust and anticrust: a one-step boundary and skeleton extraction algorithm, Proceedings of the ACM Conference on Computational Geometry, Miami, Florida, pp. 189-196.

Gold, C.M. & J. Snoeyink. 1999a. A one-step crust and skeleton extraction algorithm, Algorithmica, (en attente de publication)

Gold, C., D. Thibault & S. Liu. 1999b. Map generalization by skeleton retraction, ICA Workshop on Generalisation, Ottawa, August 1999

Green, P.J. & R.R. Sibson. 1978. Computing Dirichlet tessalations in the plane, Computer Journal, 21, pp. 168-173

Guibas, L. & J. Stolfi. 1985. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. ACM Transactions on Graphics, pp. 74-123

Lam, N.S. 1983. Spatial interpolation methods : A review, The American Cartographer 10(2), pp. 129-149

Laurini, R. & D. Thompson. 1992. Fundamentals of spatial information systems, A.P.I.C. Series, no. 37, Academic Press, 680 pages

Li, Z. 1994. A comparative study of the accuracy of digital terrain models (DTMs) based on various data models, ISPRS Journal of Photogrammetry and Remote Sensing, 49(1), pp 2-11

Makarovic, B. 1973. Progressive sampling for digital terrain models. ITC Journal (3), pp. 397-416

McMaster, R.B. 1989. The integration of simplification and smoothing algorithms in line generalization. Cartographica, v. 26, pp 101-121

Miller, V.C. & M.E. Westerback. 1989. Interpretation of Topographic Maps, Merrill Publishing Company, Columbus, Ohio, 241 pages

Ogniewicz, R. & M. Ilg. 1990. Skeletons with Euclidian metric and correct topology and their application in object recognition and document analysis. Proc. 4 th International Symposium on Spatial Data Handling, v.1, pp. 15-24

Ogniewicz, R.L. 1994. Skeleton-space: a multiscale shape description combining region and boundary information. Proceedings of Computer Vision and Pattern Recognition, 1994, pp. 746-751

Peucker, T.K., R.J. Fowler, Little J.J. & Mark D.M. 1978. The triangulated irregular network, Proceedings of the ASP Digital Terrain Models (DTM) Symposium, American Society of Photogrammetry, Falls Church Virginia, pp. 516-540

Shamos, M.I. & D. Hoey. 1975. Closest-point problems, Proc. 16th Annual IEEE Sympos. Found. Comput. Sci., pp. 151-162

Thibault, D. & C. Gold. 1999. Terrain reconstruction from contours by skeleton retraction, Second International Workshop on Dynamic and Multi-Dimensional GIS, Beijing, China, October 1999, pp. 23-27

Van Kreveld, R. 1997. Digital Elevation Models and TIN algorithms; In Van Kreveld, M. , J. Nievergelt, T. Roos & P. Widmayer., Algorithmic foundations of geographic information systems, Lecture note in computer sciences, Springer-Verlog, Berlin, pp. 37-78

Watson, D.F. 1994. Contouring. A guide to the analysis and display of spatial data, Oxford, Elsevier, 321 pages

Weibel, R. & M. Heller. 1990. Digital terrain modelling, Maguire, D.J. & D.W. Rhind (eds.) Geographical Information Systems: Principles and applications, London, pp. 269-297

Weibel, R., 1995. Three essential building blocks for automated generalization, In: GIS and Generalization - Methodology and Practice, Édité par Muller, J.C., Lagrange J.-P. & Weibel, R. (London: Taylor and Francis Ltd.), pp. 56-69.